

LEMBAR PENGESAHAN

JUDUL PROYEK AKHIR

Rancang Bangun TaBox (*Task Box*)
Sebagai Wadah Pengumpul Tugas Mahasiswa

Oleh :

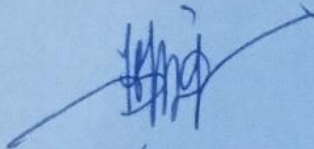
Dodi /0031607

Rahmah Nadiyah /0031621

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan Diploma III Politeknik Manufaktur Negeri Bangka Belitung

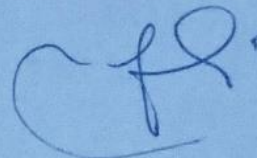
Menyetujui,

Pembimbing 1



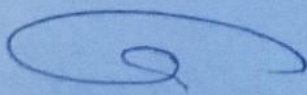
Muhammad Iqbal Nugraha, M.Eng

Pembimbing 2



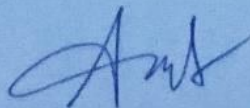
Dr. Parulian Silalahi, M.Pd

Penguji 1



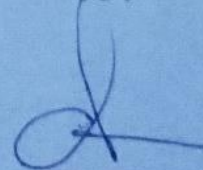
Aan Febriansyah, M.T.

Penguji 2



Sari Mubaroh, M.Pd.

Penguji 3



Ocsirendi, M.T.

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan di bawah ini:

Nama Mahasiswa 1 : Dodi NIRM : 0031608
Nama Mahasiswa 2 : Rahmah Nadiyah NIRM : 0031621

Dengan Judul : Rancang Bangun TaBox (*Task Box*) Sebagai Wadah
Pengumpul Tugas Mahasiswa

Menyatakan bahwa laporan akhir ini adalah hasil kerja kami sendiri dan bukan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya dan bila ternyata dikemudian hari ternyata melanggar pernyataan ini, kami bersedia menerima sanksi yang berlaku.

Sungailiat, 1 September 2019

Nama Mahasiswa
1. Dodi
2. Rahmah Nadiyah

Tanda Tangan

.....
.....

ABSTRAK

Di Politeknik Manufaktur Negeri Bangka Belitung mengumpulkan tugas merupakan kewajiban bagi mahasiswa, baik itu tugas kuliah teori maupun praktik. Tugas seringkali diberikan oleh dosen/pengajar dalam bentuk paper dan dikumpulkan sesuai tenggat waktu yang telah ditentukan. Namun, ada beberapa kendala saat pengumpulan tugas antara lain: 1) Dosen/pengajar tidak ada di tempat; 2) Ruangan dosen/pengajar terkunci; 3) Ruang yang tidak memadai untuk pengumpulan tugas; 4) Tidak ada pengingat deadline antara dosen/pengajar dengan mahasiswa; dan 5) Tidak ada pencatatan waktu saat pengumpulan tugas, dengan demikian pengumpulan tugas menjadi kurang efisien. Dari kondisi tersebut munculah ide untuk membuat sebuah wadah pengumpulan tugas yang disebut TaBox. TaBox menggunakan sistem pengolah data dari Arduino yang dikontrol penuh oleh admin dan user sebagai pengguna TaBox melalui RFID Tag yang digunakan sebagai kartu akses bagi user saat akan login ke sistem TaBox dengan memindai kartu akses ke RFID Reader. Hasil yang diperoleh, antara lain user dapat mengatur waktu deadline pengumpulan tugas pada box tugas yang digunakan. Kemudian hasil dari pengumpulan tugas dapat dipantau dari LCD TFT, berupa data jumlah tugas yang masuk beserta waktu pengumpulan tugas. Sehingga dengan adanya TaBox ini 5 kendala sebelumnya dapat terselesaikan. Dengan alat ini pengumpulan tugas menjadi lebih efisien dibanding pengumpulan tugas secara manual bagi dosen/pengajar dan memudahkan dosen/pengajar mengetahui waktu pengumpulan tugas mahasiswa berdasarkan fungsi monitor pada LCD TFT.

Kata kunci : Task Box, Arduino, RFID, LCD TFT

ABSTRACT

In the State Polytechnics Manufacture of Bangka Belitung, collecting assignments is an obligation for college students, both that theoretical and practical assignments. Assignments are often given by lecturer in paper form and collected according to predetermined deadlines. However, there are several obstacles when collecting tasks including: 1) The lecturer is not in place; 2) lecturer's room is locked; 3) Insufficient space for collection of tasks; 4) There are no deadline reminders between lecturer and students; and 5) There is no recording of time when collecting the tasks, thus the task collection becomes less efficient. From these conditions come the idea to create a task collection container called TaBox. TaBox uses a data processing system from Arduino which is fully controlled by the admin and user as TaBox users via RFID Tag as an access card for the user when logging by scanning the access card to the RFID Reader. The results obtained, among others, the user can set the deadline for the collection of tasks in the task box used. Then the results of the collection of tasks can be monitored from the LCD TFT, consists of data on the number of tasks entered and the time of collection of tasks. So, 5 previous obstacles can be resolved by TaBox. Because TaBox makes the task collection becomes more efficient than the manual task collection for the lecturer and makes it easier for the lecturer to know the time of the student assignment based on the monitor function on the TFT LCD.

Keywords: Task Box, Arduino, RFID, LCD TFT

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Puji dan syukur penulis panjatkan kehadirat Allah SWT yang telah memberi rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan laporan serta proyek akhir ini dengan baik dan tepat waktunya. Tujuan dari pembuatan laporan proyek akhir ini yaitu sebagai salah satu persyaratan dan kewajiban mahasiswa untuk menyelesaikan kurikulum program pendidikan Diploma III di Politeknik Manufaktur Negeri Bangka Belitung. Adapun judul proyek akhir ini adalah “RANCANG BANGUN TABOX (TASK BOX) SEBAGAI WADAH PENGUMPUL TUGAS MAHASISWA”.

Pada kesempatan ini penulis menyampaikan rasa terimakasih yang sebesar-besarnya atas segala petunjuk, bimbingan dan bantuannya kepada:

1. Bapak M. Iqbal Nugraha, M. Eng., selaku Dosen Pembimbing I yang telah membantu memberikan arahan dan bimbingan dalam penulisan serta pembuatan alat pada penyelesaian proyek akhir ini.
2. Bapak Dr. Parulian Silalahi, M. Pd., selaku Dosen Pembimbing II yang telah membantu memberikan arahan dan bimbingan dalam penulisan serta pembuatan alat pada penyelesaian proyek akhir ini.
3. Bapak Eko Sulistyio, M. T., selaku Ketua Program Studi Diploma III Teknik Elektronika Politeknik Manufaktur Negeri Bangka Belitung.
4. Bapak Aan Febriansyah, M. T., selaku Ketua Jurusan Teknik Elektro dan Informatika Politeknik Manufaktur Negeri Bangka Belitung.
5. Seluruh Dosen, Instruktur dan Staff pengajar Jurusan Teknik Elektro dan Informatika Politeknik Manufaktur Negeri Bangka Belitung yang telah mengajarkan banyak ilmu pengetahuan.
6. Kepada orangtua penulis yang telah memberikan nasihat, perhatian dan do'a kepada penulis selama menjalani pendidikan.
7. Teman-teman mahasiswa seperjuangan yang bersama penulis selama menempuh pendidikan di Politeknik Manufaktur Negeri Bangka Belitung.

8. Pihak-pihak lain yang telah memberikan bantuan secara langsung maupun tidak langsung dalam pembuatan proyek akhir ini yang tidak dapat disebutkan satu per satu.

Penulis menyadari masih banyak kekurangan dalam penulisan proyek akhir ini dikarenakan keterbatasan ilmu yang penulis miliki untuk membuat proyek akhir ini jauh dari sempurna. Untuk itu dengan tidak mengurangi rasa hormat, dengan segala kerendahan hati, penulis sangat mengharapkan saran atau kritik yang sifatnya membangun dan bermanfaat untuk kesempurnaan proyek akhir ini.

Selanjutnya penulis berharap semoga proyek akhir ini dapat bermanfaat bagi para pembaca umumnya dan dapat dikembangkan untuk memperlancar dalam melaksanakan tugas yang berkaitan dengan proyek akhir ini.

Sungailiat, September 2019

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN	Error! Bookmark not defined.
PERNYATAAN BUKAN PLAGIAT	Error! Bookmark not defined.
ABSTRAK	ii
<i>ABSTRACT</i>	v
KATA PENGANTAR	vi
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN.....	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	2
1.4 Tujuan Proyek Akhir.....	3
BAB II DASAR TEORI	4
2.1 Sistem.....	4
2.1.1 Sistem Absensi berbasis RFID	4
2.1.2 Sistem Alarm berbasis Arduino	5
2.2 Konsep Perancangan Kontruksi	5
2.3 Perancangan Sistem Kontrol.....	7
2.3.1 Arduino Mega 2560	7

2.3.2	Arduino Uno.....	8
2.3.3	RFID RC522	9
2.3.4	RTC DS3231	10
2.3.5	Serial SPI Arduino 7" TFT LCD RA8875	11
2.3.6	Servo.....	12
2.3.7	<i>Solenoid Door Lock</i>	13
2.3.8	Relai 1 <i>channel</i>	14
2.3.9	TCRT5000 <i>Infrared Reflective</i> atau <i>Proximity Sensor</i>	15
BAB III METODE PELAKSANAAN		16
3.1	Metode Pelaksanaan Pembuatan Proyek Akhir	16
3.2	Rancangan Algoritma Kerja Alat.....	19
3.3	Rancangan <i>Hardware</i>	21
3.3.1	Rancangan <i>Hardware Box</i> Tugas.....	21
3.3.2	Rancangan <i>Hardware Box</i> Kontrol	22
3.4	Diagram Blok.....	22
BAB IV PEMBAHASAN.....		24
4.1	Pembuatan <i>Hardware Box</i> Tugas dan <i>Box</i> Kontrol.....	24
4.1.1	Pembuatan <i>Hardware Box</i> Tugas.....	24
4.1.2	Pembuatan <i>Hardware Box</i> Kontrol	26
4.2	Wiring Diagram Alat	26
4.3	Pengujian Perangkat Catu Daya.....	29
4.4	Pengujian Sensor TCRT5000	31
4.5	Pengujian RFID RC522	32
4.6	Menu pada LCD TFT RA8875	35
4.6.1	Menu Admin	36

4.6.2 Menu User	42
4.7 Pengujian Fungsi TaBox	45
4.7.1 Penyimpanan Data User	45
4.7.2 Deteksi Tugas	48
4.7.3 Sistem <i>Deadline</i>	51
BAB V KESIMPULAN DAN SARAN.....	53
5.1 Kesimpulan	53
5.2 Saran	53
DAFTAR PUSTAKA	xv

DAFTAR TABEL

Tabel 4.1 Hasil Pengujian Rangkaian Penurun Tegangan	30
Tabel 4.2 Hasil Pengujian Jarak Pembacaan RFID <i>Reader</i>	33
Tabel 4.3 Kondisi RFID <i>Reader</i> Terhadap RFID <i>Tag</i>	33
Tabel 4.4 Hasil Pengujian Penyimpanan Data <i>User</i>	48
Tabel 4.5 Hasil Pengujian Sensor TCRT Mendeteksi Tugas.....	50
Tabel 4.6 Hasil Pengujian Sistem <i>Deadline</i> Pada <i>Box 00</i>	51

DAFTAR GAMBAR

Gambar 2.1 <i>Software</i> SolidWorks.....	6
Gambar 2.2 Pinout Arduino Mega 2560.....	8
Gambar 2.3 Arduino Uno.....	9
Gambar 2.4 RFID RC522	9
Gambar 2.5 RTC DS3231	11
Gambar 2.6 Serial SPI Arduino 7" TFT LCD RA8875	11
Gambar 2.7 Prinsip Kerja Motor Servo	12
Gambar 2.8 Solenoid Door Lock	13
Gambar 2.9 <i>Pinout</i> Relay.....	14
Gambar 2.10 TCRT5000.....	15
Gambar 3.1 <i>Flowchart</i> Pembuatan Proyek Akhir.....	16
Gambar 3.2 <i>Flowchart</i> Rancangan Algoritma Kerja Alat	19
Gambar 3.3 Rancangan Gambar TaBox	21
Gambar 3.4 <i>Hardware</i> Box Kontrol	22
Gambar 3.5 Diagram Blok Sistem Kontrol TaBox	23
Gambar 4.1 Hasil Pembuatan <i>hardware</i> Box Tugas.....	24
Gambar 4.2 Peletakkan Komponen di Box Tugas	25
Gambar 4.3 Box Kontrol TaBox	26
Gambar 4.4 <i>Interface</i> LCD TFT dengan Arduino Mega 2560	27
Gambar 4.5 <i>Wiring Diagram</i> Keseluruhan Sistem TaBox	27
Gambar 4.6 Skematik Rangkaian Box Komponen	29

Gambar 4.7 Rangkaian Catu Daya.....	29
Gambar 4.8 Saat Sensot Tidak Mendeteksi Obyek.....	31
Gambar 4.9 Saat Sensor Mendeteksi Obyek.....	32
Gambar 4.10 Metode Pengambilan Data Jarak Pembacaan RFID Reader	32
Gambar 4.11 Serial Monitor Pembacaan ID RFID Tag.....	34
Gambar 4.12 Tampilan Menu Utama pada LCD TFT.....	35
Gambar 4.13 Tampilan Menu <i>Admin</i>	36
Gambar 4.14 <i>Flowchart</i> Sistem Admin Untuk Penambahan User Baru	37
Gambar 4.15 <i>Flowchart</i> Sistem Admin Untuk Penghapusan User	38
Gambar 4.16 Tampilan Fitur Pilih Box pada Menu Admin	39
Gambar 4.17 Tampilan Menu Input Mata Kuliah pada Menu Admin.....	39
Gambar 4.18 Tampilan Menu <i>Deadline</i> pada Menu Admin.....	40
Gambar 4.19 Kondisi Relay Sebelum Tombol Buka Box Ditekan	41
Gambar 4.20 Kondisi Relay Setelah Tombol Buka Box Ditekan.....	41
Gambar 4.21 Tampilan Menu User.....	42
Gambar 4.22 Tampilan Menu Input Mata Kuliah User	43
Gambar 4.23 Tampilan Menu Input Deadline Pengumpulan Tugas.....	43
Gambar 4.24 Tampilan Menu Berisi Data Pengumpulan Tugas	44
Gambar 4.25 Pengujian <i>Input</i> Mata Kuliah <i>User</i>	46
Gambar 4.26 Pengujian <i>Input</i> Data Waktu <i>Deadline</i>	46
Gambar 4.27 Hasil Pengujian Tampilan Data <i>User</i> pada EEPROM.....	47
Gambar 4.28 Hasil Pengujian Data Pengumpulan Tugas	49

DAFTAR LAMPIRAN

Lampiran 1: Daftar Riwayat Hidup Penulis

Lampiran 2: Standart Operating Procedure (SOP) Penggunaan Alat

Lampiran 3: Program Arduino Uno

Lampiran 4: Program Arduino Mega 2560

BAB I

PENDAHULUAN

1.1 Latar Belakang

Salah satu kegiatan perkuliahan di perguruan tinggi adalah mengerjakan tugas. Mahasiswa tidak akan terlepas dari aktivitas belajar dan keharusan mengerjakan tugas-tugas studi yang bersifat akademis [1]. Tugas merupakan suatu hal yang harus dikerjakan oleh seorang mahasiswa atas perintah dari dosen/pengajar yang mengajar bidang studi tertentu. Terdapat 2 jenis tugas, yaitu *paper* dan *paperless*. *Paper* adalah tugas yang dibuat dalam bentuk cetak dan *paperless* adalah tugas yang dibuat dalam bentuk *file* digital yang disimpan dalam penyimpanan digital.

Di Politeknik Manufaktur Negeri Bangka Belitung mengumpulkan tugas merupakan kewajiban bagi mahasiswa, baik itu tugas kuliah teori maupun praktik. Tugas seringkali diberikan oleh dosen/pengajar dalam bentuk *paper* dan dikumpulkan sesuai tenggat waktu yang telah ditentukan. Dengan tenggat waktu tersebut, tugas dikumpulkan di atas meja dosen/pengajar yang berada di dalam ruangan, sehingga kurang efisien dan memiliki beberapa kendala saat pengumpulannya. Kendala tersebut antara lain: 1) Dosen/pengajar tidak ada ditempat; 2) Ruang dosen/pengajar terkunci; 3) Ruang yang tidak memadai untuk pengumpulan tugas; 4) Tidak ada pengingat *deadline* antara dosen/pengajar dengan mahasiswa; dan 5) Tidak ada pencatatan waktu saat pengumpulan tugas.

Kondisi demikian apabila dibiarkan dan tidak dicari alternatif lain akan berdampak pada proses pengumpulan tugas. Dampak tersebut dapat terlihat pada ketidaktertiban pengumpulan tugas baik dalam kedisiplinan mahasiswa terhadap tenggat waktu, pencatatan waktu tugas serta pada sistem manual pengumpulan tugas itu sendiri.

Untuk meningkatkan ketertiban dan kedisiplinan dalam pengumpulan tugas, maka diperlukanlah alat pengumpul tugas dengan membuat suatu wadah yang dapat dijadikan sebagai tempat untuk pengumpulan tugas yang memiliki

kualifikasi sebagai berikut: 1) Wadah tersebut tidak memerlukan kehadiran dosen/pengajar dalam pengumpulannya; 2) Wadah dapat diakses tanpa perlu masuk ke dalam ruangan dosen/pengajar; 3) Wadah tersebut memiliki ruang yang memadai untuk menampung tugas; 4) Wadah tersebut memiliki sistem pengingat *deadline* antara dosen/pengajar dengan mahasiswa; dan 5) Wadah yang memiliki sistem pencatatan waktu terhadap setiap tugas yang dikumpulkan. Maka dibuatlah TaBox (*Task Box*) sebagai wadah untuk menampung tugas mahasiswa. Dengan adanya alat ini bertujuan untuk mempermudah pengumpulan tugas bagi dosen/pengajar dan mahasiswa.

Alat ini menggunakan Arduino sebagai sistem kontrol utama TaBox. Alat ini akan berkerja berdasarkan data berupa mata kuliah dan waktu *deadline* pengumpulan tugas yang *diinput* oleh *user* melalui LCD TFT. Kemudian data tersebut akan di tampilkan pada layar LCD TFT. RFID Tag digunakan sebagai kartu akses bagi *admin* dan *user* saat akan login ke system TaBox melalui LCD TFT.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah yang akan dikaji oleh penulis dalam proyek akhir ini adalah:

1. Bagaimana merancang sistem pengumpulan tugas mahasiswa pada TaBox?
2. Bagaimana membuat sistem yang dapat menampilkan informasi mengenai box dan *deadline* pengumpulan tugas pada LCD TFT?
3. Bagaimana membuat sistem deteksi tugas beserta penyimpanan data waktu pada setiap tugas yang masuk?
4. Bagaimana membuat sistem *deadline* pengumpulan tugas?

1.3 Batasan Masalah

Mengingat permasalahan yang cukup kompleks serta untuk menghindari meluasnya ruang lingkup masalah, maka dalam penulisan ini hanya dibatasi pada masalah:

1. Sistem yang dibuat hanya sebatas untuk mengumpulkan tugas mahasiswa dengan kondisi yang berurutan dan tersusun rapi.

2. Alat ini memiliki sistem pencatatan waktu saat pengumpulan tugas.
3. Sistem ini dibuat dengan menggunakan Arduino.
4. Wadah ini dirancang untuk tugas berukuran A4 (210 x 297 mm) dengan ketebalan maksimal 1.5 cm dan berjumlah 30 tugas serta ditempatkan dalam satu area.

1.4 Tujuan Proyek Akhir

Adapun tujuan dari Rancang Bangun TaBox (*Task Box*) sebagai Wadah Pengumpul Tugas Mahasiswa adalah :

1. Merancang TaBox sebagai wadah untuk menampung tugas berbasis Arduino dengan *timer* yang memudahkan mahasiswa dan dosen/pengajar dalam pengumpulan tugas kuliah.
2. Membuat sebuah sistem untuk menampilkan informasi mengenai box dan *deadline* pengumpulan tugas yang ditampilkan dalam LCD TFT.
3. Membuat sistem deteksi tugas beserta penyimpanan data waktu pada setiap tugas yang masuk ke box tugas.
4. Membuat sistem *deadline* pengumpulan tugas menggunakan RTC DS3231 sebagai modul pengukur waktu *deadline*.

BAB II

DASAR TEORI

2.1 Sistem

Sistem dapat diartikan dengan pendekatan prosedur dan komponen, sistem dapat didefinisikan sebagai kumpulan dari prosedur-prosedur yang mempunyai tujuan tertentu [2]. Dalam pembuatan proyek akhir yang berjudul “Rancang Bangun TaBox (*Task Box*) Sebagai Wadah Pengumpul Tugas Mahasiswa” ini, penulis menerapkan beberapa sistem penelitian terdahulu yang berkaitan dengan konsep sistem yang diperlukan untuk membuat sistem TaBox.

2.1.1 Sistem Absensi berbasis RFID

Penulis mencari referensi dari makalah proyek akhir yang berjudul “Rancang Bangun Mesin Absensi Otomatis Dengan Menggunakan Sensor RFID Berbasis Arduino”. Dalam makalah tersebut, penulis makalah menggunakan RFID untuk membaca nomer ID yang kemudian nomer ID RFID Tag tersebut akan tersimpan sementara ke IC RFID Reader (penyimpanan terbatas) yang kemudian dikirim ke penyimpanan IC Arduino Uno (penyimpanan terbatas). Lalu terdapat sistem *admin* dan *user* yang penulis makalah gunakan untuk membedakan hak akses pada mesin absensi yang dibuat. Penulis makalah juga membuat mesin absensi mampu merekam waktu saat karyawan melakukan absensi [3].

Pada hasil proyek akhir yang dijadikan referensi tersebut, penulis menerapkan RFID Reader untuk membaca RFID Tag yang digunakan sebagai suatu ID bagi *user* dalam syarat mengakses TaBox. Hasil pembacaan RFID Reader tersebut akan tersimpan ke IC RFID Reader (penyimpanan sementara) yang kemudian dikirim ke Arduino Uno (penyimpanan tetap). Penulis juga menambahkan sistem *admin* dan *user* agar mempermudah dalam membedakan hak akses TaBox. Untuk perekaman waktu yang ada di mesin absensi, penulis juga menerapkan sistem tersebut, namun dengan tujuan yang berbeda. Penulis menerapkan perekaman waktu pada sistem deteksi tugas yang nantinya apabila ada

tugas yang masuk, sensor akan mendeteksi tugas tersebut sekaligus merekam waktu saat tugas terdeteksi oleh sensor.

2.1.2 Sistem Alarm berbasis Arduino

Referensi sistem alarm berbasis Arduino diambil dari laman web AnakKendali.com. Dimana sistem ini diterapkan oleh penulis web menggunakan *pushbutton*, LCD 16 x 2 I2C dan RTC DS3231. Cara kerja sistem tersebut adalah ada 3 buah *pushbutton* dimana fungsi *pushbutton* satu untuk pilihan menu Running yang menampilkan jam dan alarm, *set* nilai jam untuk alarm dan *set* nilai untuk alarm. Saat tidak ditekan (perintah awal, *default*) LCD akan menampilkan untuk pengaturan nilai jam alarm, ketika di tekan untuk kedua kalinya, LCD akan menampilkan untuk pengaturan nilai menit. Kemudian *pushbutton* dua untuk menaikkan nilai dan *pushbutton* tiga untuk menurunkan nilai [4].

Sistem ini kami terapkan sebagai pembatas waktu *deadline* pengumpulan tugas dimana prinsipnya sama yaitu, memasukkan batas waktu *deadline* pengumpulan tugas yang diinginkan setelah batas waktu *deadline* tersebut terpenuhi maka waktu pengumpulan tugas akan berakhir. Untuk sistem alarm pada referensi yang didapat dari laman web tersebut, saat waktu alarm terpenuhi akan ada bunyi *buzzer* sebagai indikator bahwa alarm selesai. Namun penulis mengganti *buzzer* dengan program servo untuk bergerak menutupi pintu tugas masuk sebagai tanda waktu *deadline* telah usai.

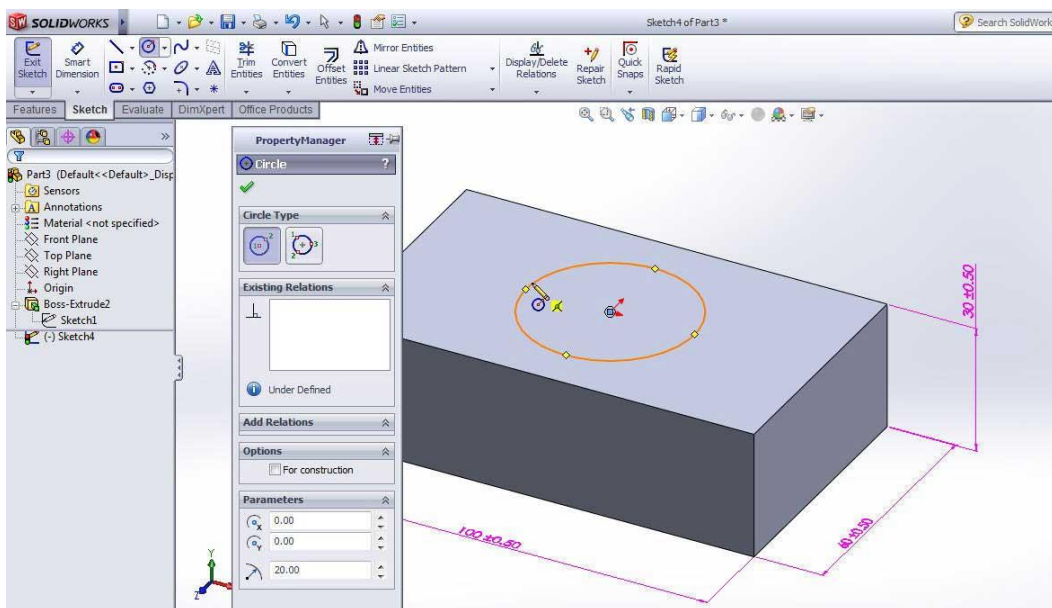
2.2 Konsep Perancangan Kontruksi

Dalam pembuatan suatu alat atau produk, dibutuhkan sebuah rancangan yang menjadi acuan dalam proses pembuatannya, agar pembuatan lebih sistematis dan terarah sehingga kesalahan yang timbul dapat ditekan dan dihindari [5]. Penulis merancang kontruksi alat menggunakan SolidWorks.

SolidWorks adalah *software* CAD 3D yang mudah digunakan untuk merancang *part* permesinan atau susunan *part* permesinan yang berupa *assembling* dengan tampilan 3D untuk merepresentasikan *part* sebelum *real part* nya dibuat atau tampilan 2D untuk gambar proses permesinan. *Software* ini adalah *software*

Automasi Design berbasis *parametric* yang memudahkan pengguna mengedit *file-file* gambar yang sudah dibuat. Dengan SolidWorks, pengguna dapat mendesain gambar dengan intuitif. SolidWorks biasa digunakan untuk membuat gambar sederhana maupun gambar-gambar yang kompleks atau rumit [6].

Pada Gambar 2.1 menunjukkan proses dalam mendesain menggunakan SolidWorks [6].



Gambar 2.1 Software SolidWorks

SolidWorks adalah sistem *multidocuments* yang menjadikan SolidWorks dapat menggunakan berbagai jenis *file* untuk keperluan yang berbeda. *Part*, *drawing*, dan *assembly* merupakan beberapa contoh jenis *file* untuk keperluan yang berbeda. Ada banyak tipe *file* SolidWorks yang terkait dengan *file* lain. *Drawing* merujuk pada *part* untuk membuat tampilan detail. *Assembly* merujuk *part* dan *assembly* lainnya. Berbagai jenis *file* sepenuhnya asosiatif dalam SolidWorks, jadi perubahan pada *part* menyebar ke setiap *drawing* atau *part* yang menggunakan bagian tersebut [7].

Berikut merupakan *template* utama yang disediakan oleh SolidWork [7]:

1. *Part*

File part terdiri dari kumpulan fitur *part* (*bases, extrude, revolve*). SolidWorks melampirkan ekstensi *.sldrpt* ke *file part* [7].

2. *Assembly*

Kumpulan *part* terkait yang disimpan dalam satu *file* dengan ekstensi *.sldasm*. Suatu perakitan dapat terdiri dari dua hingga lebih dari seribu komponen, yang dapat berupa *part* atau rakitan lain yang disebut *subassemblies* [7].

3. *Drawing*

Dokumen 2D yang berisikan desain *File drawing* terdiri dari satu lembar atau lebih yang berisi tampilan model yang berbeda. SolidWorks memberikan *drawing file* ekstensi *.slddrw* [7].

2.3 Perancangan Sistem Kontrol

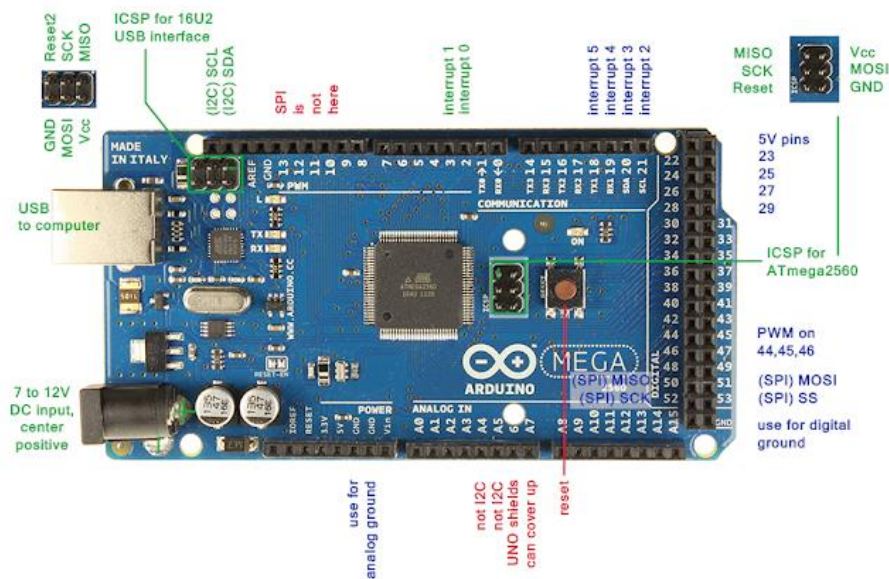
Sistem pengendalian (*system control*) merupakan sekumpulan peralatan yang bekerja sama dengan tujuan untuk mengendalikan sesuatu [8]. Sistem kontrol merupakan sebuah sistem yang memiliki hubungan satu sama lain antar komponen yang akan membentuk suatu konfigurasi sistem yang akan memberikan respon atau keluaran sistem yang diharapkan. Sistem kontrol dapat memberikan perintah, mengontrol sistemnya sendiri, atau sistem lainnya sehingga didapatkan keluaran sistem yang dikehendaki [9]. Sistem yang akan di rancang oleh penulis membutuhkan komponen elektronika yang dapat mendukung kinerja sistem absensi berbasis RFID dan sistem alarm berbasis Arduino.

Berikut penulis jelaskan komponen serta peralatan yang dibutuhkan dalam membuat sistem TaBox.

2.3.1 Arduino Mega 2560

Arduino Mega 2560 adalah papan pengembangan mikrokontroler yang berbasis Arduino dengan menggunakan *chip ATmega 2560*. *Board* ini sudah sangat lengkap dan memiliki segala sesuatu yang dibutuhkan untuk sebuah

mikrokontroler. Mikrokontroler itu sendiri adalah *chip* atau IC (*Integrated Circuit*) yang bisa diprogram menggunakan PC. Tujuan menanamkan program pada mikrokontroler agar rangkaian elektronik dapat membaca *input*, memproses *input* dan menghasilkan *output* sesuai yang diinginkan. Adapun *pin out* diagram dari Arduino Mega 2560 dapat dilihat pada Gambar 2.2 [10].



Gambar 2.2 Pin out Arduino Mega 2560

Arduino tidak perlu perangkat *chip programmer* karena didalamnya sudah ada *bootloader* yang akan menangani *upload* program dari PC. Bahasa pemrograman relatif mudah karena *software* Arduino dilengkapi dengan kumpulan *library* yang cukup lengkap. Bahasa pemrograman yang digunakan adalah bahasa C [10]. Arduino Mega digunakan sebagai pengolah data utama untuk memprogram LCD TFT, RTC DS3231, relai, sensor TCRT5000 dan servo.

2.3.2 Arduino Uno

Arduino Uno adalah *board* sirkuit berbasis mikrokontroler ATmega328. IC (*Integrated Circuit*) ini memiliki 14 *input/output* digital (6 *output* untuk PWM), 6 analog *input*, resonator kristal keramik 16 MHz, koneksi USB, *socket adaptor*, *pin header* ICSP, dan tombol *reset*. Untuk pemrograman cukup menggunakan koneksi

USB tipe A ke tipe B. Sama seperti yang digunakan pada USB printer [11]. Berikut merupakan Arduino Uno yang dapat dilihat pada Gambar 2.3 dibawah ini [12].



Gambar 2.3 Arduino Uno

Arduino Uno digunakan sebagai mikrokontroler yang mengolah data pada RFID Reader.

2.3.3 RFID RC522

RFID RC522 (*Radio Frequency Identification*) merupakan suatu teknologi yang memanfaatkan frekuensi radio sebagai pengidentifikasian terhadap suatu obyek. RFID mempunyai 2 bagian komponen utama yang tak dapat dipisahkan, yaitu RFID Tag dan RFID Reader, yang ditunjukkan pada Gambar 2.4 [13].



Gambar 2.4 RFID RC522

1. RFID Tag

Merupakan sebuah perangkat yang akan diidentifikasi oleh RFID *reader* yang dapat berupa perangkat pasif maupun aktif yang berisi suatu data atau informasi. Perangkat pasif tidak menggunakan catu daya, sedangkan perangkat aktif wajib menggunakan catu daya. Dipasaran yang paling banyak digunakan yaitu tipe perangkat RFID *reader* yang pasif dikarenakan harganya yang relatif murah. Pada RFID *tag* terdapat 2 jenis, yaitu *Read-Write* dan *Only Read*. Selain itu RFID *tag* mempunyai 2 komponen utama yang penting, antara lain [13]:

- IC (*Integrated Circuit*): Sebagai pemroses informasi, modulasi serta demodulasi sinyal RF, yang beroperasi dengan catu daya DC.
- *Antenna*: Mempunyai fungsi untuk mengirim maupun menerima sinyal RF.

2. RFID Reader

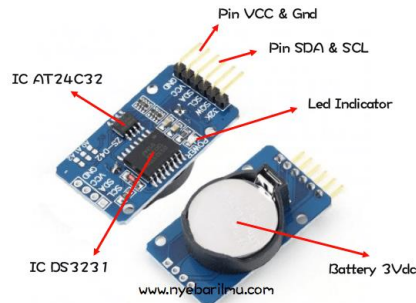
Berfungsi untuk membaca data dari RFID *tag*. RFID *reader* dibedakan menjadi 2 macam, antara lain [13] :

- Pasif: hanya bisa membaca data dari RFID *tag* aktif.
- Aktif: dapat membaca data RFID *tag* pasif

2.3.4 RTC DS3231

Modul RTC DS3231 merupakan salah satu jenis modul yang berfungsi sebagai RTC (*Real Time Clock*) atau pewaktuan digital serta penambahan fitur pengukur suhu yang dikemas kedalam 1 modul. Secara sederhana modul RTC merupakan sistem pengingat waktu dan tanggal yang menggunakan baterai sebagai pemasok power agar modul ini tetap berjalan. Modul DS3231 RTC ini pada umumnya sudah tersedia dengan baterai CR2032 3V yang berfungsi sebagai *back up* RTC apabila catu daya utama mati. 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 memberikan 32K EEPROM untuk menyimpan data [14].

Keterangan mengenai komponen yang ada di RTC dapat dilihat pada Gambar 2.5 dibawah ini [14].



Gambar 2.5 RTC DS3231

RTC dapat menyimpan waktu berupa jam, menit, detik, hari, tanggal, bulan dan tahun. Data ini akan terus tersimpan pada EEPROM AT24C32 sekalipun catu daya tidak ada atau mati. Hal ini membantu penulis dalam pembuatan sistem kontrol yang memerlukan fitur data *logging*, dengan presisi waktu yang lebih tinggi [14].

2.3.5 Serial SPI Arduino 7" TFT LCD RA8875

Serial SPI Arduino 7" TFT LCD *Touch Shield* RA8875 merupakan jenis LCD TFT yang umum dan sering digunakan sebagai *display unit* dan *input device* pada proyek yang menggunakan Arduino. Berikut merupakan Serial SPI Arduino 7" TFT LCD RA8875 yang dapat dilihat pada Gambar 2.6 dibawah ini [15].



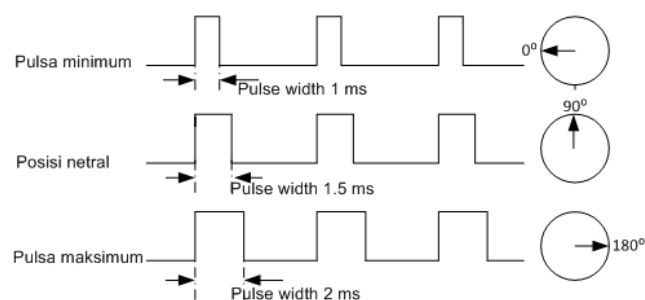
Gambar 2.6 Serial SPI Arduino 7" TFT LCD RA8875

LCD TFT ini memiliki ukuran yang besar yaitu 7.0" dengan *bright* (14 *white-LED backlight*) dan *color* 800 x 480 pixel. LCD TFT dikemas dalam bentuk *shield* yang sesuai dengan Arduino klasik (Uno/ Due/ Mega 2560) [15].

2.3.6 Servo

Motor servo adalah aktuator kontrol gerakan umum yang digunakan dalam robotika, perangkat yang dikendalikan radio, printer 3D dan sistem *drive* gerak lainnya. Motor servo terdiri dari tiga bagian utama: motor, sirkuit pengontrol dan sistem umpan balik, yang biasanya terdiri dari potensiometer yang terhubung ke poros *output* motor. Biasanya motor servo digunakan untuk mengontrol gerakan sudut antara 0 dan 180 derajat. Biasanya motor servo tidak dapat berputar lebih jauh kecuali mereka dimodifikasi, karena *mechanical stop* yang dibangun ke *output gear* utama atau potensiometer itu sendiri. Motor servo umumnya digunakan sebagai alternatif kinerja tinggi untuk motor *stepper* [16].

Motor servo dikendalikan dengan memberikan sinyal modulasi lebar pulsa (*Pulse Wide Modulation / PWM*) melalui kabel kontrol. Lebar pulsa sinyal kontrol yang diberikan akan menentukan posisi sudut putaran dari poros motor servo. Sebagai contoh, lebar pulsa dengan waktu 1.5 ms (mili detik) akan memutar poros motor servo ke posisi sudut 90°. Bila pulsa lebih pendek dari 1.5 ms maka akan berputar ke arah posisi 0° atau ke kiri (berlawanan dengan arah jarum jam), sedangkan bila pulsa yang diberikan lebih lama dari 1.5 ms maka poros motor servo akan berputar ke arah posisi 180° atau ke kanan (searah jarum jam). Lebih jelasnya perhatikan Gambar 2.7 dibawah ini [17].



Gambar 2.7 Prinsip Kerja Motor Servo

Ketika lebar pulsa kendali telah diberikan, maka poros motor servo akan bergerak atau berputar ke posisi yang telah diperintahkan, kemudian berhenti pada posisi tersebut dan akan tetap bertahan pada posisi tersebut. Jika ada kekuatan eksternal yang mencoba memutar atau mengubah posisi tersebut, maka motor servo akan mencoba menahan atau melawan dengan besarnya kekuatan torsi yang dimilikinya (rating torsi servo). Namun motor servo tidak akan mempertahankan posisinya untuk selamanya, sinyal lebar pulsa kendali harus diulang setiap 20 ms (mili detik) untuk menginstruksikan agar posisi poros motor servo tetap bertahan pada posisinya [17].

2.3.7 Solenoid Door Lock

Pada proyek akhir ini penulis menggunakan *solenoid door lock*. Berikut *solenoid door lock* yang dapat dilihat pada Gambar 2.8 [18].

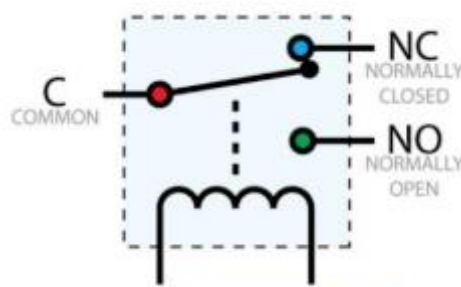


Gambar 2.8 Solenoid Door Lock

Solenoid door lock adalah salah satu *solenoid* yang difungsikan khusus sebagai *solenoid* untuk pengunci pintu secara elektronik. Solenoid ini mempunyai dua sistem kerja, yaitu *Normaly Close* (NC) dan *Normaly Open* (NO). Saat kondisi *solenoid* NC apabila diberi tegangan, maka solenoid akan memanjang (tertutup). Dan saat *solenoid* dalam kondisi NO maka hal yang terjadi merupakan kebalikan dari kondisi *solenoid* NC. *Solenoid Door Lock* yang digunakan membutuhkan tegangan kerja 12 VDC sehingga dibutuhkan *power supply* 12 V dan sebuah relai untuk mengaktifkannya [18].

2.3.8 Relai 1 channel

Relai adalah komponen yang dapat digunakan sebagai saklar elektronik. Secara singkat, cara kerja relai adalah memanfaatkan magnet buatan untuk memicu kontaktor dari keadaan *off* menjadi *on*, atau sebaliknya. Ibaratnya begini, jika ingin menhidupkan lampu, kita harus menekan saklar yang menempel pada dinding rumah. Tapi jika menggunakan relai, kita dapat menhidupkan atau mematikan lampu tanpa menyentuh saklar lagi. Relai disebut juga sebagai saklar elektronik, yaitu saklar yang dapat dikontrol dengan alat elektronik lainnya seperti Arduino. Berikut merupakan *pin out* diagram yang ada pada relai 1 channel yang ditunjukkan pada Gambar 2.9 [19].



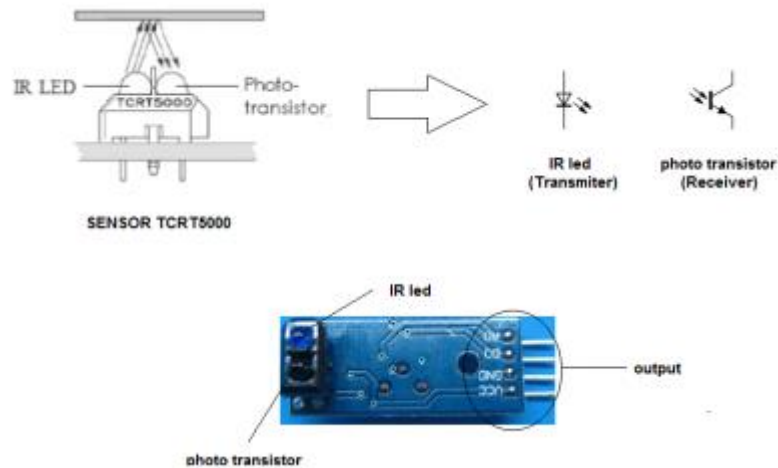
Gambar 2.9 *Pin out* Relai

Relai umumnya mempunyai 5 pin atau kaki, yang terdiri dari [19]:

1. Dua kaki untuk listrik + dan GND, jika arus dan tegangannya cukup, maka relai akan aktif yang ditandai dengan bunyi 'tek'.
2. Satu kaki sumber C (*common*), kaki yang akan dihubungkan ke kaki NC atau NO. Jika relai akan digunakan untuk mengontrol lampu rumah, maka kaki C disambungkan ke salah satu jalur listrik dari PLN.
3. Kaki NC (*Normally Close*), sebelumnya relai aktif, kaki NC nyambung ke kaki C. Tapi jika relai aktif, kaki NC terputus dari kaki C.
4. Kaki NO (*Normally Open*), sebelum relai aktif, kaki NO tidak tersambung kemanapun. Namun, saat relai aktif, kaki NO terhubung ke kaki C.

2.3.9 TCRT5000 *Infrared Reflective* atau *Proximity Sensor*

Sensor *proximity* mempunyai pengertian sebagai sensor yang mampu mendeteksi ada atau tidaknya suatu obyek. Prinsip kerja sensor TCRT5000 dapat dilihat pada Gambar 2.10 [20].



Gambar 2.10 TCRT5000

Prinsip kerja sensor TCRT5000 adalah mendeteksi warna berdasarkan penyerapan warna dan intensitas pemantulan sinar infra merah yang dipancarkan oleh *transmitter* (IR led) dan diterima oleh *receiver* (foto transistor). Dari perbedaan identitas tersebut digunakan sebagai bias pada basis foto transistor yang terkandung didalam sensor TCRT5000 [20].

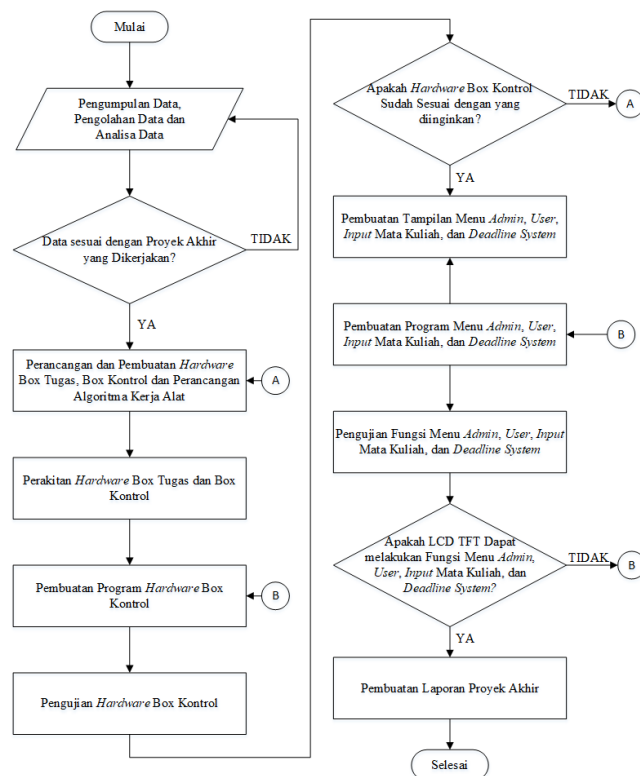
BAB III

METODE PELAKSANAAN

Dalam pembuatan proyek akhir dengan judul “Rancang Bangun TaBox (*Task Box*) Sebagai Wadah Pengumpul Tugas Mahasiswa” ini dilakukan beberapa metode penelitian yang bertujuan untuk mempermudah dalam penyelesaian proyek akhir. Adapun metode penelitian yang diterapkan dalam pembuatan proyek akhir seperti berikut:

3.1 Metode Pelaksanaan Pembuatan Proyek Akhir

Proses pembuatan proyek akhir dari mulai sampai selesainya proyek akhir dijelaskan pada *flowchart* pembuatan proyek akhir. *Flowchart* pembuatan proyek akhir ditunjukkan pada Gambar 3.1 berikut ini.



Gambar 3.1 *Flowchart* Pembuatan Proyek Akhir

Gambar 3.1 Merupakan *flowchart* pembuatan proyek akhir yang menjelaskan alur dari proses pengerjaan proyek akhir. Hal yang pertama dilakukan proses pembuatan proses akhir yaitu pengumpulan data yang diperlukan dalam menyelesaikan proyek akhir. Proses pengumpulan data dapat dilakukan secara langsung (primer) dan secara tidak langsung (sekunder). Maksud dari pengumpulan data secara langsung merupakan pengumpulan data yang diperoleh dari konsultasi dengan dosen pembimbing sedangkan untuk pengumpulan data secara tidak langsung (sekunder) merupakan data yang diperoleh dari membaca makalah proyek akhir dan dari referensi – referensi dari penelitian yang masih berhubungan dengan proyek akhir yang ingin dibuat. Data – data yang didapatkan dari proses pengumpulan data selanjutnya diolah dan dianalisa. Data yang telah diolah dan dianalisa dikumpulkan dan dipilih untuk menjadi referensi dan acuan dalam pembuatan proyek akhir.

Perancangan dan Pembuatan *hardware* box tugas, *box* kontrol dan perancangan algoritma kerja alat dibuat agar *hardware* dan prinsip kerja alat sesuai dengan spesifikasi proyek akhir. Pada proyek akhir ini desain rancangan dibuat dengan menggunakan *software* SolidWorks. *Box* Tugas dengan material yang memiliki ketebalan 6 mm dan berukuran panjang 30 cm, lebar 30 cm serta tinggi 60 cm. Pada tahap pembuatan *Box* Tugas menggunakan bahan dari triplek yang memiliki ketebalan 6 mm dan berukuran panjang 30 cm, lebar 30 cm serta tinggi 60 cm dan *box* kontrol menggunakan 2 *junction box* tipe Duradus MG Polos persegi panjang dengan ukuran panjang 22 cm, lebar 15 cm, tinggi 7 cm dan ukuran panjang 10 cm, lebar 7.6 cm dan tinggi 3.3 cm. Proses pembuatan *hardware* TaBox dilakukan di lingkungan kos Nangnung Utara.

Proses perakitan *hardware box* tugas dan *hardware box* kontrol dilakukan dengan cara merakit keseluruhan pada setiap bagian dari *hardware Box* Tugas, *hardware box* kontrol dan instalasi keseluruhan sistem TaBox yang disatukan dan menjadi satu kesatuan. Proses perakitan menggunakan proses-proses permesinan, misalnya pemotongan material triplek kerangka utama *box* dan proses pengeboran lubang untuk baut serta *wiring* pada komponen elektronika, misalnya penyusunan kabel-kabel dan peletakan komponen elektronika yang digunakan.

Pembuatan program *hardware box* kontrol harus sesuai dengan koneksi *pin* dan harus memenuhi spesifikasi proyek akhir yang ingin dibuat. Pembuatan program *hardware* elektrik *box* kontrol dibuat pada aplikasi Arduino IDE. Pada tahapan ini kegiatan yang dilakukan diantaranya pembuatan program RFID *reader* untuk pembacaan RFID *tag* agar dapat mengidentifikasi apakah itu *admin* atau *user* yang sedang *login*, program LCD TFT untuk menampilkan informasi mengenai *box* yang sedang aktif, program pencatatan waktu pada tugas yang telah dimasukkan ke dalam pintu masuk tugas TaBox dan program fitur *admin* untuk penambahan maupun penghapusan *user* serta wewenang akses ke semua *box* yang terprogram.

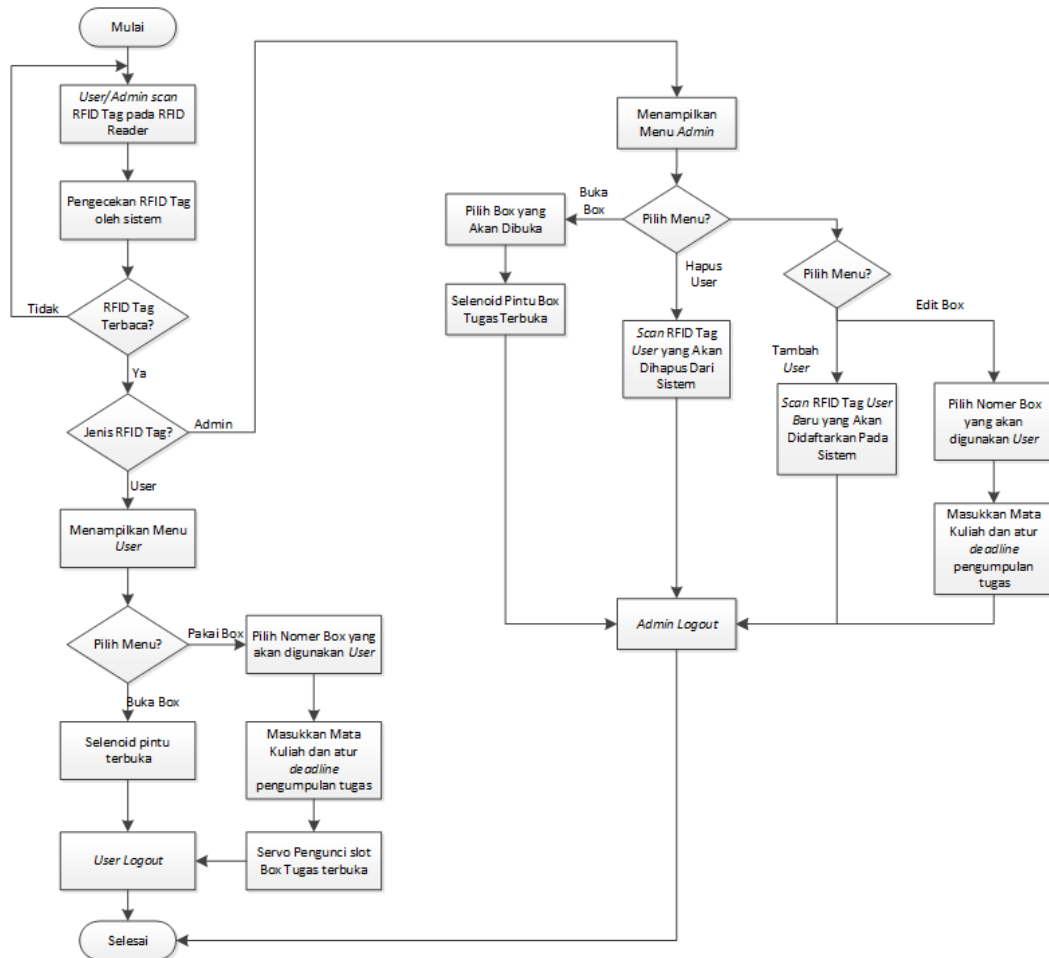
Setelah pembuatan program *hardware box* kontrol, maka dilakukan pengujian, yang dimana tujuannya untuk mengetahui apakah *hardware box* kontrol sudah sesuai dengan spesifikasi yang diinginkan. Pengujian dilakukan sebagai tolak ukur berhasil atau tidaknya sistem kontrol yang telah dibuat. Dengan begitu, penulis dapat mengevaluasi sistem kerja dari alat tersebut. Apabila sistem kerja tidak sesuai dengan yang diinginkan, maka kembali ke proses perancangan *hardware box* tugas dan *hardware box* kontrol pada TaBox.

Pembuatan tampilan dan pemrograman menu Admin, User, Input Mata Kuliah dan Deadline System dibuat menggunakan *software* Arduino IDE yang kemudian di upload pada Arduino yang terhubung dengan LCD TFT Shield RA8875. Pengujian fungsi menu Admin, User, Input Mata Kuliah dan Deadline System dilakukan dengan cara memasukkan data berupa mata kuliah dan *deadline* pengumpulan tugas melalui *touchscreen* pada LCD TFT RA8875. Pengujian dilakukan secara keseluruhan, baik dari segi Admin, User, Input Mata Kuliah dan Deadline System harus dapat bekerja sesuai dengan fungsinya.

Pembuatan laporan proyek akhir merupakan proses terakhir dalam pembuatan proyek akhir. Pada tahap ini, semua kegiatan selama pembuatan proyek akhir disimpulkan dan dilaporkan dalam bentuk laporan proyek akhir. Bertujuan untuk merangkum keseluruhan yang berkaitan dengan proyek akhir. Laporan ini dibuat berdasarkan format yang telah ditentukan pada Panduan Tugas Akhir yang berisikan latar belakang masalah, perancangan alat, pembuatan alat, pengujian alat, dan hasil pengujiannya.

3.2 Rancangan Algoritma Kerja Alat

Rancangan algoritma kerja alat merupakan rancangan dari proses cara kerja alat untuk memenuhi spesifikasi yang diinginkan. Berikut adalah *flowchart* rancangan algoritma kerja *hardware* ditunjukkan pada Gambar 3.2 berikut.



Gambar 3.2 *Flowchart* Rancangan Algoritma Kerja Alat

Pada Gambar 3.2 merupakan *flowchart* rancangan algoritma kerja alat yang digunakan dalam mencapai target penyelesaian sesuai dengan yang diinginkan. Saat *user* atau *admin* akan mengakses TaBox maka *admin* atau *user* perlu memindai RFID tag yang telah didaftarkan pada sistem. Saat RFID tag dipindai dan berhasil diidentifikasi oleh sistem, sistem secara otomatis akan menampilkan menu sesuai dengan jenis RFID tag yang telah dipindai pada RFID reader.

Ketika RFID *tag* yang dipindai teridentifikasi sebagai *user*, sistem akan menampilkan menu User. Menu User memiliki 2 fitur, yaitu pakai *box* dan buka *box*. Saat *user* menggunakan fitur Buka Box, maka sistem akan menampilkan jumlah *box* tugas yang dapat diakses dan memberikan *user* pilihan untuk memilih *box* tugas nomer berapa yang akan dipakai. Setelah *user* memilih *box* tugas yang akan dipakai, *user* perlu memasukkan data berupa mata kuliah *user* dan *deadline* pengumpulan tugas yang diinginkan oleh *user*. Data yang dimasukkan akan tersimpan pada *database* dan ditampilkan pada menu Status Box di layar LCD TFT. Kemudian pintu masuk tugas yang sebelumnya terkunci akan membuka selama waktu *deadline* pengumpulan tugas belum tercapai. Fitur kedua, yaitu buka *box*, merupakan fitur untuk membuka pintu *box* tugas saat *deadline* telah tercapai. Saat Buka Box digunakan, pada layar LCD TFT akan menampilkan Data Tugas Masuk mahasiswa yang telah mengumpulkan tugas.

Jika yang teridentifikasi oleh RFID *reader* adalah *admin*, maka sistem akan menampilkan menu Admin. Menu Admin memiliki 4 fitur, yaitu tambah user, hapus user, edit *box* dan buka *box*. Tambah *user* digunakan oleh *admin* saat ada *user* baru yang ingin memakai TaBox. Admin akan mengambil RFID *tag* yang baru dan memindai RFID *tag* tersebut sebagai *user* baru sehingga *user* baru dapat menggunakan TaBox. Kebalikan dari fitur tambah user, fitur hapus user digunakan untuk menghapus data *user* yang tidak ingin menggunakan TaBox lagi. *Admin* hanya perlu menekan tombol Hapus User dan memindai RFID *tag* milik *user* tersebut, sehingga sistem secara otomatis akan menghapus data *user* tersebut dalam daftar *user* TaBox.

Fitur edit *box* adalah fitur yang digunakan *admin* saat *user* ingin mengganti *deadline* pengumpulan tugas atau mengubah mata kuliah *user* yang salah dimasukkan oleh *user*. Fitur ini sama seperti fitur buka *box* pada menu User dimana *admin* harus memasukkan ulang data yang ingin diubah oleh *user*. Kemudian apabila menu User memiliki kendala tidak dapat membuka *box* tugas, maka pada menu Admin terdapat fitur Buka Box untuk membuka seluruh *box* tugas yang ada serta mendapatkan Data Tugas Masuk pada tiap *box*.

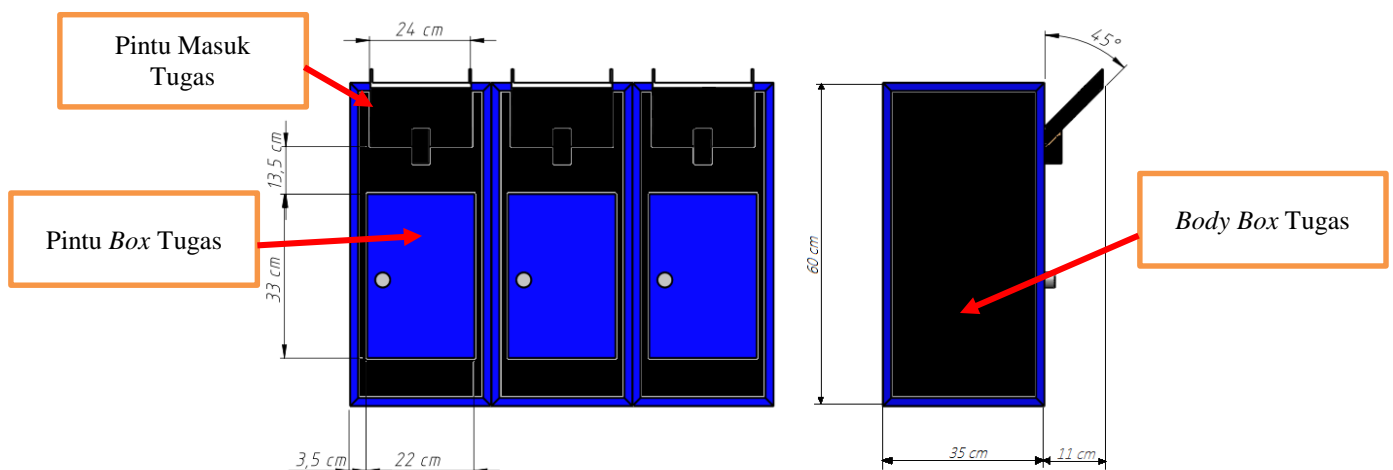
3.3 Rancangan *Hardware*

Rancangan *hardware* terdiri dari 2 *hardware* yang digabungkan jadi satu kesatuan. Dalam rancangan ini dibuatlah desain *hardware box* tugas dan *box* kontrol TaBox.

3.3.1 Rancangan *Hardware Box* Tugas

Pada tahap ini dibuat desain rancangan *hardware box* tugas untuk mensimulasikan sistem kontrol TaBox. Rancangan desain dibuat menggunakan *software* SolidWorks. Desain *hardware box* tugas berbentuk balok dengan ukuran panjang 30 cm, lebar 30 cm dan tinggi 60 cm yang memiliki pintu masuk tugas untuk memasukkan tugas dan sebuah pintu *box* tugas untuk pengambilan tugas saat tenggat waktu pengumpulan terpenuhi.

Dalam pembuatan rancangan *hardware*, dibuatlah sebuah gambar untuk menjadi acuan untuk pembuatan *box* tugas nanti. Berikut adalah gambar desain rancangan *box* tugas yang dapat dilihat pada Gambar 3.3.



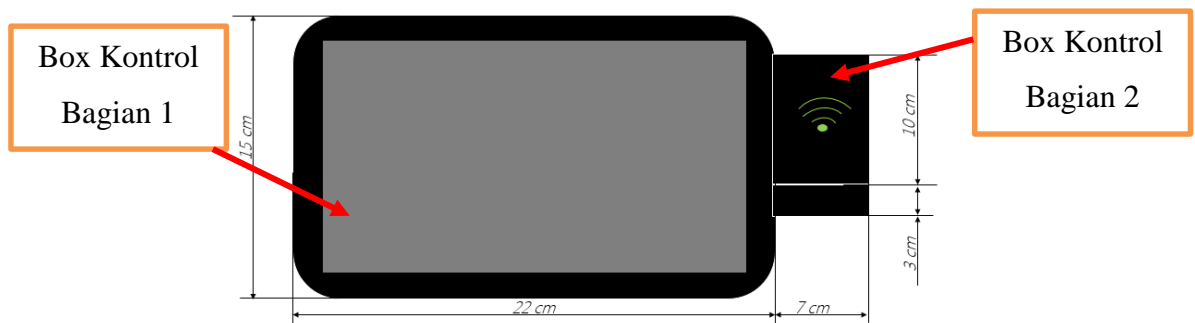
Gambar 3.3 Rancangan Gambar TaBox

Pada Gambar 3.3 diatas, pintu masuk tugas dirancang untuk memasukkan tugas berukuran A4 (210 x 297 mm) sehingga di desain dengan ukuran panjang 26 agar memberi ruang yang pas namun tidak pas-pasan saat memasukkan tugas. *Body box* tugas dirancang dengan ukuran 30 cm x 30 cm x 60 cm untuk memuat tugas

berukuran A4 (210 x 297 mm) dengan ketebalan maksimal 1.5 cm dan berjumlah 30 tugas. Sedangkan pintu *box* tugas dirancang sebagai pintu untuk membuka *box* tugas saat pengambilan tugas mahasiswa oleh *user*.

3.3.2 Rancangan *Hardware Box* Kontrol

Proses perancangan *hardware* pada *box* kontrol dibuat menggunakan *software* SolidWorks. *Hardware box* kontrol didesain dengan 2 kotak yang berukuran 22 cm x 15 cm x 7 cm dan 10 cm x 7 cm x 3 cm. Berikut merupakan desain *hardware box* kontrol yang ditunjukkan pada Gambar 3.4 berikut.



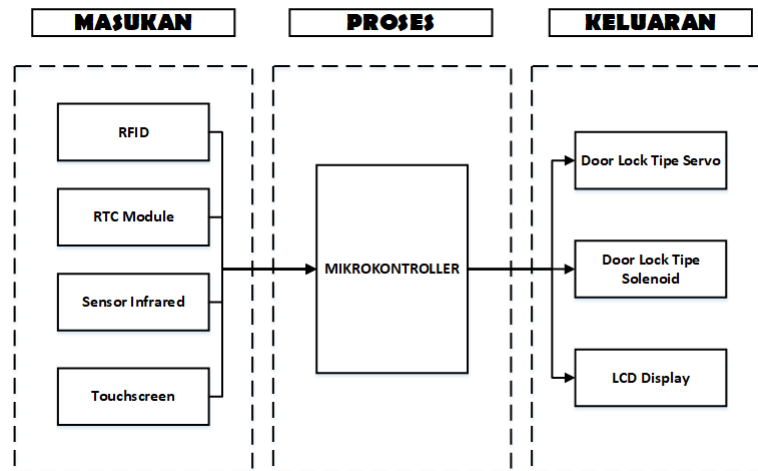
Gambar 3.4 *Hardware Box* Kontrol

Pada Gambar 3.4 *box* kontrol bagian 1 didesain untuk menempatkan komponen elektronika seperti LCD TFT, Arduino Mega 2560, RTC DS3231 rangkaian catu daya serta *pin-pin* yang terhubung pada komponen elektronika pada *box* komponen di *box* tugas dengan kabel sebagai media penghubung. *Box* kontrol bagian 2 didesain lebih kecil untuk menempatkan RFID *tag* dan Arduino Uno serta didesain dengan posisi agak miring untuk memudahkan *admin/user* saat akan memindai RFID *tag* nya..

3.4 Diagram Blok

Dalam pembuatan proyek akhir ini, sistem TaBox memiliki diagram blok untuk memudahkan penulis dalam merancang sistem yang akan dibuat. Serta mengetahui komponen apa saja yang dibutuhkan dan spesifikasi yang sesuai dengan diagram blok yang telah dibuat terlebih dahulu.

Berikut diagram blok sistem kontrol TaBox ditunjukkan pada Gambar 3.5 dibawah ini.



Gambar 3.5 Diagram Blok Sistem Kontrol TaBox

Arduino digunakan sebagai pengolah data, penyimpanan data dari RTC dan RFID serta sebagai pengontrol pergerakan servo dan solenoid. RFID *reader* untuk membaca RFID *tag*. RTC berfungsi untuk pencatatan waktu. LCD *Display* sebagai penampil data yang disimpan Arduino. Servo sebagai buka tutup pintu masuk tugas pada TaBox dan *solenoid* untuk pengunci pintu *box* tugas.

BAB IV

PEMBAHASAN

Pada bab ini akan membahas proses pembuatan TaBox (*Task Box*) dan pengujian alat serta metode yang dipakai dalam pembuatan proyek akhir yang berjudul “Rancang Bangun TaBox (*Task Box*) Sebagai Wadah Pengumpul Tugas Mahasiswa”, yaitu sebagai berikut:

4.1 Pembuatan *Hardware Box* Tugas dan *Box* Kontrol

Pembuatan *Hardware* terdiri dari 2 *hardware* yang dijadikan satu kesatuan. *Hardware* yang dibuat adalah *box* tugas dan *box* kontrol.

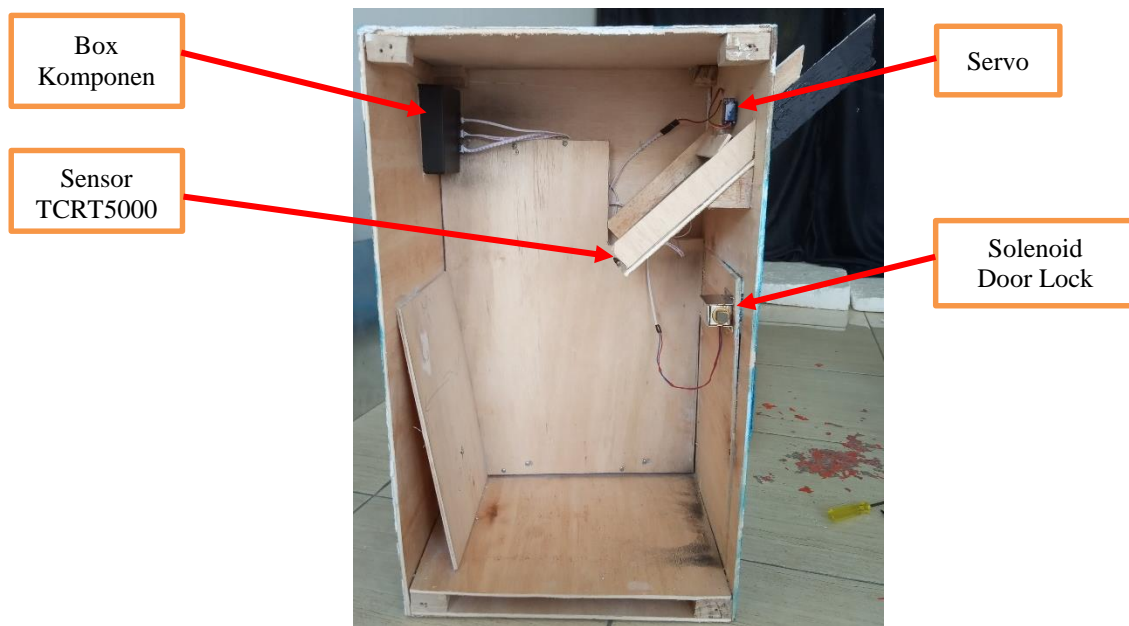
4.1.1 Pembuatan *Hardware Box* Tugas

Box tugas terbuat dari triplek yang berfungsi sebagai pembentuk *body box* tugas sekaligus kerangka utama pembentuk *box* tugas. Triplek dengan ketebalan 6 mm dipotong dengan menggunakan gergaji sesuai dengan ukuran kerangka utama *box* tugas, yaitu panjang 30 cm, lebar 30 cm dan tinggi 60 cm. Berikut merupakan hasil pembuatan *box* tugas yang dapat dilihat pada Gambar 4.1 dibawah ini.



Gambar 4.1 Hasil Pembuatan *hardware Box* Tugas

Pada *box* tugas, servo berfungsi sebagai pembuka dan penutup pintu masuk tugas sesuai dengan waktu yang telah di atur oleh *user*, TCRT 5000 sebagai sensor yang mendeteksi ada atau tidaknya tugas yang masuk melewati slot TaBox, dan relai untuk menggerakkan solenoid membuka ataupun mengunci pintu TaBox. Peletakkan komponen tersebut dapat dilihat pada Gambar 4.2 dibawah ini.



Gambar 4.2 Peletakkan Komponen di *Box* Tugas

Pada Gambar 4.2 menunjukkan peletakkan komponen elektronika dan *box* komponen. *Box* komponen digunakan untuk menempatkan komponen elektronika seperti relai dan pin komponen elektronika yang ada di *box* tugas. *Box* komponen dibuat dengan memodifikasi sebuah *junction box* tipe Duradus MG polos persegi panjang yang sudah jadi dengan ukuran 3 cm x 4 cm x 4 cm. *Box* komponen berjumlah 3 buah, dimana *box* tersebut diletakkan pada masing-masing *box* tugas dan terhubung dengan sensor TCRT5000, servo pintu masuk tugas dan *solenoid door lock* pada pintu *box* tugas.

4.1.2 Pembuatan *Hardware Box Kontrol*

Box kontrol berfungsi sebagai tempat penyimpanan komponen elektronik yang digunakan untuk mengontrol kerja TaBox, yaitu Arduino Mega dan Uno, LCD TFT RA8875, RFID *reader* dan RTC DS3231. *Box* kontrol terbuat dari 2 *junction box* tipe Duradus MG Polos persegi panjang dengan ukuran panjang 22 cm, lebar 15 cm, tinggi 7 cm dan ukuran panjang 10 cm, lebar 7.6 cm dan tinggi 3.3 cm yang hasil pembuatannya dapat dilihat pada Gambar 4.3 berikut.



Gambar 4.3 *Box* Kontrol TaBox

Box kontrol 1 merupakan *box* kontrol utama yang berisi LCD TFT RA8875, Arduino Mega 2560, RTC DS3231 dan rangkaian catu daya. *Box* ini dibuat dengan menyesuaikan ukuran pada LCD TFT RA8875. Kemudian untuk *box* kontrol 2 ini berisikan Arduino Uno dan RFID *reader*. *Box* kontrol 2 dibuat dengan posisi sedikit miring agar memudahkan *user* saat proses pemindaian RFID tag nya.

4.2 Wiring Diagram Alat

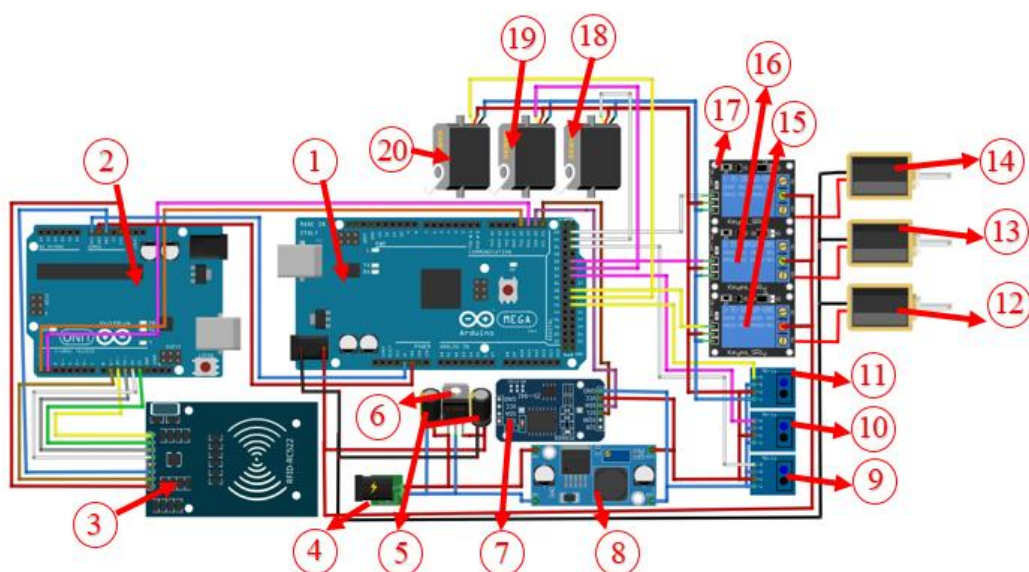
Diagram pengkabelan atau *wiring diagram* adalah gambar kerja/gambar diagram sederhana yang menggambarkan rangkaian pengkabelan atau pengkawatan peralatan elektronik. *Wiring diagram* ibarat peta yang menunjukkan fungsi dari suatu peralatan elektronik dan komponen penyusunnya yang saling tersambung sebagai satu rangkaian elektronik serta menunjukkan aliran arus pada

rangkaian elektronik. *Wiring diagram* diperlukan sebagai penuntun penulis dalam tahap perakitan yaitu penyatuan kedua *hardware*. Berikut adalah *interface wiring diagram* sistem kontrol TaBox bagian LCD TFT RA8875 yang dapat dilihat pada Gambar 4.4 dibawah ini.



Gambar 4.4 *Interface* LCD TFT dengan Arduino Mega 2560

Interface antara LCD TFT RA8875 dan Arduino Mega 2560 menggunakan *shield* LCD TFT RA8875 sebagai modul penghubung antara kedua komponen tersebut. Hal ini memudahkan penulis dalam menghubungkan LCD TFT RA8875 dan Arduino Mega 2560 tanpa harus melakukan pengkabelan secara manual. Kemudian untuk *wiring diagram* keseluruhan sistem kontrol TaBox ditunjukkan pada Gambar 4.5 berikut.



Gambar 4.5 *Wiring Diagram* Keseluruhan Sistem TaBox

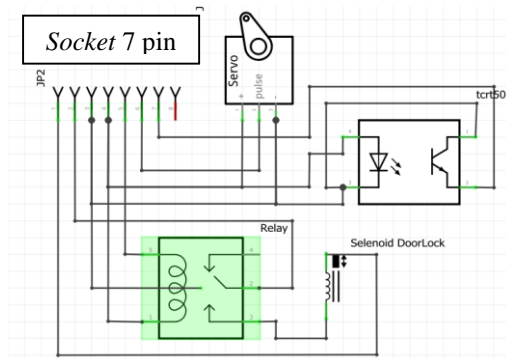
Berikut merupakan keterangan pada Gambar 4.5:

1. Arduino Mega 2560
2. Arduino Uno
3. RFID RC522
4. *Barrel* Adaptor Female
5. Kapasitor 470 μ F 16V
6. IC Regulator LM7809
7. RTC DS3231
8. Modul Step Down LM295S
9. Sensor TCRT5000 *Box* 00
10. Sensor TCRT5000 *Box* 01
11. Sensor TCRT5000 *Box* 02
12. Solenoid *Box* 02
13. Solenoid *Box* 01
14. Solenoid *Box* 00
15. Relai *Box* 02
16. Relai *Box* 01
17. Relai *Box* 00
18. Servo *Box* 00
19. Servo *Box* 01
20. Servo *Box* 02

Pada *wiring diagram* Gambar 4.5 dapat dijelaskan, saat *box* kontrol mendapatkan sumber tegangan 12 volt dari Adaptor 3A/12V. Tegangan tersebut masuk ke *barrel jack* lalu mengalir ke dua buah stepdown (stepdown 12V dan 9V), kemudian tegangan akan melewati kapasitor pertama dan selanjutnya melewati LM7809 yang keluarannya dibatasi oleh kapasitor kedua. Setelah itu arus mengalir menuju *fan*, *solenoid door lock* dan Arduino Mega 2560. Arduino Mega 2560 yang telah aktif akan memberikan masukan tegangan untuk LCD TFT RA8875 dan Arduino Uno. Saat Arduino Uno aktif maka akan ikut mengaktifkan RFID *reader*. Untuk menurunkan tegangan ke *socket* 5 volt maka LM298 yang akan menurunkannya secara langsung dari 12 volt ke 5 volt, keluaran 5 volt kemudian membuat RTC DS3231, servo, sensor TCRT5000 dan relai dalam kondisi siap digunakan. Komponen tersebut akan aktif dan menjalankan intruksi saat mendapat sinyal sesuai dengan kondisi yang terprogram.

Pada *box* tugas terdapat *box* komponen yang di lakukan pengkabelan sesuai dengan skematik yang telah dibuat oleh penulis. Dalam skematik tersebut digambarkan jalur pin yang terhubung dari *socket* 7 pin ke masing-masing komponen elektronika yang ada.

Berikut merupakan skematik *box* komponen yang ditunjukkan pada Gambar 4.6 berikut.

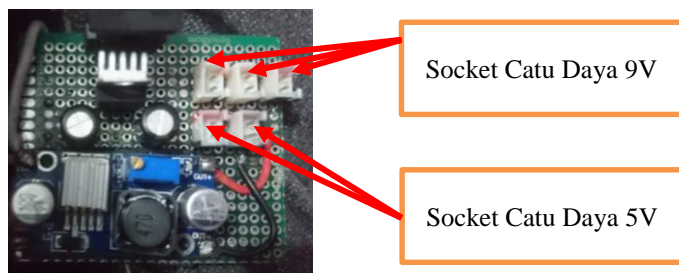


Gambar 4.6 Skematik Rangkaian *Box* Komponen

Dari Gambar 4.6 dapat dilihat bahwa komponen elektronika yang ada di *box* tugas dipisahkan oleh sebuah *socket*. Dimana *socket* tersebut di tempatkan pada *box* komponen dalam *box* tugas. *Box* komponen yang berisikan relai serta servo, *solenoid door lock* dan sensor TCRT5000 mendapatkan sumber yang sama dengan *box* kontrol, yaitu rangkaian catu daya dari *socket* berukuran 9 volt.

4.3 Pengujian Perangkat Catu Daya

Dalam rangkaian catu daya digunakan 1 buah IC regulator LM 7809, 1 buah transistor TIP3055 dan modul *step-down* LS289S. Dalam pengujian modul catu daya yang digunakan untuk beberapa rangkaian yang menggunakan tegangan 5 volt dan 9 volt. Catu daya yang digunakan adalah Adaptor 12V/ 3A sebagai sumber utama rangkaian catu daya. Pada Gambar 4.7 dapat dilihat rangkaian catu daya yang telah di rangkai oleh penulis.



Gambar 4.7 Rangkaian Catu Daya

Penggunaan catu daya 5 volt digunakan untuk RTC, sensor TCRT5000, servo dan relai yang ada pada *box* komponen pada masing-masing *box* tugas. Kemudian untuk penggunaan catu daya 9 volt digunakan untuk Arduino Mega 2560 dan *solenoid door lock* pada setiap *box* tugas. Berikut merupakan hasil pengujian pengukuran rangkaian catu daya yang dapat dilihat pada Tabel 4.1.

Tabel 4.1 Hasil Pengujian Rangkaian Penurun Tegangan

Pengukuran ke-	Tegangan Pada Catu Daya 5 volt	Tegangan Pada Catu Daya 9 volt
1	4.97 V	8.86 V
2	5.02 V	9 V
3	5 V	9.02 V
4	5.01 V	9.23
5	5 V	9 V

Dari beberapa pengukuran pada rangkaian penurun tegangan dapat dilihat adanya nilai selisih antara tegangan terukur dan tegangan catu daya. Kemudian dilanjutkan dengan rumus untuk mencari perhitungan % *error* diambil dari sampel pengukuran ke-1 tegangan pada catu daya 5 volt adalah:

$$\%error = \frac{|Tegangan\ Catu\ Daya - Tegangan\ yang\ Terukur|}{Tegangan\ Catu\ Daya} \times 100\% \dots\dots\dots (4.1)$$

$$\%error = \frac{|5\ V - 4.97\ V|}{5\ V} \times 100\%$$

$$\%error = 0.006\%$$

Adanya perbedaan pengukuran tegangan pada penurun tegangan 5 volt disebabkan oleh kurang stabilnya tegangan masukan dari Adaptor 12V/3A. Tegangan pada rangkaian catu daya dapat digunakan, karena tegangan masih belum melewati batas tegangan pada komponen yang diperlukan. Dari hasil pengujian dapat disimpulkan bahwa rangkaian catu daya dapat digunakan untuk mensuplai beberapa komponen dan modul yang digunakan untuk sistem TaBox.

4.4 Pengujian Sensor TCRT5000

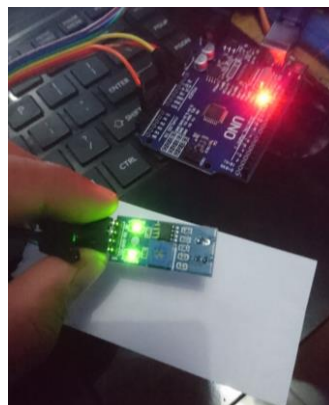
Sensor ini digunakan untuk mendeteksi ada atau tidaknya obyek yang berada pada jarak deteksi sensor. Pada tahap pengujian ini sensor yang dipakai bertipe sensor TCRT5000. Pengujian sensor dilakukan dengan cara meletakkan suatu obyek di depan sensor. Sensor ini kemudian dihubungkan dan diprogram menggunakan *board* Arduino Mega 2560. Berikut adalah *sketch/source code* program pengujian pada sensor TCRT5000.

```
void loop()
{
  if ( digitalRead (sensor) == LOW)
  {
    digitalWrite (LED, HIGH);
    Serial.println ("Tugas Masuk");
    delay(3000);}
  else {
    digitalWrite (LED, LOW);}}
```

} Pembacaan Sensor Apabila Sensor Mendeteksi Obyek

} Pembacaan Sensor Apabila Sensor Tidak Mendeteksi Obyek

Dengan menggunakan *coding* program yang telah dibuat didapatkan data berupa deteksi tugas yang masuk. *Output* yang digunakan sensor adalah *output* digital jadi apabila sensor mendeteksi suatu obyek yang dapat merefleksikan cahaya maka akan bernilai LOW atau 0 dan 2 lampu LED kuning akan menyala serta serial monitor akan menampilkan tulisan tugas masuk yang dapat dilihat pada Gambar 4.8 berikut.



Gambar 4.8 Saat Sensor Tidak Mendeteksi Obyek

Sebaliknya, jika sensor tidak mendeteksi adanya suatu obyek yang dapat merefleksikan cahaya, maka akan bernilai 1 dan hanya ada 1 LED kuning yang menyala. Kondisi ini dapat dilihat pada Gambar 4.9 dibawah ini.

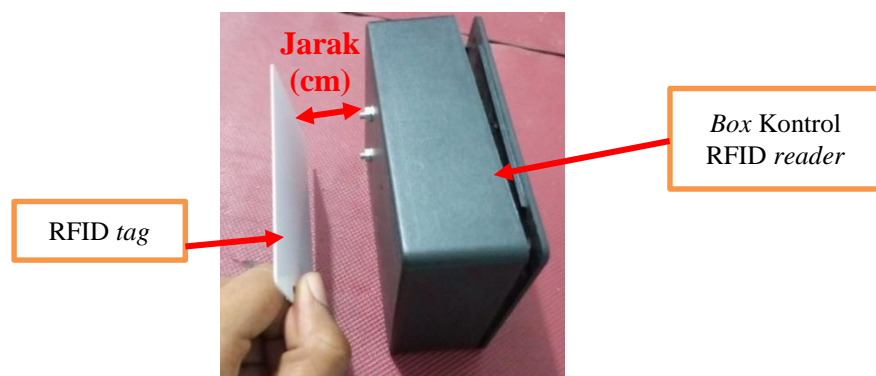


Gambar 4.9 Saat Sensor Mendeteksi Obyek

Pada pengujian ini dapat disimpulkan bahwa sensor dapat bekerja dengan baik sesuai dengan yang diinginkan oleh penulis.

4.5 Pengujian RFID RC522

Pengujian ini bertujuan untuk mengetahui berapa jarak pembacaan kartu RFID *tag* yang dapat dilakukan oleh RFID *reader*. Pengujian dilakukan dengan mendekatkan kartu RFID *tag* ke RFID *reader* dengan jarak tertentu dan kemudian diukur oleh penggaris. Dalam pengujian ini posisi RFID *tag* diletakkan di depan dan belakang RFID *reader*. Metode yang digunakan untuk melakukan pengujian dapat dilihat pada Gambar 4.10 berikut.



Gambar 4.10 Metode Pengambilan Data Jarak Pembacaan RFID *Reader*

Dengan pengujian tersebut, didapatkan hasil uji coba jarak pembacaan RFID *reader* yang dapat dilihat pada Tabel 4.2 dibawah ini.

Tabel 4.2 Hasil Pengujian Jarak Pembacaan RFID *Reader*

Jarak Baca (cm)	Posisi RFID <i>Tag</i>	
	Tanpa Halangan	Dengan Halangan
1	Terbaca	Terbaca
2	Terbaca	Tidak Terbaca
3	Terbaca	Tidak Terbaca
4	Tidak Terbaca	Tidak Terbaca
5	Tidak Terbaca	Tidak Terbaca

Pembacaan tersebut dilakukan tanpa adanya penghalang diantara RFID *tag* dengan *reader*. Dari data yang terdapat pada Tabel 4.2 tersebut, dapat disimpulkan bahwa jarak maksimal untuk RFID *reader* membaca data dari kartu RFID *tag* tanpa halangan adalah 3 cm dan dengan halangan adalah 1 cm. Hal tersebut terjadi karena adanya kemungkinan bahwa daya yang dipancarkan oleh RFID *reader* bernilai rendah. Kemudian sebagai indikator pada RFID *reader* digunakan sebuah LED yang aktif jika RFID *reader* mendeteksi kartu RFID *tag*. LED indikator berfungsi sebagai penanda jika RFID *reader* melakukan komunikasi dengan RFID *tag*. Kondisi tersebut dapat dilihat pada Tabel 4.3 berikut.

Tabel 4.3 Kondisi RFID *Reader* Terhadap RFID *Tag*

Kondisi RFID <i>Tag</i>	Kondisi RFID <i>Reader</i> (LED)
1	Aktif (Menyala)
2	Tidak Aktif (Mati)

Setelah itu dilakukan juga pengujian untuk memperoleh identitas dari RFID *tag* tersebut. Pengujian dilakukan dengan mendekatkan RFID *tag* ke RFID *reader*, lalu hasil pembacaan akan ditampilkan pada serial monitor di *software*

Arduino IDE. Dengan meng-*upload sketch/source code* pada Arduino Uno, kita dapat memperoleh identitas dari RFID Tag tersebut. Berikut merupakan *sketch/source code* pembacaan nomer ID pada RFID Tag.

```
void loop() {
  Serial.print(F("PICC type: "));
  MFRC522::PICC_Type piccType=rfid.PICC_GetType(rfid.uid.sak);
  Serial.println(rfid.PICC_GetTypeName(piccType));

  if (rfid.uid.uidByte[0] != nuidPICC[0] ||
  rfid.uid.uidByte[1] != nuidPICC[1] ||
  rfid.uid.uidByte[2] != nuidPICC[2] ||
  rfid.uid.uidByte[3] != nuidPICC[3] ) {
    Serial.println(F("Kartu Baru akan segera dipindai."));

    for (byte i = 0; i < 4; i++) {
      nuidPICC[i] = rfid.uid.uidByte[i];}

    Serial.println();
    Serial.print(F("No. ID: "));
    printDec(rfid.uid.uidByte, rfid.uid.size);
    Serial.println();}
}
```

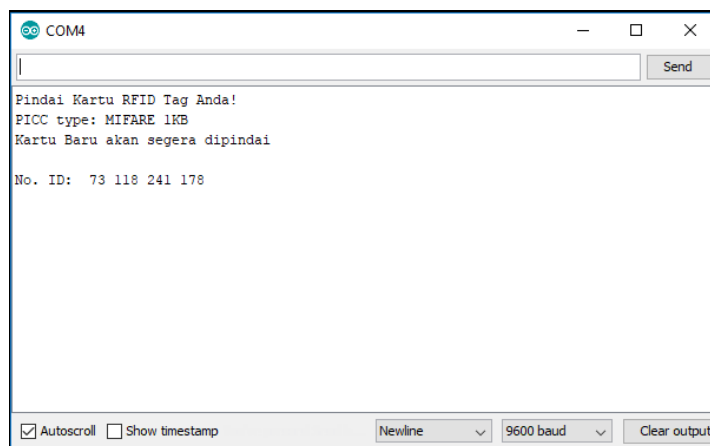
Pembacaan Tipe RFID Tag

Pembacaan Setiap ID

Pengelompokkan setiap ID yang terbaca

Menampilkan Nomer ID RFID Tag pada Serial Monitor

Dari program tersebut, didapatkan hasil pembacaan RFID tag yang ditampilkan pada serial monitor. Hasil tersebut dapat dilihat pada Gambar 4.11 berikut.

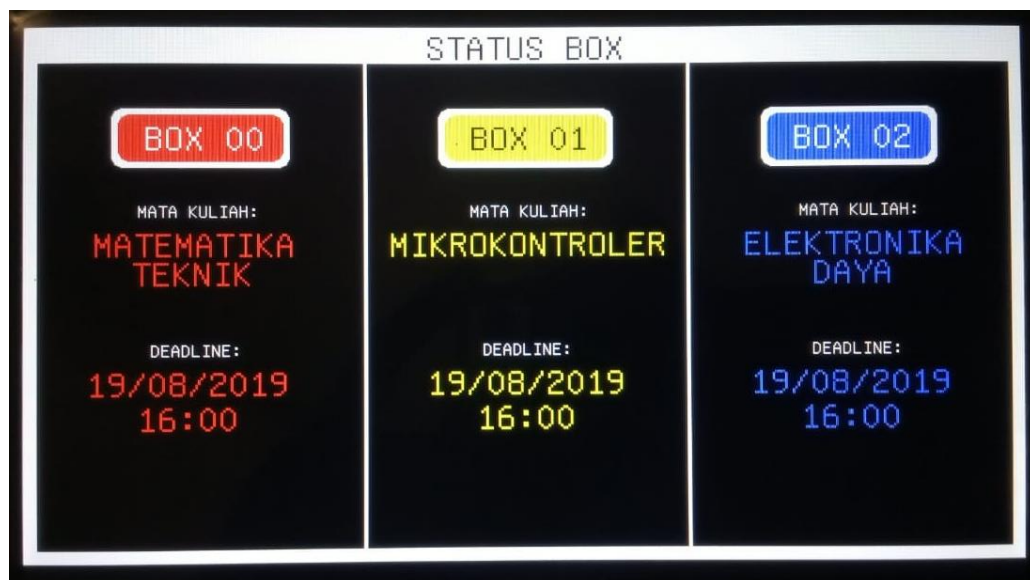


Gambar 4.11 Serial Monitor Pembacaan ID RFID Tag

Pada Gambar 4.11 tersebut, serial monitor menampilkan data berupa tipe RFID *tag* dan nomer ID RFID *tag* dalam bentuk bilangan desimal. Kemudian data ini akan diolah nantinya untuk di kelompokkan menjadi dua jenis, yaitu *admin* dan *user*.

4.6 Menu pada LCD TFT RA8875

Sistem kontrol TaBox menggunakan sistem *user* untuk mempermudah pengguna dan sistem *admin* untuk mengatur serta mengontrol penuh sistem dan semua hak akses *user* yang boleh masuk ke dalam sistem. Pada tampilan menu utama LCD TFT akan menampilkan informasi mengenai TaBox yang digunakan oleh *user* (dalam hal ini ialah dosen/pengajar), berupa nomer *box*, mata kuliah dan *deadline* pengumpulan tugas. Berikut merupakan tampilan menu utama TaBox, yaitu menu Status Box yang dapat dilihat pada Gambar 4.12 dibawah ini.



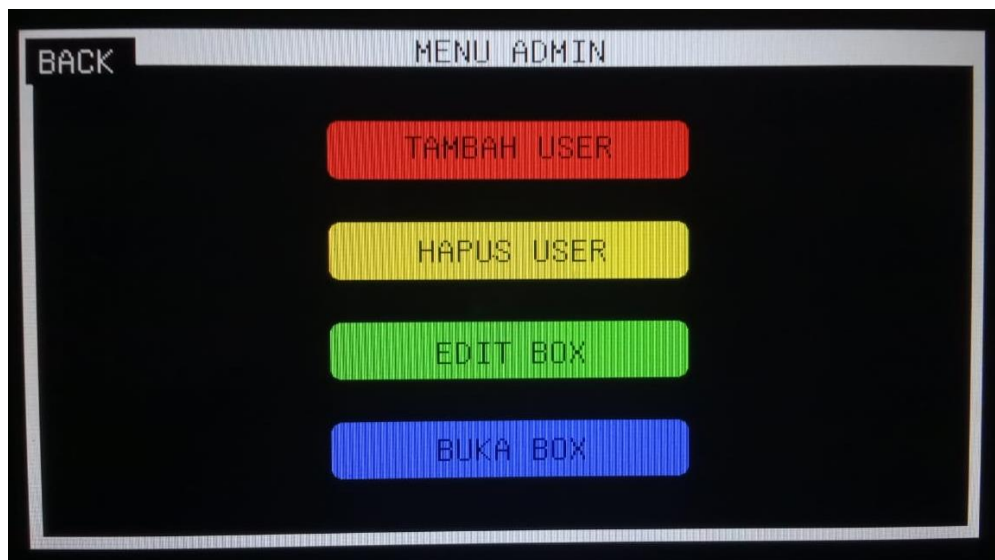
Gambar 4.12 Tampilan Menu Status Box

Tampilan menu Status Box tidak akan berganti tampilan kecuali apabila ada *admin* atau *user* yang *login* ke dalam sistem menggunakan RFID *tag*. RFID *tag* sebagai kartu akses memiliki dua jenis kartu, yaitu *admin* dan *user*, sehingga pada LCD TFT memiliki 2 fitur menu utama yaitu menu Admin dan menu User. Untuk

login ke dalam sistem TaBox, *admin* atau *user* perlu mendekatkan RFID *tag* nya ke RFID *reader*. Apabila RFID *reader* telah mengidentifikasi RFID *tag* tersebut merupakan *admin* atau *user*, sistem secara otomatis akan membuka menu Admin apabila RFID *tag* teridentifikasi sebagai *admin* dan membuka menu User apabila RFID *tag* teridentifikasi sebagai *user*.

4.6.1 Menu Admin

Sistem *admin* merupakan sistem yang digunakan untuk memelihara, mengelola dan mengoperasikan sistem komputer atau jaringan. Dalam hal ini, sistem admin atau disebut menu Admin digunakan untuk mengatur serta mengontrol penuh sistem dan semua hak akses *user* yang boleh masuk ke dalam sistem. Admin saat ini merupakan penulis. Gambar 4.13 dibawah ini merupakan tampilan menu Admin beserta fitur yang ada.

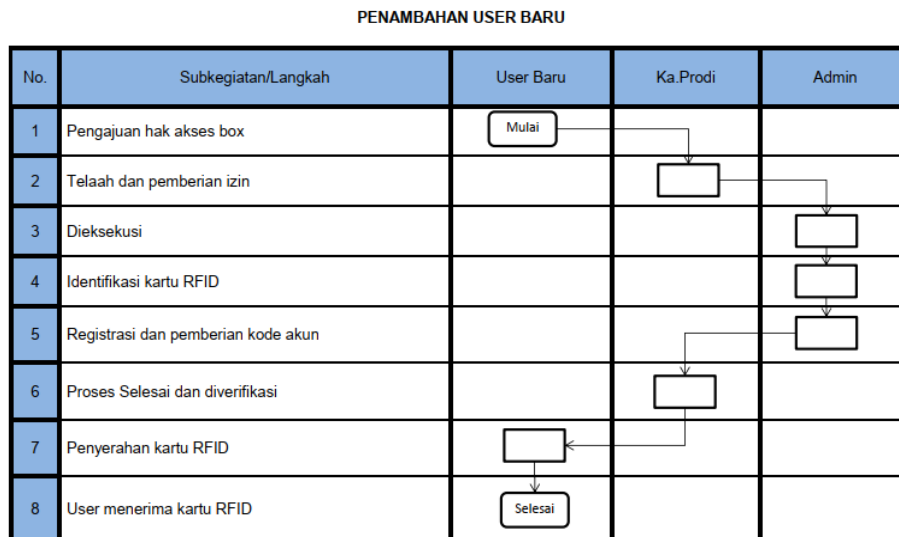


Gambar 4.13 Tampilan Menu *Admin*

Menu *admin* memiliki fungsi untuk memberi hak akses *user* untuk menggunakan TaBox. *Admin* dapat menambahkan dan menghapus *user* serta memiliki hak akses dalam membuka dan mengedit semua *box* tugas yang ada. Menu Admin hanya dapat diakses oleh *admin* melalui kartu yang telah diidentifikasi sebagai kartu *admin*. Berikut merupakan fitur yang dimiliki oleh menu Admin:

A. Tambah User

Fitur tambah *user* memungkinkan admin dapat menambahkan *user* baru yang telah mendapatkan ijin penggunaan hak akses oleh ketua Prodi. Saat *admin login* dan menekan tombol Tambah User, tombol tersebut akan mengaktifkan program penambahan *user* baru. Kemudian *admin* hanya perlu mendekatkan RFID *tag user* baru ke RFID *reader* sehingga RFID *tag* yang baru sudah dapat digunakan. Prosedur penambahan *user* baru tersebut dapat dilihat pada Gambar 4.14 dibawah ini.



Gambar 4.14 *Flowchart* Sistem Admin Untuk Penambahan User Baru

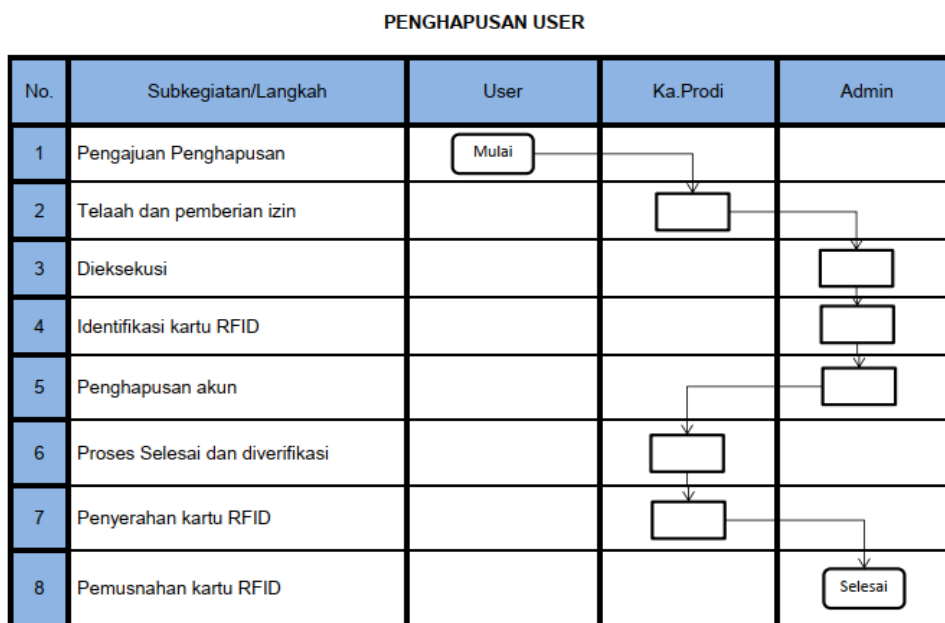
Gambar 4.14 menjelaskan tentang interaksi antara *user* dengan sistem *admin*, dimana sebelum *user login* maka *user* baru harus mengajukan hak akses yang kemudian perizinan akan diberikan oleh Ketua Prodi dan selanjutnya *admin* akan memasukkan terlebih dahulu data *user*, adapun data *user* di *input* oleh *admin* setelah *user* mendapatkan izin atas hak akses penggunaan TaBox.

B. Hapus User

Hapus user merupakan fitur milik menu admin yang memberikan *admin* wewenang untuk menonaktifkan hak akses *user*. Hapus user dilakukan *admin*

dengan cara menekan tombol Hapus User setelah *login*, kemudian *admin* hanya perlu mendekatkan RFID *tag user* yang akan di nonaktifkan, sehingga mengaktifkan program penghapusan *user*. Setelah itu, data *user* akan terhapus secara otomatis dari EEPROM.

Prosedur penghapusan *user* tersebut dapat dilihat pada Gambar 4.15 dibawah ini.



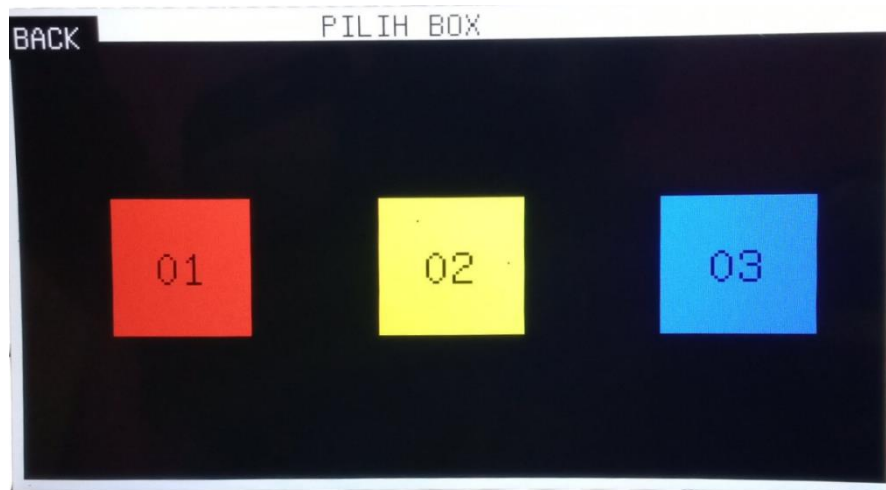
Gambar 4.15 *Flowchart* Sistem *Admin* Untuk Penghapusan *User*

Gambar 4.15 menjelaskan tentang interaksi antara *user* dengan sistem *admin*, dimana sebelum *user* menon aktifkan hak aksesnya pada EEPROM yang ada di sistem maka *user* harus mengajukan izin penghapusan data yang kemudian perizinan tersebut akan diberikan oleh Ketua Prodi dan selanjutnya *admin* akan menghapus data *user*, setelah data *user* dihapus, kartu RFID milik *user* akan dimusnahkan oleh *admin*.

C. Edit Box

Edit *box* adalah fitur untuk *admin* merubah *inputan* data yang salah saat di *input* oleh *user*. *User* dapat menghubungi *admin* apabila ingin melakukan perubahan pada *input* data. Saat *admin login* dan menekan tombol Edit Box, maka

admin akan memasuki menu untuk memilih *box* tugas nomer berapa yang akan dilakukan pengeditan. Pada menu ini, terdapat 3 *box* yang dapat di edit oleh *admin*. Tampilan menu Pilih Box dapat dilihat pada Gambar 4.16 dibawah ini.



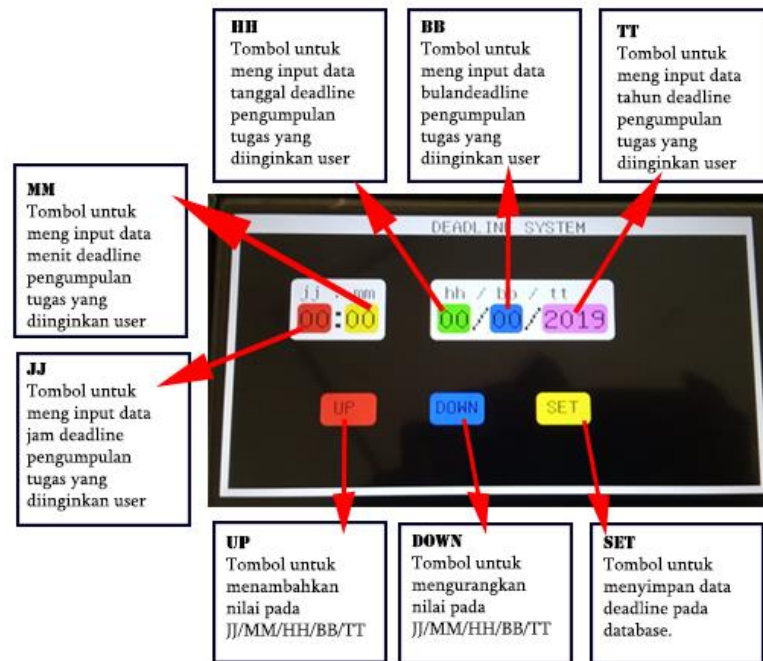
Gambar 4.16 Tampilan Fitur Pilih *Box* pada Menu Admin

Setelah *box* dipilih, maka LCD TFT akan menampilkan *input* data mata kuliah milik *user*. Dimana saat *admin* meng *input* data mata kuliah yang baru dan menekan tombol Enter, data yang sebelumnya dimasuki *user* akan terhapus dan diganti oleh data yang telah di edit oleh *admin*. Berikut merupakan tampilan menu saat *admin* akan memasukkan kembali data mata kuliah *user* terdapat pada Gambar 4.17 berikut.



Gambar 4.17 Tampilan Menu Input Mata Kuliah pada Menu Admin

Selanjutnya LCD TFT akan menampilkan menu Deadline System pengumpulan tugas. Menu ini memiliki 5 data yang harus dimasukkan oleh *admin*, yaitu data jam, menit, tanggal, bulan dan tahun. Dibawah ini adalah tampilan menu Deadline System yang ditunjukkan pada Gambar 4.18 berikut.



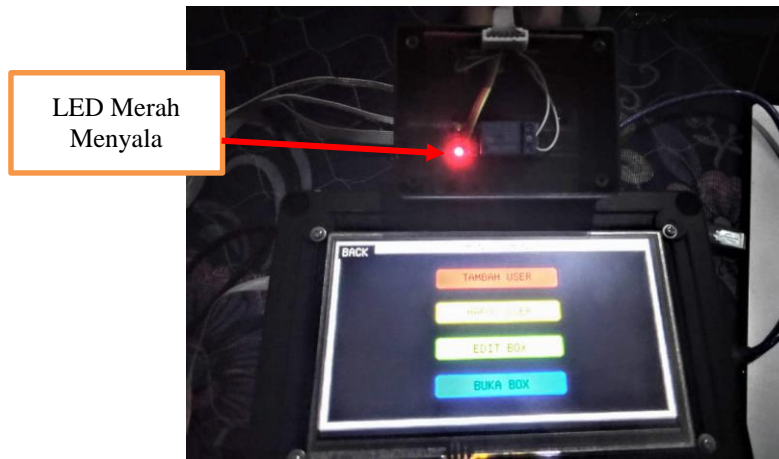
Gambar 4.18 Tampilan Menu Deadline System pada Menu Admin

Setelah data *deadline* pengumpulan tugas di *input*, *admin* hanya perlu menekan tombol SET sehingga data yang di ubah oleh *admin* telah tersimpan di EEPROM sistem TaBox. Nantinya data tersebut akan ditampilkan kembali di menu Status Box LCD TFT.

D. Buka Box

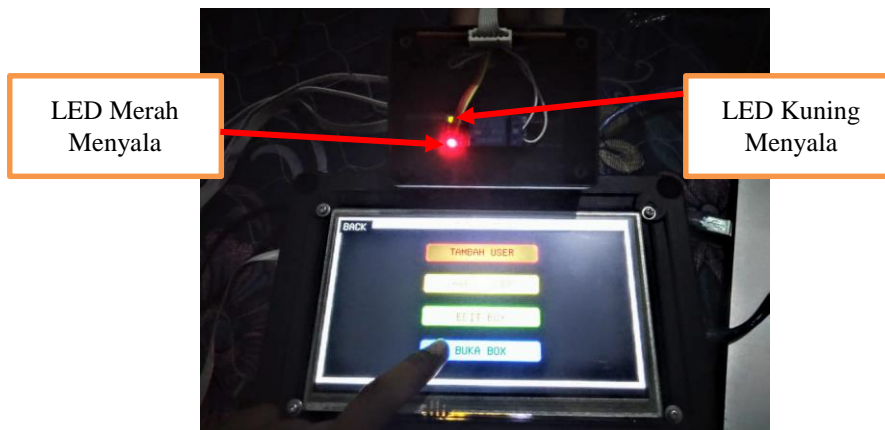
Fitur buka box digunakan *admin* saat *user* memiliki kondisi yang tidak memungkinkan untuk membuka TaBox yang *user* gunakan, sehingga mengharuskan *admin* membuka TaBox tersebut. *Admin* dapat membuka semua *box* tugas yang ada yaitu, 3 buah *box*. Saat *admin* menekan tombol Buka Box, maka akan tersedia pilihan untuk membuka *box* nomer berapa untuk dibuka. Pada

Gambar 4.19 dibawah ini menunjukkan kondisi relai sebelum intruksi buka *box* ditekan.



Gambar 4.19 Kondisi Relai Sebelum Tombol Buka Box Ditekan

Setelah menekan tombol nomer *box* tugas yang akan dibuka, maka indikasi relai yang sebelumnya LED relai hanya berwarna merah akan menghidupkan LED warna merah dan kuning. Kondisi ini dapat dilihat pada Gambar 4.20 berikut.

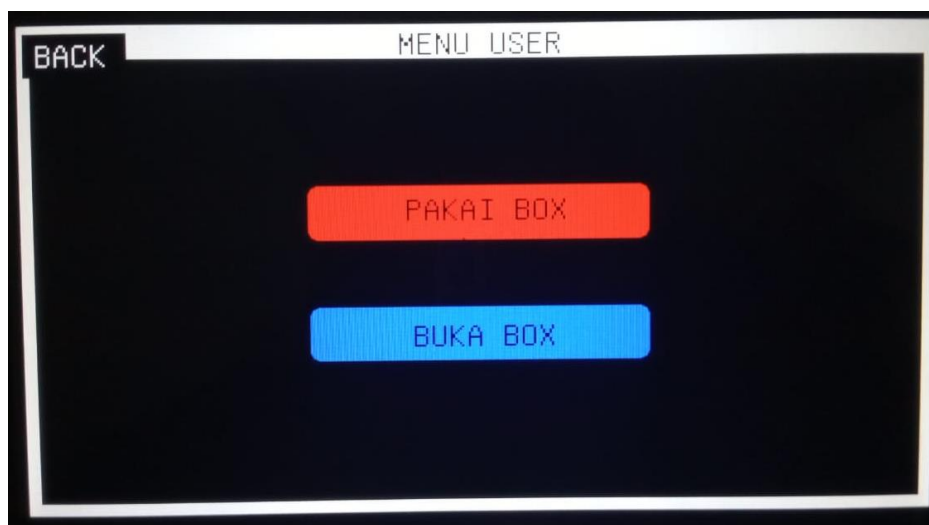


Gambar 4.20 Kondisi Relai Setelah Tombol Buka Box Ditekan

Setelah Relai aktif, relai akan menggerakkan *solenoid* untuk membuka pintu *box* tugas. Kemudian pindai RFID Tag kembali untuk mengunci pintu box tugas yang digunakan sebelumnya.

4.6.2 Menu User

Sistem *user* atau menu User merupakan sistem yang digunakan oleh *user* (pengguna) untuk menangkap, mengvalidasi, memasukkan, merespon, menyimpan serta melakukan pertukaran data dan informasi. Dalam hal ini, *user* merupakan dosen, pengajar ataupun *staff* Politeknik Manufaktur Negeri Bangka Belitung yang telah terdaftar dalam sistem TaBox. Menu User ditunjukkan pada Gambar 4.21 dibawah ini.



Gambar 4.21 Tampilan Menu User

Fitur pada menu User berfungsi untuk memungkinkan *user* mengakses TaBox yang akan digunakan. *User* dapat memilih TaBox yang akan digunakan dan membuka TaBox saat *deadline* pengumpulan tugas telah tercapai. *User* sebagai pengguna memiliki perbedaan utama dengan *admin* yaitu, *user* hanya dapat mengakses satu *box* tugas dan hanya dapat *login* ke *box* yang telah dipilih.

A. Pakai Box

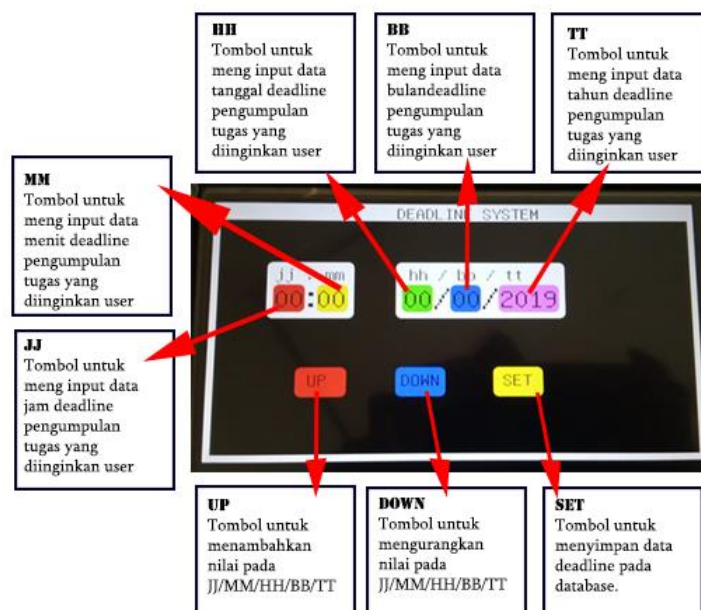
Pakai *box* merupakan fitur yang disediakan untuk *user* memilih *box* nomer berapa yang akan digunakan. *User* hanya dapat memilih satu *box* untuk digunakan. Terdapat 3 pilihan *box* yang telah penulis sediakan untuk dipilih oleh *user*. Saat *user login* dan menekan tombol Pilih Box, maka selanjutnya *user* perlu memasukkan data yang dibutuhkan berupa mata kuliah dan deadline pengumpulan tugas.

Berikut merupakan tampilan menu Masukkan Mata Kuliah *user* yang ditunjukkan pada Gambar 4.22 berikut.



Gambar 4.22 Tampilan Menu Input Mata Kuliah User

User perlu mengetikkan mata kuliah *user*, hasil ketikkan akan muncul pada *input data*. Kemudian *user* menekan tombol Enter untuk menyimpan data mata kuliah ke EEPROM sistem TaBox. Selanjutnya *user* akan dihadapkan ke menu Deadline System, dimana *user* harus memasukkan data waktu *deadline* pengumpulan tugas sesuai dengan yang diinginkan *user*. Tampilan menu Deadline System ditunjukkan pada Gambar 4.23 berikut.

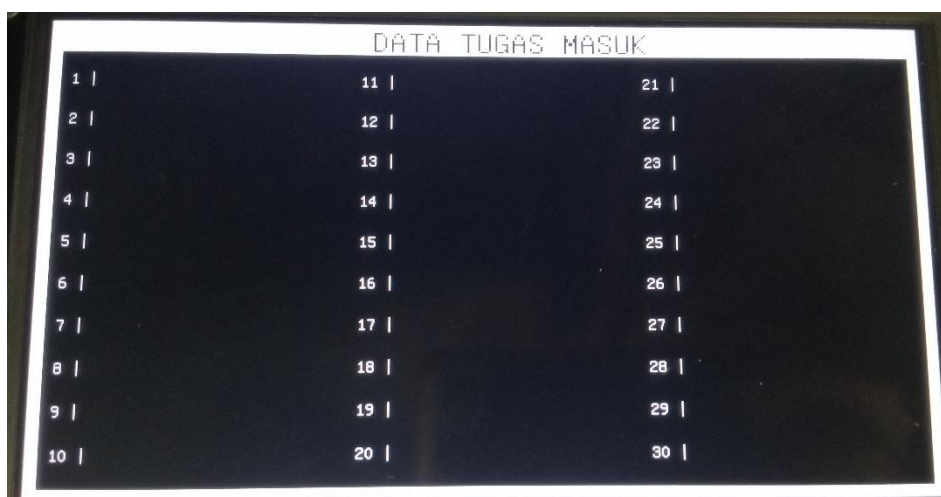


Gambar 4.23 Tampilan Menu Deadline System

Gambar 4.23 menunjukkan tampilan menu Deadline System. Inputan data dilakukan dengan menekan tombol JJ/MM/HH/BB/TT lalu *user* menekan tombol Up atau Down untuk menaikkan atau menurunkan nilai *input* yang diinginkan. Apabila *user* telah menginput data yang diinginkan. *User* harus menekan tombol SET untuk menyimpan data *deadline* pengumpulan tugas dan mengaktifkan *timer* pada RTC untuk menghitung waktu sampai batas yang ditentukan *user* pada *input deadline* sebelumnya.

B. Buka Box

Buka *box* memiliki fungsi yang sama seperti pada fitur buka *box* di menu Admin. Jika pada menu Admin dapat membuka semua *box* yang tersedia, maka fitur buka *box* pada menu User hanya bisa dilakukan pada *box* tugas yang digunakan oleh *user*. Saat *user login* kembali dan menekan tombol Buka Box, maka akan muncul tampilan menu Data Tugas Masuk. Selain itu, relai pada *box* tugas akan aktif dan menggerakkan *solenoid* untuk membuka pintu *box* tugas. Kemudian *user* dapat mengambil tugas yang ada di dalam *box* tugas. Data tugas masuk dapat dilihat pada Gambar 4.24 dibawah ini.



DATA TUGAS MASUK		
1	11	21
2	12	22
3	13	23
4	14	24
5	15	25
6	16	26
7	17	27
8	18	28
9	19	29
10	20	30

Gambar 4.24 Tampilan Menu Berisi Data Pengumpulan Tugas

Pada Gambar 4.24 menunjukkan tampilan menu yang memuat data pengumpulan tugas. Menu ini memuat jumlah tugas yang telah dikumpulkan

beserta waktu pengumpulan pada masing-masing tugas. Batas tugas yang dapat masuk adalah 30 tugas dengan ketebalan maksimal 1.5 cm.

Hasil dari tampilan menu pada LCD TFT sebagai *output* tampilan dapat bekerja dengan baik. LCD TFT mampu menampilkan data yang dibutuhkan untuk *user* serta mahasiswa sehingga dapat melihat informasi mengenai *box* untuk mahasiswa ataupun data pengumpulan tugas untuk *user*. Data dari bagian proses mikrokontroler dapat ditampilkan dengan baik pada LCD TFT. Tampilan data berupa angka dan huruf jelas terlihat diamati dengan panca indra mata. Tampilan menu Admin, User, Pilih *Box*, Buka *Box*, Tambah User, Hapus User, Edit *Box*, Input Mata Kuliah dan Deadline System tertampil pada LCD TFT telah sesuai dengan tingkat *error* 0%.

4.7 Pengujian Fungsi TaBox

Terdapat 3 fungsi utama pada sistem TaBox yang akan dilakukan pengujian pada fungsinya. Pengujian fungsi TaBox untuk mengetahui berfungsi tidaknya penyimpanan data masukan dari *user*, pendeteksian tugas dan *timer deadline* pengumpulan tugas.

4.7.1 Penyimpanan Data User

TaBox memiliki sistem untuk menyimpan data *user* yang kemudian akan disimpan ke dalam EEPROM. Data masukan *user* tersebut nantinya akan ditampilkan saat LCD TFT berada pada menu utama TaBox. Data yang ditampilkan adalah informasi mengenai *box* berupa data mata kuliah *user* dan waktu *deadline* pengumpulan tugas.

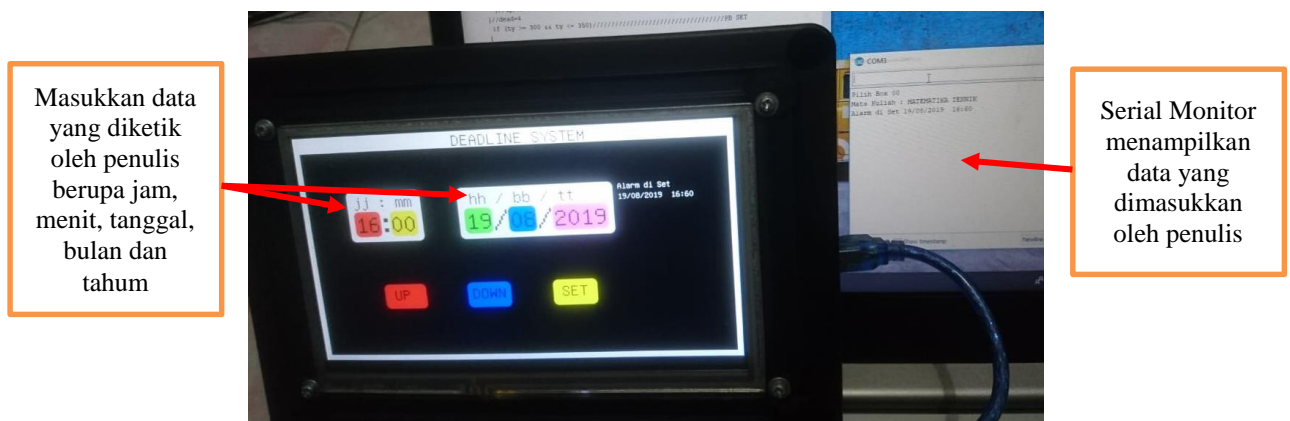
Pengujian dilakukan oleh penulis dengan *login* ke sistem TaBox menggunakan RFID *tag* milik *user* yang telah terdaftar pada sistem untuk masuk ke dalam menu User. Penulis akan mendekati RFID *tag user* ke *box* kontrol 2 untuk dilakukan pemindaian. Setelah selesai dipindai oleh sistem, LCD TFT akan menampilkan menu User kemudian penulis menekan tombol Pakai Box dan masuk ke dalam menu untuk memilih *box* nomer berapa yang akan dipakai. Penulis memilih *box* 00 dan harus memasukkan data pada menu Input Mata Kuliah *user*.

Berikut merupakan data mata kuliah *user* yang dimasukkan oleh penulis, ditunjukkan pada Gambar 4.25 berikut.



Gambar 4.25 Pengujian *Input* Mata Kuliah *User*

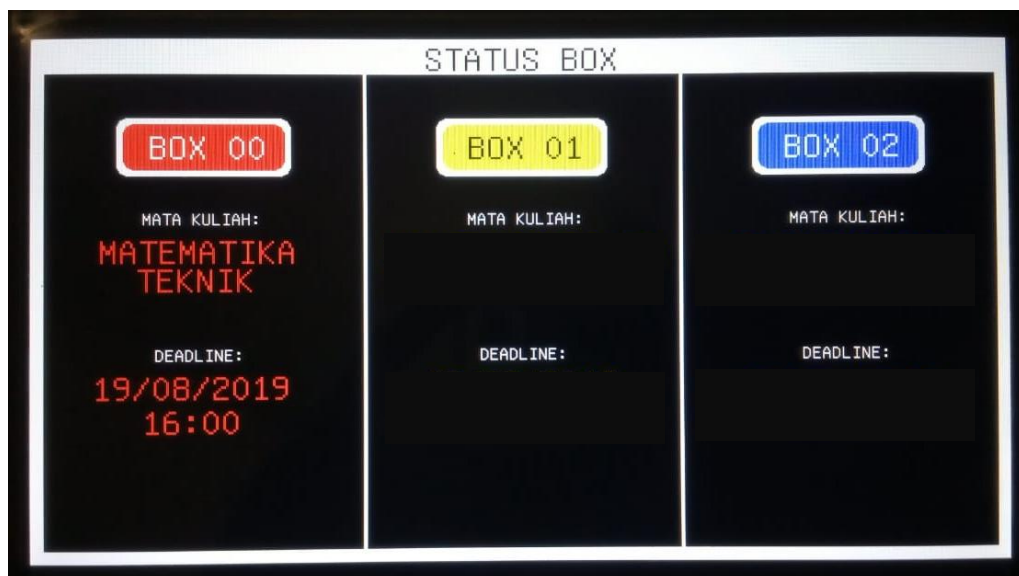
Penulis memasukkan data mata kuliah *user* “MATEMATIKA TEKNIK”, lalu penulis menekan tombol Enter. Tombol Enter ini akan menyimpan data mata kuliah *user* di EEPROM. Terlihat dari serial monitor yang menampilkan mata kuliah *user* yang berarti mata kuliah *user* berhasil tersimpan pada EEPROM saat penulis menekan tombol Enter. Kemudian LCD TFT akan menampilkan menu Deadline System, berikut adalah tampilan menu Deadline System yang telah diisikan data waktu oleh penulis pada Gambar 4.26 berikut.



Gambar 4.26 Pengujian *Input* Data Waktu *Deadline*

Penulis memasukkan data waktu *deadline* pengumpulan tugas *user* “16:00 19/08/2019”, lalu penulis menekan tombol SET. Tombol SET ini akan menyimpan data waktu *deadline* di EEPROM. Terlihat dari serial monitor yang menampilkan waktu *deadline* pengumpulan tugas *user* yang berarti waktu *deadline* pengumpulan tugas *user* berhasil tersimpan pada EEPROM saat penulis menekan tombol SET.

Saat proses *input* data terakhir telah selesai, ditandai dengan penulis menekan tombol SET pada menu Deadline System, maka EEPROM telah menyimpan 2 data, yaitu mata kuliah *user* dan waktu *deadline* pengumpulan tugas. Setelah penekanan tombol SET oleh penulis, LCD TFT akan kembali menampilkan menu utama yang telah menampilkan data informasi *box* pada kolom *box* 00. Menu utama hasil dari masukan data oleh penulis dapat dilihat pada Gambar 4.27 dibawah ini.



Gambar 4.27 Hasil Pengujian Tampilan Data *User* pada EEPROM

Hasil *input* data oleh penulis berhasil disimpan pada EEPROM dan ditampilkan di menu utama TaBox. Terbukti pada Gambar 4.27 diatas. Untuk memastikan kembali berhasil tidaknya fungsi penyimpanan data *user*, maka penulis melakukan pengujian seperti langkah sebelumnya, mulai dari pemindaian RFID *tag user*. Pengujian dilakukan kembali sebanyak 4 kali lagi dan dijabarkan dalam bentuk tabel.

Berikut merupakan tabel hasil pengujian penyimpanan data *user* yang dapat dilihat pada Tabel 4.4 berikut.

Tabel 4.4 Hasil Pengujian Penyimpanan Data *User*

Uji ke-	Mata Kuliah User		Waktu <i>Deadline</i>	
	Mata Kuliah Yang di input Pada Menu Input Mata Kuliah	Mata Kuliah Yang Tersimpan Pada EEPROM dan Tampil di Menu Status Box	Waktu <i>Deadline</i> Yang di input Pada Menu Deadline System	Waktu <i>Deadline</i> Yang Tersimpan Pada EEPROM dan Tampil di Menu Status Box
1	MATEMATIKA TEKNIK	MATEMATIKA TEKNIK	19/08/2019 16:00	19/08/2019 16:00
2	MIKROKONTROLER	MIKROKONTROLER	01/08/2019 12:00	01/08/2019 12:00
3	RMO	RMO	03/08/2019 22:00	03/08/2019 22:00
4	B.INDO	B.INDO	15/08/2019 16:00	15/08/2019 16:00
5	BHI	BHI	21/08/2019 16:00	21/08/2019 16:00

Dari Tabel 4.4 diatas, dapat disimpulkan bahwa data yang di *input* dengan data yang di simpan pada EEPROM yang kemudian ditampilkan di menu Status Box pada LCD TFT adalah sama. Hal ini membuktikan sistem penyimpanan data *user* telah sesuai dengan yang diinginkan oleh penulis dan memiliki tingkat *error* 0%.

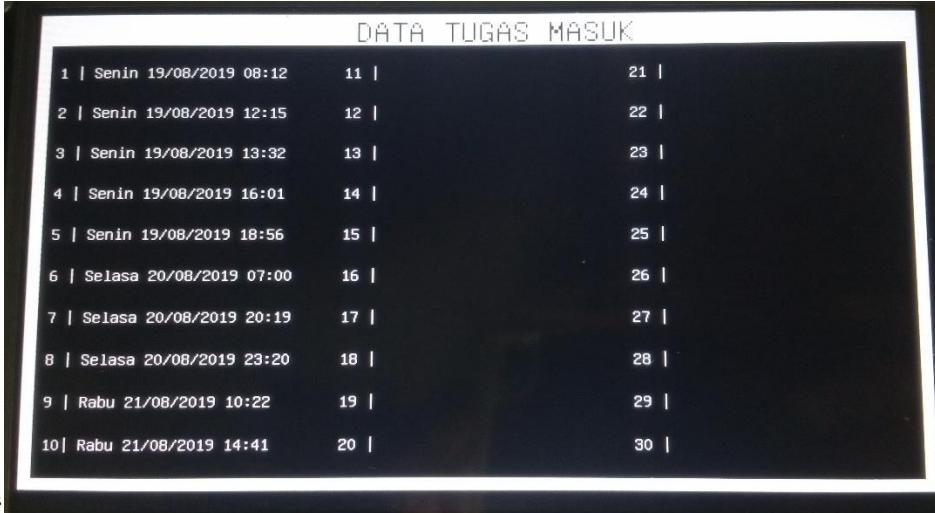
4.7.2 Deteksi Tugas

TaBox memiliki sistem untuk mendeteksi dan mencatat waktu apabila terdeteksi ada tugas yang masuk ke dalam *box* tugas. Pendeteksi tugas menggunakan sensor TCRT5000 *infrared reflective* yang mendeteksi ada atau tidaknya suatu obyek, dalam hal ini tugas. TCRT5000 mendeteksi tugas

berdasarkan intensitas pemantulan sinar infra merah yang dipancarkan oleh *transmitter* (IR led) dan diterima oleh *receiver* (photo transistor).

Saat tugas dimasukkan ke dalam *box* tugas melalui pintu masuk tugas, sensor yang diletakkan pada ujung pintu masuk tugas dalam *box* tugas akan mendeteksi adanya tugas dan memberikan sinyal LOW ke sensor. Kemudian hal tersebut, akan mengaktifkan program pengambilan data waktu pada RTC DS3231. Data yang ada pada RTC DS3231 akan tersimpan pada EEPROM, data ini akan ditampilkan saat *user* login dan menekan fitur buka *box* pada menu User.

Penulis melakukan pengujian fungsi deteksi tugas dengan memasukkan 10 tugas ke dalam *Box* 00 selama 3 hari tehitung dari tanggal 19 – 21 Agustus 2019, uji coba ini dilakukan dalam waktu yang berbeda-beda pada setiap tugas. Hasil pengujian dapat dilihat pada LCD TFT yang akan menampilkan data saat tugas masuk tersebut. Tampilan data tugas yang masuk dapat dilihat pada Gambar 4.28 berikut.



DATA TUGAS MASUK			
1		Senin 19/08/2019 08:12	11 21
2		Senin 19/08/2019 12:15	12 22
3		Senin 19/08/2019 13:32	13 23
4		Senin 19/08/2019 16:01	14 24
5		Senin 19/08/2019 18:56	15 25
6		Selasa 20/08/2019 07:00	16 26
7		Selasa 20/08/2019 20:19	17 27
8		Selasa 20/08/2019 23:20	18 28
9		Rabu 21/08/2019 10:22	19 29
10		Rabu 21/08/2019 14:41	20 30

Gambar 4.28 Hasil Pengujian Data Tugas Masuk

Dari 10 tugas yang dimasukkan, tugas dipastikan terdeteksi semua dan data waktu dapat diambil dari RTC untuk masing-masing tugas. Pada tampilan menu Data Tugas Masuk, data yang masuk telah di tampilkan berurutan dan sesuai dengan yang diinginkan oleh penulis. Apabila dijabarkan dalam bentuk tabel dan

dibandingkan antara waktu pada RTC dengan waktu *real* pada Time.is, maka hasil pengujian tersebut dapat dilihat pada Tabel 4.5 berikut.

Tabel 4.5 Hasil Pengujian Sensor TCRT Mendeteksi Tugas

Tugas ke-	Tugas Terdeteksi	Nasional Standart Time Time.is (Jam:Menit:Detik)	RTC DS3231 (Jam:Menit:Detik)	Error (second)
1	Terdeteksi	08:12:57	08:12:32	25
2	Terdeteksi	12:16:20	12:15:55	25
3	Terdeteksi	13:32:25	13:32:00	25
4	Terdeteksi	16:01:49	16:01:24	25
5	Terdeteksi	18:56::32	18:56::07	25
6	Terdeteksi	07:01:10	07:00:45	25
7	Terdeteksi	20:19:48	20:19:23	25
8	Terdeteksi	23:20:48	23:20:48	25
9	Terdeteksi	10:23:18	10:22:53	25
10	Terdeteksi	14:41:27	14:41:02	25

Pada Tabel 4.5 dapat dianalisa bahwa semua tugas dapat terdeteksi dan data waktu pengumpulannya sesuai dengan RTC DS3231. Namun, RTC DS3231 mengalami perbedaan waktu dengan waktu yang sebenarnya dikarenakan saat proses *upload* program membutuhkan waktu selama 25 detik. Hal inilah menyebabkan adanya perbedaan waktu pada RTC DS3231. Dari perbedaan waktu tersebut didapatkan nilai selisih yaitu 25 detik, dilanjutkan dengan rumus untuk mencari perhitungan % *error* diambil dari sampel tugas ke-1 yang dikumpulkan pada 08:12:32 dan waktu sebenarnya 08:12:57 adalah:

$$\%error = \frac{|Waktu\ sekarang - Waktu\ yang\ terprogram|}{Waktu\ sekarang} \times 100\% \dots\dots\dots (4.2)$$

$$\%error = \frac{|57\ second - 32\ second|}{57\ second} \times 100\%$$

$$\%error = 0.4\%$$

4.7.3 Sistem *Deadline*

Pengujian program sistem *deadline* bertujuan untuk mengetahui sesuai atau tidaknya waktu *deadline* yang di atur oleh *user* dengan kondisi *box* tugas di lapangan. Cara kerja sistem *deadline* yaitu dengan melakukan pengaturan *input* data waktu *deadline* yang diinginkan untuk pengumpulan tugas. Data yang di *input* berupa jam, menit, tanggal, bulan dan tahun.

Pengujian dilakukan dengan memberikan *input* data waktu untuk mengetahui respon yang diberikan setelah waktu habis, apakah servo bisa menutup lobang pada pintu masuk tugas atau tidak. Pengujian dilakukan sebanyak 5 kali dengan waktu yang berbeda-beda. Hal ini dilakukan untuk memastikan bahwa sistem kontrol TaBox terus berjalan sesuai dengan *timer deadline* pengumpulan tugas yang telah diatur oleh *user* saat pemakaian TaBox. *Output* yang diinginkan saat fungsi *deadline* pengumpulan tugas ini berhasil adalah Bergeraknya servo untuk menutup pintu masuk tugas pada *box* tugas. Hasil pengujian fungsi *deadline* pengumpulan tugas dapat dilihat pada Tabel 4.6 dibawah ini.

Tabel 4.6 Hasil Pengujian Sistem *Deadline* Pada *Box* 00

No	Seting Waktu (Tanggal/Bulan/Hari Jam:Menit)		Keterangan
	Dari	Sampai	
1	28/07/2019 13:27	28/07/2019 13:30	Pintu Masuk Tugas Tertutup
2	01/08/2019 10:15	01/08/2019 12:00	Pintu Masuk Tugas Tertutup
3	02/08/2019 22:00	03/08/2019 22:00	Pintu Masuk Tugas Tertutup
4	13/08/2019 07:00	15/08/2019 16:00	Pintu Masuk Tugas Tertutup
5	19/08/2019 07:00	21/08/2019 16:00	Pintu Masuk Tugas Tertutup

Hasil pengujian *Deadline System* sebagai *timer* pengumpulan tugas menggunakan RTC DS3231 sebagai perwaktuan digital. RTC DS3231 mampu melakukan pengaturan *deadline* sesuai dengan yang di *input* oleh penulis secara

tepat. Cara kerja dari pengaturan *deadline* yaitu berfungsi untuk menggerakkan servo untuk menutupi pintu masuk tugas yang menandakan bahwa waktu *deadline* telah habis. Hasil pengujian pada Tabel 4.6 menunjukkan pengaturan *deadline* telah sesuai dengan data waktu yang dimasukkan dengan tingkat *error* 0%.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pembahasan tentang proyek akhir ini, maka dapat diambil kesimpulan:

1. TaBox dirancang sebagai wadah pengumpulan tugas mahasiswa berbasis Arduino sebagai kontrol utama yang memiliki kemampuan untuk mendeteksi tugas masuk dengan sensor TCRT5000, penyimpanan data *user* dengan SD Card dan EEPROM serta sistem *deadline* sebagai pembatas waktu pengumpulan tugas dengan menggunakan RTC DS3231.
2. Informasi mengenai *box* dan *deadline* pengumpulan tugas ditampilkan dalam LCD TFT berfungsi dengan baik. Hal ini dibuktikan dengan hasil pengujian penyimpanan data *user*, dimana data yang disimpan dapat ditampilkan pada menu utama LCD TFT RA8875 dengan tingkat *error* 0%.
3. Dari hasil pengujian deteksi tugas, sensor TCRT5000 dapat dengan akurat mendeteksi setiap tugas yang masuk diikuti dengan pencatatan waktu oleh RTC DS3231.
4. Hasil pengujian sistem *deadline* menunjukkan sistem *deadline* pengumpulan tugas telah sesuai dengan data waktu yang dimasukkan oleh penulis saat pengujian sistem *deadline* dengan tingkat *error* 0%.

5.2 Saran

Berdasarkan kesimpulan di atas, maka penulis dapat memberikan saran yang dapat digunakan untuk pengembangan lebih lanjut alat ini, yaitu:

1. Mengganti kontrol utama yang memiliki spesifikasi setingkat lebih tinggi dari Arduino agar mempermudah dalam pemrosesan data dan mempercepat respon pembacaan data dari sensor TCRT5000 maupun sensitivitas *touchscreen* pada LCD TFT RA8875.

2. Menambahkan informasi mengenai pemakai *box* pada menu utama LCD TFT RAA8875 dan membuat tampilan LCD TFT agar lebih menarik dengan menambahkan gambar ataupun bentuk yang beragam.
3. Menggunakan sensor yang lebih akurat dibanding sensor TCRT5000 karna ada kondisi jika terdapat tugas yang dimasukkan ke dalam *box* tugas sudah terbaca oleh sensor, namun tugas tersebut di ambil kembali dan sensor tetap membaca tugas tersebut sebagai tugas yang telah dikumpulkan.
4. Mempertahankan ketepatan waktu *deadline* pengumpulan tugas.

DAFTAR PUSTAKA

- [1] Avico, S. Rico dan Mujidin, "Hubungan antara Konformitas dengan Prokrastinasi Akademik pada Mahasiswa Bengkulu yang Bersekolah Di Yogyakarta Vo. 1," dalam *Jurnal Psikologi*, vol. 1, Yogyakarta, Fakultas Psikologi Universitas Ahmad Dahlan, 2015, p. 1.
- [2] Mustakini and H. Jogiyanto, *Sistem Informasi Teknologi*, Yogyakarta: Andi Offset, 2009.
- [3] W. Sri, "repository.unej.ac.id," 7 Juni 2016. [Online]. Available: <http://repository.unej.ac.id/bitstream/handle/123456789/76585/Sri%20Wulandari%20-%2020131903102019%20-1.pdf?sequence=1>. [Accessed 4 September 2019].
- [4] Herul, "AnakKendali," 30 Desember 2018. [Online]. Available: <https://www.anakkendali.com/2018/12/arduino-cara-set-alarm-dengan-tombol-rtc-ds3231-lcd-16x2-i2c/>. [Accessed 4 September 2019].
- [5] F. A. Muhammad, "BAB III PERANCANGAN ALAT," 2013. [Online]. Available: <https://docplayer.info/47426955-Bab-iii-perancangan-alat.html>. [Accessed 4 September 2019].
- [6] A. P. Sigit, *Easy to Use SolidWorks 2009*, Yogyakarta: ANDI, 2009.
- [7] J. Greg and D. Richard, "Taking Control of SolidWorks," in *SolidWorks For Dummies*, Hoboken, Wiley Publishing, Inc, 2011, pp. 19-20.
- [8] Y. Wachid, "Sejarah Sistem Kontrol," dalam *SISTEM KONTROL OTOMOTIF*, Yogyakarta, Deepublish, 2017, p. 1.
- [9] Y. Erni, "Konsep Dasar Sistem Kontrol," dalam *Belajar Sistem Kontrol SOAL & PEMBAHASAN*, Malang, UB Press, 2017, p. 2.
- [10] Lab Elektronika, "ARDUINO MEGA 2560 MIKROKONTROLER ATmega2560," 17 Februari 2017. [Online]. Available: <http://www.labelektronika.com/2017/02/arduino-mega-2560-mikrokontroler.html>. [Diakses 3 April 2019].
- [11] Arduino, "ARDUINO UNO REV3," ARDUINO, 2019. [Online]. Available: <https://store.arduino.cc/usa/arduino-uno-rev3>. [Diakses 1 Mei 2019].

- [12] Caratekno, "PENGERTIAN ARDUINO UNO MIKROKONTROLER ATMEGA328," 6 Juli 2015. [Online]. Available: <https://www.caratekno.com/pengertian-arduino-uno-mikrokontroler/>. [Diakses 5 April 2019].
- [13] F. Ahmad, "Tutorial Arduino mengakses module RFID RC522," 20 Desember 2017. [Online]. Available: <https://www.nyebarilmu.com/tutorial-arduino-mengakses-module-rfid-rc522/>. [Diakses 3 April 2019].
- [14] F. Ahmad, "Tutorial Arduino mengakses module RTC DS3231," 3 Desember 2017. [Online]. Available: <https://www.nyebarilmu.com/tutorial-arduino-mengakses-module-rtc-ds3231/>. [Diakses 3 April 2019].
- [15] EastRising, "Serial SPI Arduino 7" TFT LCD Touch Shield RA8875 for Mega/Due/Uno," [Online]. Available: <https://www.buydisplay.com/default/serial-spi-arduino-7-inch-tft-lcd-touch-shield-ra8875-for-mega-due-uno>. [Diakses 3 April 2019].
- [16] R. Fluck, "Servo Motor Control with AndroidDAQ," dalam *THE ULTIMATE ANDROIDAQ GUIDE*, Victoria, Leanpub, 2014, p. 76.
- [17] Zonaelektro, "Motor Servo," 14 Desember 2014. [Online]. Available: <http://zonaelektro.net/motor-servo/>. [Diakses 5 April 2019].
- [18] Super Electronics, "Door Lock Solenoid 12V Canggih | Door Lock Full Cover 12V," Tokosuperelectronicx.com, [Online]. Available: <http://tokosuperelectronics.com/door-lock-solenoid-12v-canggih-door-lock-full-cover-12v/>. [Diakses 1 Mei 2019].
- [19] S. Hari, "Modul Relay," in *MONSTER ARDUINO 2: Panduan Praktis Belajar Arduino untuk Pemula*, ELANGSAKTI.com, 2017, pp. 76-78.
- [20] [Online]. Available: <http://sttar.ac.id/wp-content/uploads/2017/06/BAB-11-2.docx>.
- [21] Wikipedia, "Arduino," 18 Januari 2019. [Online]. Available: <https://id.wikipedia.org/wiki/Arduino>. [Diakses 3 April 2019].
- [22] Fantastic, "Layar Sentuh (Touchscreen) pada HP atau Tablet atau Smartphone," 21 Februari 2019. [Online]. Available: <https://mainthebest.com/smartphones/pengertian-layar-lcd-tft-ips-oled-super-amoled-retina/>. [Diakses 3 April 2019].

[23] Arduino, "Arduino Mega ADK," Arduino, 2019. [Online]. Available: <https://www.arduino.cc/en/Main/ArduinoBoardMegaADK?from=Main.ArduinoBoardADK>. [Diakses 1 Mei 2019].

[24] Time.is, "Time in Sungailiat, Kepulauan Bangka Belitung, Indonesia now," Time.is, Agustus 2019. [Online]. Available: <https://time.is/id/Sungailiat>. [Accessed Agustus 2019].

LAMPIRAN 1
(Daftar Riwayat Hidup Penulis)

DAFTAR RIWAYAT HIDUP

II. Data Pribadi

Nama Lengkap : Dodi
Tempat, Tanggal Lahir : Cirebon, 21 Mei 1997
Alamat : Jl. R. Hundani, Gabek
No. HP : 0821 8629 3994
e-mail : dodipolman@gmail.com



III. Riwayat Pendidikan

Pendidikan	Tahun	Jurusan
SD N 37 Pangkalpinang	2004 - 2010	-
SMP N 1 Pangkalpinang	2010 - 2013	-
SMK N 2 Pangkalpinang	2013 - 2016	Teknik Informatika
Politeknik Manufaktur Negeri Bangka Belitung	2016 – 2019	Teknik Elektro dan Informatika

IV. Pengalaman Kerja

Tempat Kerja	Periode Kerja
PT Timah Tbk	Februari – Maret 2015
PT Renerconsis System	September 2018 – Januari 2019

DAFTAR RIWAYAT HIDUP

I. Data Pribadi

Nama Lengkap : Rahmah Nadiyah
Tempat, Tanggal Lahir : Pangkalpinang, 6 Mei 1998
Alamat : Jl. Usman Ambon No. 103
No. HP : +62 813 7324 7357
e-mail : rahamapkp@gmail.com



II. Riwayat Pendidikan

Pendidikan	Tahun	Jurusan
SDIT Pondok Duta Depok Jawa Barat	2004 - 2010	-
SMPIT Al Multazam Kuningan Jawa Barat	2010 - 2013	-
SMAN 1 Pangkalpinang	2013 - 2016	Matematika Ilmu Alam
Politeknik Manufaktur Negeri Bangka Belitung	2016 – 2019	Teknik Elektro dan Informatika

III. Pengalaman Kerja

Tempat Kerja	Periode Kerja
PT Semen Padang	September 2018 – Januari 2019

LAMPIRAN 2
(Standart Operating Procedure
(SOP) Penggunaan Alat)



STANDART OPERATING PROCEDURE TABOX

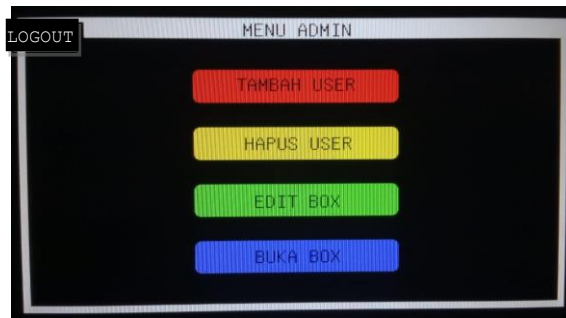
NO : 1

REVISI : 0

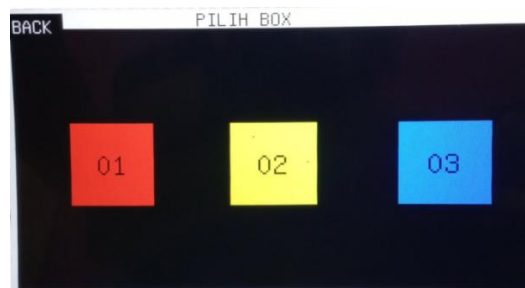
TANGGAL : 01 September 2019

HALAMAN : 1

1. Hubungkan kabel sambungan *box* tugas no. 00, 01 dan 02 ke *socket* yang ada pada *box* kontrol sesuai dengan urutan nomer *box* tugas tersebut.
2. Hubungkan Adaptor 12V/3A ke sumber AC lalu *barrelnya* dihubungkan ke masukan untuk *box* kontrol.
3. Tekan saklar yang ada di samping *box* kontrol, *box* kontrol aktif ditandai dengan hidupnya lampu indikator pada saklar. Saat *box* kontrol telah mendapat masukan daya, maka kipas di dalam *box* kontrol akan berputar dan LCD TFT akan menampilkan menu Status Box sebagai tampilan utama.
4. Pindai RFID *tag* atau kartu akses milik *admin* ke RFID Reader. Apabila kartu terdaftar sebagai *asmin* dalam sistem, maka LCD TFT akan menampilkan menu Admin yang ditunjukkan pada gambar dibawah ini.



5. Perhatikan gambar pada *step* 4 untuk melihat setiap menu yang tersedia. Apabila ingin mengakses tombol menu Tambah User, maka lanjut ke *step* 6. Untuk tombol menu Hapus User, maka lanjut ke *step* 8. Jika ingin mengakses tombol menu Edit Box, maka lanjut ke *step* 10 dan untuk tombol menu Buka Box, maka dapat dilihat pada *step* 13.
6. Pada tombol menu Tambah User, akan ada perintah pada layar untuk memindai RFID *tag* milik *user* yang ingin didaftarkan oleh *admin* kedalam sistem TaBox.
7. Setelah RFID *tag user* baru telah dipindai, apabila RFID *tag user* baru telah terdaftar maka LCD TFT akan kembali pada menu utama yaitu, menu Status Box. Untuk memastikan terdaftar tidaknya kartu akses milik *user*, *admin* dapat mengecek kartu dengan memindai kembali kartu akses tersebut. Apabila kartu berhasil dipindai dan masuk ke menu User, maka kartu akses berhasil didaftarkan.
8. Pada tombol menu Hapus User, akan ada perintah pada layar untuk memindai RFID *tag* milik *user* yang ingin dihapus datanya oleh *admin* dari sistem TaBox.
9. Setelah RFID *tag user* tersebut telah dipindai. apabila RFID *tag user* tersebut berhasil dihapus maka LCD TFT akan kembali pada menu utama yaitu, menu Status Box. Untuk memastikan telah terhapus tidaknya kartu akses milik *user*, *admin* dapat mengecek kartu dengan memindai kembali kartu akses tersebut. Apabila kartu tidak berhasil masuk ke menu User kembali, maka data pada kartu akses *user* berhasil dihapus dari sistem TaBox.
10. Pada tombol menu Edit Box, akan ada pilihan untuk memilih box nomer berapa yang akan di edit datanya, yang ditunjukkan pada gambar dibawah ini.

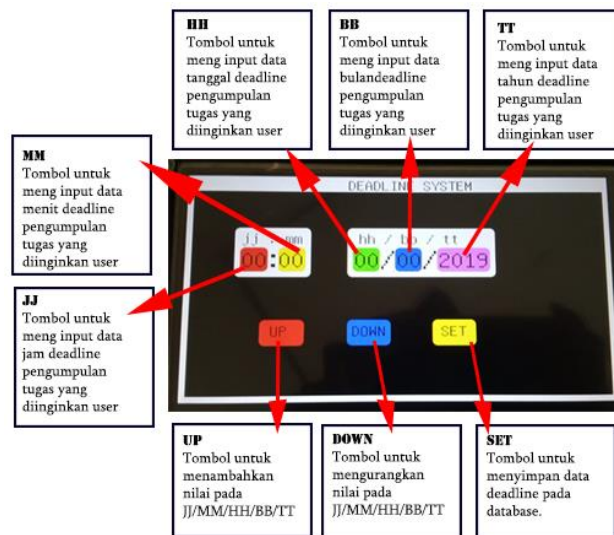


11. Pilih box yang akan di edit datanya. Kemudian akan tampil menu Masukkan Mata Kuliah, berikut merupakan gambar beserta keterangan fungsi tombol pada menu ini.



Apabila *admin* ingin mengubah data mata kuliah *user*, maka *admin* perlu mengklik tombol CLEAR pada keyboard terlebih dahulu baru mengetikkan mata kuliah *user* dan menekan tombol ENTER. Namun jika *admin* tidak ingin mengubah mata kuliah *user*, *admin* hanya perlu menekan tombol ENTER saja.

12. Setelah selesai memasukkan data mata kuliah *user*, LCD TFT akan menampilkan menu Deadline System. Berikut merupakan gambar beserta keterangan fungsi tombol pada menu Deadline System



Apabila telah selesai mengedit waktu *deadline*, maka *admin* perlu menekan tombol SET untuk menyimpan data dan kembali ke tampilan menu utama.

13. Pada tombol menu Buka Box, akan ada pilihan untuk memilih box nomer berapa yang akan di buka, tampilan menu pilih *box* seperti pada gambar pada *step 10* diatas.

14. Pilih box yang akan di buka. Kemudian akan tampil menu Data Tugas Masuk yang berisikan data jumlah tugas yang masuk beserta waktu pengumpulan tugas mahasiswa.

15. Buka pintu *box* tugas untuk mengambil tugas mahasiswa setelah itu tutup kembali dan klik tombol LOGOUT. Lalu LCD TFT akan kembali menampilkan menu Status Box

LAMPIRAN 3
(Program Arduino Uno)

```

#include <EEPROM.h> //Untuk Menyimpan ID card
#include <SPI.h> //RFID reader menggunakan komunikasi SPI
#include <MFRC522.h> //Library RC522 Module
#define SS_PIN 10
#define RST_PIN 9
int menu = '0'; // kondisi menu yang aktif
String stat;
char c;
boolean match = false; // cek kartu sesuai atau tidak
boolean programMode = false; // inisialisasi program sekarang kondisi false
int successRead; // variabel bahwa kartu dapat di baca
byte storedCard[4]; // Simpan ID ke EEPROM
byte readCard[4]; // Baca ID dari RFID Module
byte masterCard[4]; // Simpan master card ID baca dari EEPROM
uint16_t tx, ty;
MFRC522 mfrc522(SS_PIN, RST_PIN);
void setup() {
  Serial.begin(9600);
  SPI.begin();
  mfrc522.PCD_Init();
  mfrc522.PCD_SetAntennaGain(mfrc522.RxGain_max);
  if (EEPROM.read(1) != 1) { //JIKA MASTER CARD BELUM DI MASUKAN
    Serial.write("0");
    delay(100);
    do {successRead = getID();}
    while (!successRead); //berjalan jika successread belum sesuai
    for ( int j = 0; j < 4; j++ ) {
      EEPROM.write( 2 + j, readCard[j] );
      delay(50);}
    EEPROM.write(1, 1);
    Serial.write("1");
  }
  for ( int i = 0; i < 4; i++ ) { // Baca Master Card dari EEPROM
    masterCard[i] = EEPROM.read(2 + i); // simpan Master Card ke EPROM
    delay(50); }
}
void loop() {
  if (menu == '0');////////Menu Awal
  {do{successRead = getID();}
  while (!successRead); //Stop looping sampai kartu terbaca
  if ( isMaster(readCard) ){
    Serial.write("3");//Admin Login
    menu = '1'; }
  else{
    if ( findID(readCard) ) {
      Serial.write("4");//User login
      menu = '2'; }
    else{
      Serial.write("5");//Gagal login
      return 0; }
  }
} //Batas Menu Awal
while (menu == '1') Masuk Menu Asmin
{if (Serial.available()>0)
  {delay(10);
  c = Serial.read();}
  stat = c;
  if (stat == "6") ///menu tambah user
  {menu = '3'; }
  if (stat == "7") //menu hapus user
  {menu = '4'; }
  if (stat == "8") //Logout
  {delay(100);

```

```

    menu = '0'; }
} //Batas Menu Admin
while (menu == '3') ///Masuk Menu Admin Tambah User
{
    do{successRead = getID();}
    while (!successRead);
    if ( !findID(readCard) )
    {Serial.write("9");
     Serial.write("a");
     writeID(readCard); }
    else
    {Serial.write("c");
     delay(1000);
     menu = '0';
     break; }
} //Batas Menu Tambah User
while (menu == '4') //Masuk Menu Admin Hapus User
{do{successRead = getID();}
 while (!successRead);
 if ( findID(readCard) )
 {Serial.write("d");
  Serial.write("e");
  deleteID(readCard); }
 else
 {Serial.write("g");
  menu = '0';
  break; }
} //Batas Menu Hapus User
while (menu == '2') // Masuk Menu User
{do {successRead = getID();}
 while (!successRead);
 if ( findID(readCard) )
 {Serial.write("z");
  menu='0'; }
 else
 {Serial.write("z");
  menu='0'; }
} //Batas Menu User
} //Batas Void Loop
int getID() { //Baca ID RFID tag
    if ( ! mfrc522.PICC_IsNewCardPresent() ) { return 0; }
    if ( ! mfrc522.PICC_ReadCardSerial() ) {}
    for (int i = 0; i < 4; i++) { //Mulai Membaca
        readCard[i] = mfrc522.uid.uidByte[i];}
    mfrc522.PICC_HaltA(); // Berhenti Membaca
    return 1; }
boolean isMaster( byte test[] ) {
    if ( checkTwo( test, masterCard ) )
        return true;
    else{return false;}
} //Batas Fungsi
boolean findID( byte find[] ) { //mencari ID di EEPROM
    int count = EEPROM.read(0); // Baca Byte Pertama EEPROM
    for ( int i = 1; i <= count; i++ ) { // Loop untuk bacaEEPROM
        readID(i); // Baca ID dari EEPROM, yang disimpan di storedCard[4]
        if ( checkTwo( find, storedCard ) ) { //Cek kembali data di EEPROM
            return true;
            break; // keluar jika benar }
        else { // jika salah ulang lagi
            }
        }
    return false;
} //Batas Fungsi

```



```

boolean checkTwo ( byte a[], byte b[] ) { //Fungsi Cek ulang EEPROM
    if ( a[0] != NULL ) // Cek isi array
        match = true; // asumsikan isinya sama
    for ( int k = 0; k < 4; k++ ) { // Looping 4 kali
        if ( a[k] != b[k] ) // jika a != b maka match = false, jika salah satu maka
            match = false; }
    if ( match ) { // Jika sesuai maka benar
        return true; // kembali ke kondisi }
    else {
        return false; // kembali ke kondisi }
} //Batas fungsi

void readID( int number ) { //Baca ID RFID tag
    int start = ( number * 4 ) + 2; // Output dari start
    for ( int i = 0; i < 4; i++ ) { // Loop 4 kali untuk dapat 4 Bytes
        storedCard[i] = EEPROM.read(start + i); // Ubah dari EEPROM ke Array}
} //Batas Fungsi

int findIDSLOT( byte find[] ) { //Cek ruang kosnmg di EEPROM
    int count = EEPROM.read(0); // Baca Byte pertama dari EEPROM
    for ( int i = 1; i <= count; i++ ) {
        readID(i);
        if ( checkTwo( find, storedCard ) ) {
            return i;
            break; }
    }
} //Batas Fungsi

void writeID( byte a[] ) { //Meyimpan ID ke EEPROM
    if ( !findID( a ) ) { // Cek Apakah sudah tersimpan apa belum
        int num = EEPROM.read(0);
        int start = ( num * 4 ) + 6;
        num++;
        EEPROM.write( 0, num );
        for ( int j = 0; j < 4; j++ ) {
            EEPROM.write( start + j, a[j] );
            delay(50); }
        Serial.write("b");
        delay(1000); }
    else {Serial.print("Error"); }
} //Batas Fungsi

void deleteID( byte a[] ) {
    if ( !findID( a ) ) { // Cek Apakah sudah tersimpan apa belum
        Serial.print("Error"); // Jika tidak}
    else {
        int num = EEPROM.read(0);
        int slot, looping, j;
        int start; // = ( num * 4 ) + 6;
        int count = EEPROM.read(0);
        slot = findIDSLOT( a );
        start = (slot * 4) + 2;
        looping = ((num - slot) * 4);
        num--;
        EEPROM.write( 0, num );
        for ( j = 0; j < looping; j++ ) {
            EEPROM.write( start + j, EEPROM.read(start + 4 + j));
            delay(50); }
        for ( int k = 0; k < 4; k++ ) {
            EEPROM.write( start + j + k, 0);
            delay(50); }
        delay(1000);
        Serial.write("f");
    }
} //Batas Fungsi

```

LAMPIRAN 4
(Program Arduino Mega 2560)


```

#define sensor_Box00 27
#define relay_Box01 31
#define sensor_Box01 35
#define relay_Box02 39
#define sensor_Box02 43
Servo servo_Box00,servo_Box01, servo_Box02;
int pos = 0;
//////////////////////////////////// Variabel Ketyboard
String Nama_box00, Nama_box01, Nama_box02;
char _put00[10],_put01[10],_put02[10];
char _get00[10]=" ",_get01[10]=" ",_get02[10]=" ";
String stat;
char c;

void setup ()
{Serial.begin(9600);
  Serial1.begin(9600);
  tft.begin(RA8875_800x480);
  tft.touchBegin(RA8875_INT);
  rtc_setup();
  box_setup ();
  digitalWrite (relay_Box00, HIGH);
  digitalWrite (relay_Box01, HIGH);
  digitalWrite (relay_Box02, HIGH);
  servo_Box00.write(0);
  servo_Box01.write(0);
  servo_Box02.write(0);
  drawHomeScreen();
  menu='0';}

void loop()
{tft.changeMode(GRAPHIC);
  if (menu=='0') // Menu Home
  {tft.setTextColor(RA8875_RED);
  tft.setFontScale(1);
  tft.setCursor(55,170);
  tft.print(Nama_box00);
  tft.setTextColor(RA8875_YELLOW);
  tft.setFontScale(1);
  tft.setCursor(290,170);
  tft.print(Nama_box01);
  tft.setTextColor(RA8875_BLUE);
  tft.setFontScale(1);
  tft.setCursor(575,170);
  tft.print(Nama_box02);
  while(menu=='0')//Masuk Menu Utma
  { deteksi_tugas00(); deteksi_tugas01(); deteksi_tugas02();
    cekAlarm_box00();cekAlarm_box01();cekAlarm_box02();
  if (Serial1.available())
  {c = Serial1.read();}
  stat = c;
  if (stat=="0" ){ break; }
  if (stat=="1"){ break; }
  if (stat=="2"){ break; }
  if (stat=="3")
  {drawMenuAdmin();
  menu='1';
  break; }
  if (stat=="4")
  {drawMenuUser();
  menu='2';
  break; }
  if (stat=="5")
  {tft.print("tidak terdeteksi");

```

```

        menu='0'; } }
    }// BatasMENU 0
    if (menu=='1')//Menu Admin
    {if (tft.touchDetect())
        {tft.touchReadPixel (&tx, &ty);
            tx=800-tx;ty=480-ty;
            if (ty >= 2 && ty <= 27)//////////////////// Tombol LOGOUT
            {if (tx > 2 && tx < 62)
                {Serial1.write("8");
                    menu='0';
                    drawHomeScreen();
                    delay (100);
                    while(tft.touchDetect()){};
                }
            }
        }
        if (tx > 250 && tx < 550)
        {if (ty >= 80 && ty <= 130)//////////////////// Tombol Tambah USER
            {drawMenuDataTugasMasuk();
                Serial1.write("6");
                menu='3';
                delay (100);
                while(tft.touchDetect()){};
            }
            if (ty >= 170 && ty <= 220)//////////////////// Tombol HAPUS
                USER
                {drawMenuDataTugasMasuk();
                    Serial1.write("7");
                    menu='4';
                    delay (100);
                    while(tft.touchDetect()){};}
            if (ty >= 260 && ty <= 310)//////////////////// Tombol Edit
                Box
                {menu='5';
                    drawMenuPilihBox ();
                    delay (100);
                    while(tft.touchDetect()){};}
            if (ty >= 350 && ty <= 400)//////////////////// Tombol Buka
                Box
                {menu='D';
                    drawMenuBukaBox ();
                    delay (100);
                    while(tft.touchDetect()){};
                }
            }
        }
    }//Batas MENU 1
    if (menu=='2')//Menu User
    {if (Serial1.available())
        {c = Serial1.read();}
        stat = c;
        if (stat=="z" )
            {drawHomeScreen();
                menu='0'; }
        if (tft.touchDetect())
        {tft.touchReadPixel (&tx, &ty);
            tx=800-tx;ty=480-ty;
            if (tx > 250 && tx < 550)
            {if ((ty >= 150 && ty <= 200))//////////////////// Tombol Pakai Box
                {menu='5';
                    drawMenuPilihBox ();
                    delay (100);
                    while(tft.touchDetect()){};}
                if (ty >= 265 && ty <= 315)//////////////////// Tombol Buka
                    Box
                    { menu='D';
                        drawMenuBukaBox ();
                        delay (100);
                    }
            }
        }
    }

```

```

        while(tft.touchDetect()){};
    }}}
} //Batas MENU 2
if (menu=='3') //Menu Admin Tambah User
{if (Serial1.available()>0)
  { c = Serial1.read();
    stat = c;
    if (stat=="9")
    {tft.setTextColor(RA8875_WHITE);
     tft.setFontScale(0);
     tft.setCursor(20,50);
     tft.write("Kartu Belum Terdaftar ");}
    if (stat=="a")
    {tft.setTextColor(RA8875_WHITE);
     tft.setFontScale(0);
     tft.setCursor(20,90);
     tft.write("Mulai Mendaftarkan Kartu"); }
    if (stat=="b")
    { tft.setTextColor(RA8875_WHITE);
      tft.setFontScale(0);
      tft.setCursor(20,130);
      tft.write("Berhasil ditambahkan,");
      tft.write("Pindai Sekali lagi untuk membuktikan kartu telah terdaftar ");
      tft.setCursor(20,170);
      tft.write("Jika kembali ke menu utama maka kartu berhasil didaftarkan ");}
    if (stat=="c")
    { tft.setTextColor(RA8875_WHITE);
      tft.setFontScale(0);
      tft.setCursor(20,170);
      tft.write("Kartu telah terdaftar kembali ke menu utama");
      drawHomeScreen ();
      menu='0'; }
} //Batas MENU 3
if (menu=='4') //Menu Admin Hapus User
{if (Serial1.available()>0)
  {c = Serial1.read();
   stat = c;
   if (stat=="d")
   { tft.setTextColor(RA8875_WHITE);
     tft.setFontScale(0);
     tft.setCursor(20,50);
     tft.write("Kartu Telah Terdaftar ");}
   if (stat=="e")
   {tft.setTextColor(RA8875_WHITE);
    tft.setFontScale(0);
    tft.setCursor(20,90);
    tft.write("Mulai Menghapus Kartu"); }
   if (stat=="f")
   { tft.setTextColor(RA8875_WHITE);
     tft.setFontScale(0);
     tft.setCursor(20,130);
     tft.write("Berhasil dihapus, Scan Sekali lagi untuk membuktikan");
     tft.setCursor(20,170);
     tft.write("Jika kembali ke menu utama maka kartu berhasil dihapus ");}
   if (stat=="g")
   {tft.setTextColor(RA8875_WHITE);
    tft.setFontScale(0);
    tft.setCursor(20,130);
    tft.write("Berhasil dihapus, Kembali ke menu utama ");
    delay(200);
    drawHomeScreen ();
    menu='0';
  }
}

```

```

} //Batas MENU 4
if (menu=='5') // Menu Admin Edit Box
{ Serial.println ("menu 5 masuk");
  if (tft.touchDetect())
  { tft.touchReadPixel (&tx, &ty);
    tx=800-tx;ty=480-ty;
    if (ty >= 7 && ty <= 47) // Tombol LOGOUT
    {if ((tx > 5 && tx < 95))
      {Serial1.write("8");
        menu='0';
        drawHomeScreen();
        delay (100);
        while(tft.touchDetect()){};}
    }
  if (ty >= 180 && ty <= 310) // BOX 1,2,3
  {if ((tx > 100 && tx < 230)) // Box1
    { Box_Aktif='0';
      menu='6';
      Keyboard_Is_Running_box00();
      drawMenuInputMatkul();
      delay (100);
      while(tft.touchDetect()){};}
    else if ((tx > 350 && tx < 480)) //
Box2
    { Box_Aktif='1';
      menu='6';
      drawMenuInputMatkul();
      delay (100);
      while(tft.touchDetect()){};}
    else if ((tx > 600 && tx < 730)) // Box3
    { Box_Aktif='2';
      menu='6';
      drawMenuInputMatkul();
      delay (100);
      while(tft.touchDetect()){};
    }
  }
} //Batas MENU 5
if (menu=='6') //Panggil Keyboard
{ Serial.println("key 00");
  Keyboard_Is_Running_box00();
} //Batas MENU 6
if (menu=='A') {
  Alarm_Is_Running_Box00();
} //Batas MENU A
if (menu=='B') { //Set Deadline Box01
  Alarm_Is_Running_Box01();
} //Batas MENU B
if (menu=='C') { //Set Deadline Box02
  Alarm_Is_Running_Box02();
} //Batas MENU C
if (menu=='D') // Buka Box
{if (tft.touchDetect())
  { tft.touchReadPixel (&tx, &ty);
    tx=800-tx;ty=480-ty;
    if (ty >= 7 && ty <= 47) // Tombol LOGOUT
    {if (tx > 5 && tx < 95)
      {Serial1.write("8");
        menu='0';
        drawHomeScreen();
        delay (100);
        while(tft.touchDetect()){};}
    }
  if (ty >= 180 && ty <= 310) // BOX 1,2,3

```

```

{if ((tx > 100 && tx < 230))////////////////////////////////////// Box00
  { drawDataTugasBox00();
    Serial.print ("buka box 00");
    digitalWrite (relay_Box00, LOW);
    menu='F';
    delay (100);
    while(tft.touchDetect()){};}
else if ((tx > 350 && tx < 480))////////////////////////////////////// Box01
  { digitalWrite (relay_Box01, LOW);
    menu='G';
    drawDataTugasBox01();
    delay (100);
    while(tft.touchDetect()){};}
else if ((tx > 600 && tx < 730))////////////////////////////////////// Box02
  { digitalWrite (relay_Box02, LOW);
    menu='H';
    drawDataTugasBox02();
    delay (100);
    while(tft.touchDetect()){};
  }}
} //Batas MENU D
Pakai Box
if (menu=='E') //Menu Pakai Box
{if (tft.touchDetect())
  { tft.touchReadPixel (&tx, &ty);
    tx=800-tx;ty=480-ty;
    if (ty >= 7 && ty <= 47)////////////////////////////////////// Tombol LOGOUT
    {if ((tx > 5 && tx < 95))
      {Serial1.write("8");
        menu='0';
        drawHomeScreen();
        delay (100);
        while(tft.touchDetect()){};}
    }
if (ty >= 180 && ty <= 310)////////////////////////////////////// BOX 1,2,3
{if ((tx > 100 && tx < 230))// Box00
  { menu='6';
    drawMenuInputMatkul();
    delay (100);
    while(tft.touchDetect()){};}
else if ((tx > 350 && tx < 480)// Box01
  {menu='7';
    drawMenuInputMatkul();
    delay (100);
    while(tft.touchDetect()){};}
else if ((tx > 600 && tx < 730))// Box02
  {menu='8';
    drawMenuInputMatkul();
    delay (100);
    while(tft.touchDetect()){};
  }}
} //Batas MENU E
if (menu=='F') //Buka Box 00
{if (tft.touchDetect())
  { tft.touchReadPixel (&tx, &ty);
    tx=800-tx;ty=480-ty;
    if (ty >= 7 && ty <= 47) // Tombol LOGOUT
    {if ((tx > 5 && tx < 95))
      { Serial1.write("8");
        digitalWrite (relay_Box00, HIGH);
        atahun=0000;
        EEPROM.write(addrjj_box00, 0);
        EEPROM.write(addrmm_box00, 0);
      }
    }
  }
}

```



```

        EEPROM.write(addrhh_box00, 0);
        EEPROM.write(addrbb_box00, 0);
        menu='0';
        drawHomeScreen();
        delay (100);
        while(tft.touchDetect()){};
    }}}
} // Batas MENU F
if (menu=='G') //Buka Box01
{if (tft.touchDetect())
    { tft.touchReadPixel (&tx, &ty);
      tx=800-tx;ty=480-ty;
      if (ty >= 7 && ty <= 47) // Tombol LOGOUT
      {if ((tx > 5 && tx < 95))
          { Serial1.write("8");
            digitalWrite (relay_Box01, HIGH);
            atahun=0000;
            EEPROM.write(addrjj_box01, 0);
            EEPROM.write(addrmm_box01, 0);
            EEPROM.write(addrhh_box01, 0);
            EEPROM.write(addrbb_box01, 0);
            menu='0';
            drawHomeScreen();
            delay (100);
            while(tft.touchDetect()){};
          }}}
} //Batas MENU G
if (menu=='H') //Buka Box 02
{if (tft.touchDetect())
    { tft.touchReadPixel (&tx, &ty);
      tx=800-tx;ty=480-ty;
      if (ty >= 7 && ty <= 47) // Tombol LOGOUT
      {if ((tx > 5 && tx < 95))
          { Serial1.write("8");
            digitalWrite (relay_Box02, HIGH);
            atahun=0000;
            EEPROM.write(addrjj_box02, 0);
            EEPROM.write(addrmm_box02, 0);
            EEPROM.write(addrhh_box02, 0);
            EEPROM.write(addrbb_box02, 0);
            menu='0';
            drawHomeScreen();
            delay (100);
            while(tft.touchDetect()){};
          }}}
} //MENU H
} /////////////////////////////////////////////////// Batas Void
Loop
void box_setup ()
{pinMode (sensor_Box00, INPUT_PULLUP);
  pinMode (sensor_Box01, INPUT_PULLUP);
  pinMode (sensor_Box02, INPUT_PULLUP);
  pinMode (relay_Box00, OUTPUT);
  pinMode (relay_Box01, OUTPUT);
  pinMode (relay_Box02, OUTPUT);
  servo_Box00.attach(25);
  servo_Box01.attach(33);
  servo_Box02.attach(41); }
void drawHomeScreen() /////////////////////////////////////////////////// MENU 0
{
  //Title
  tft.changeMode(GRAPHIC);
  tft.fillRect(RA8875_BLACK);

```

```

tft.fillRect(0,10,10,450,RA8875_WHITE); //Bingkai putih left
tft.fillRect(790,0,10,450,RA8875_WHITE); //Bingkai putih right
tft.fillRect(0,0,800,30,RA8875_WHITE); //Bingkai putih up
tft.fillRect(0,450,800,10,RA8875_WHITE); //Bingkai putih down
tft.changeMode(TEXT);
tft.setTextColor(RA8875_BLACK);
tft.setFontScale(1);
tft.setCursor(320,0);
tft.print("STATUS BOX");
tft.fillRoundRect(70,70,140,50,10,RA8875_WHITE);
tft.fillRoundRect(75,75,130,40,10,RA8875_RED);
tft.setTextColor(RA8875_WHITE);
tft.setFontScale(1);
tft.setCursor(95,78);
tft.print("BOX 00");
tft.fillRoundRect(330,70,140,50,10,RA8875_WHITE);
tft.fillRoundRect(335,75,130,40,10,RA8875_YELLOW);
tft.setTextColor(RA8875_BLACK);
tft.setFontScale(1);
tft.setCursor(355,78);
tft.print("BOX 01");
tft.fillRoundRect(590,70,140,50,10,RA8875_WHITE);
tft.fillRoundRect(595,75,130,40,10,RA8875_BLUE);
tft.setTextColor(RA8875_WHITE);
tft.setFontScale(1);
tft.setCursor(615,78);
tft.print("BOX 02");
tft.fillRect(270,30,2,420,RA8875_WHITE);
tft.fillRect(530,30,2,420,RA8875_WHITE);
tft.setTextColor(RA8875_WHITE);
tft.setFontScale(0);
tft.setCursor(90,150);
tft.print("MATA KULIAH:");
tft.setCursor(100,270);
tft.print("DEADLINE:");
tft.setCursor(355,150);
tft.print("MATA KULIAH:");
tft.setCursor(365,270);
tft.print("DEADLINE:");
tft.setCursor(620,150);
tft.print("MATA KULIAH:");
tft.setCursor(630,270);
tft.print("DEADLINE:");
tft.setTextColor(RA8875_RED);
tft.setFontScale(1);
tft.setCursor(55,170);
EEPROM.readBlock<char>(100,_get00,10);
String print00=String(_get00);
tft.print(print00);
tft.setTextColor(RA8875_YELLOW);
tft.setFontScale(1);
tft.setCursor(290,170);
EEPROM.readBlock<char>(100,_get01,10);
String print01=String(_get01);
tft.print(print01);
tft.setTextColor(RA8875_BLUE);
tft.setFontScale(1);
tft.setCursor(575,170);
EEPROM.readBlock<char>(100,_get02,10);
String print02=String(_get02);
tft.print(print02);
tft.changeMode(GRAPHIC);
tft.fillRect(2,2,60,25,RA8875_BLACK);

```

```

tft.changeMode(TEXT);
tft.setTextColor(RA8875_WHITE);
tft.setFontScale(0);
tft.setCursor(7,5);
tft.print("ADMIN");
tft.changeMode(GRAPHIC);
tft.fillRect(70,2,60,25,RA8875_BLACK);
tft.changeMode(TEXT);
tft.setTextColor(RA8875_WHITE);
tft.setFontScale(0);
tft.setCursor(85,5);
tft.print("USER");}
void drawMenuAdmin()
{
tft.changeMode(GRAPHIC);
tft.fillScreen(RA8875_BLACK);
tft.fillRect(0,10,10,450,RA8875_WHITE); //Bingkai putih left
tft.fillRect(790,0,10,450,RA8875_WHITE); //Bingkai putih right
tft.fillRect(0,0,800,30,RA8875_WHITE); //Bingkai putih up
tft.fillRect(0,450,800,10,RA8875_WHITE); //Bingkai putih down
tft.changeMode(TEXT);
tft.setTextColor(RA8875_BLACK);
tft.setFontScale(1);
tft.setCursor(322,0);
tft.print("MENU ADMIN");
tft.changeMode(GRAPHIC);
tft.fillRoundRect(250,80,300,50,10,RA8875_RED);
tft.setTextColor(RA8875_BLACK);
tft.setFontScale(1);
tft.setCursor(312,85);
tft.print("TAMBAH USER");
tft.changeMode(GRAPHIC);
tft.fillRoundRect(250,170,300,50,10,RA8875_YELLOW);
tft.setTextColor(RA8875_BLACK);
tft.setFontScale(1);
tft.setCursor(323,178);
tft.print("HAPUS USER");
tft.changeMode(GRAPHIC);
tft.fillRoundRect(250,260,300,50,10,RA8875_GREEN);
tft.setTextColor(RA8875_BLACK);
tft.setFontScale(1);
tft.setCursor(337,270);
tft.print("EDIT BOX");
tft.changeMode(GRAPHIC);
tft.fillRoundRect(250,350,300,50,10,RA8875_BLUE);
tft.setTextColor(RA8875_BLACK);
tft.setFontScale(1);
tft.setCursor(337,358);
tft.print("BUKA BOX");}
void drawMenuUser()
{tft.changeMode(GRAPHIC);
tft.fillScreen(RA8875_BLACK);
tft.fillRect(0,10,10,450,RA8875_WHITE); //Bingkai putih left
tft.fillRect(790,0,10,450,RA8875_WHITE); //Bingkai putih right
tft.fillRect(0,0,800,30,RA8875_WHITE); //Bingkai putih up
tft.fillRect(0,450,800,10,RA8875_WHITE); //Bingkai putih down
tft.changeMode(TEXT);
tft.setTextColor(RA8875_BLACK);
tft.setFontScale(1);
tft.setCursor(330,0);
tft.print("MENU USER ");
tft.changeMode(GRAPHIC);
tft.fillRoundRect(250,150,300,50,10,RA8875_RED);

```

```

tft.setTextColor(RA8875_BLACK);
tft.setFontScale(1);
tft.setCursor(335,155);
tft.print("PAKAI BOX");
tft.changeMode(GRAPHIC);
tft.fillRect(250,265,300,50,10,RA8875_BLUE);
tft.setTextColor(RA8875_BLACK);
tft.setFontScale(1);
tft.setCursor(340,275);
tft.print("BUKA BOX");}
void drawMenuPilihBox ()
{ tft.changeMode(GRAPHIC);
  tft.fillScreen(RA8875_BLACK);
  tft.fillRect(0,10,10,450,RA8875_WHITE); //Bingkai putih left
  tft.fillRect(790,0,10,450,RA8875_WHITE); //Bingkai putih right
  tft.fillRect(0,0,800,30,RA8875_WHITE); //Bingkai putih up
  tft.fillRect(0,450,800,10,RA8875_WHITE); //Bingkai putih down
  tft.changeMode(TEXT);
  tft.setTextColor(RA8875_BLACK);
  tft.setFontScale(1);
  tft.setCursor(300,0);
  tft.print("PAKAI BOX");
  tft.fillRect(100,180,130,130,RA8875_RED); //box1
  tft.fillRect(350,180,130,130,RA8875_YELLOW); //box2
  tft.fillRect(600,180,130,130,RA8875_BLUE); //box3
  tft.setTextColor(RA8875_BLACK);
  tft.setFontScale(2);
  tft.setCursor(140,220);
  tft.print("01");
  tft.setCursor(390,220);
  tft.print("02");
  tft.setCursor(640,220);
  tft.print("03");}

void drawMenuBukaBox ()//////////////////////////////////// MENU
5
{
  //Title
  tft.changeMode(GRAPHIC);
  tft.fillScreen(RA8875_BLACK);
  tft.fillRect(0,10,10,450,RA8875_WHITE); //Bingkai putih left
  tft.fillRect(790,0,10,450,RA8875_WHITE); //Bingkai putih right
  tft.fillRect(0,0,800,30,RA8875_WHITE); //Bingkai putih up
  tft.fillRect(0,450,800,10,RA8875_WHITE); //Bingkai putih down
  tft.changeMode(TEXT);
  tft.setTextColor(RA8875_BLACK);
  tft.setFontScale(1);
  tft.setCursor(300,0);
  tft.print("BUKA BOX");
  tft.fillRect(100,180,130,130,RA8875_RED); //box00
  tft.fillRect(350,180,130,130,RA8875_YELLOW); //box01
  tft.fillRect(600,180,130,130,RA8875_BLUE); //box02
  tft.setTextColor(RA8875_BLACK);
  tft.setFontScale(2);
  tft.setCursor(140,220);
  tft.print("00");
  tft.setCursor(390,220);
  tft.print("01");
  tft.setCursor(640,220);
  tft.print("02");}
void drawMenuDataTugasMasuk ()
{ tft.changeMode(GRAPHIC);
  tft.fillScreen(RA8875_BLACK);

```

```

tft.fillRect(0,10,10,450,RA8875_WHITE); //Bingkai putih left
tft.fillRect(790,0,10,450,RA8875_WHITE); //Bingkai putih right
tft.fillRect(0,0,800,30,RA8875_WHITE); //Bingkai putih up
tft.fillRect(0,450,800,10,RA8875_WHITE); //Bingkai putih down
tft.changeMode(TEXT);
tft.setTextColor(RA8875_BLACK);
tft.setFontScale(1);
tft.setCursor(290,0);
tft.print("DATA TUGAS MASUK");}

```

```

void drawKeyboard ()
{ tft.changeMode(GRAPHIC);
  tft.fillScreen(RA8875_BLACK);
  tft.setTextColor(RA8875_WHITE);
  tft.setFontScale(3);
  tft.fillRoundRect (5, 160, 792, 300,10, RA8875_WHITE); //Output
  tft.fillRoundRect (10, 165, 65, 55,10, 0xC0C0C0); //0
  tft.fillRoundRect (10+(80*1), 165, 65, 55,10, 0xC0C0C0); //1
  tft.fillRoundRect (10+(80*2), 165, 65, 55,10, 0xC0C0C0); //2
  tft.fillRoundRect (10+(80*3), 165, 65, 55,10, 0xC0C0C0); //3
  tft.fillRoundRect (10+(80*4), 165, 65, 55,10, 0xC0C0C0); //4
  tft.fillRoundRect (10+(80*5), 165, 65, 55,10, 0xC0C0C0); //5
  tft.fillRoundRect (10+(80*6), 165, 65, 55,10, 0xC0C0C0); //6
  tft.fillRoundRect (10+(80*7), 165, 65, 55,10, 0xC0C0C0); //7
  tft.fillRoundRect (10+(80*8), 165, 65, 55,10, 0xC0C0C0); //8
  tft.fillRoundRect (10+(80*9), 165, 61, 55,10, 0xC0C0C0); //9
  tft.fillRoundRect (10, 225, 65, 55,10, (0x192, 0x192, 0x192)); //Q
  tft.fillRoundRect (10+(80*1), 225, 65, 55,10, (0x192, 0x192, 0x192)); //W
  tft.fillRoundRect (10+(80*2), 225, 65, 55,10, (0x192, 0x192, 0x192)); //E
  tft.fillRoundRect (10+(80*3), 225, 65, 55,10, (0x192, 0x192, 0x192)); //R
  tft.fillRoundRect (10+(80*4), 225, 65, 55,10, (0x192, 0x192, 0x192)); //T
  tft.fillRoundRect (10+(80*5), 225, 65, 55, 10, (0x192, 0x192, 0x192)); //Y
  tft.fillRoundRect (10+(80*6), 225, 65, 55,10, (0x192, 0x192, 0x192)); //U
  tft.fillRoundRect (10+(80*7), 225, 65, 55,10, (0x192, 0x192, 0x192)); //I
  tft.fillRoundRect (10+(80*8), 225, 65, 55,10, (0x192, 0x192, 0x192)); //O
  tft.fillRoundRect (10+(80*9), 225, 61, 55,10, (0x192, 0x192, 0x192)); //P
  tft.fillRoundRect (10, 285, 65, 55,10, (0x192, 0x192, 0x192)); //A
  tft.fillRoundRect (10+(80*1), 285, 65, 55,10, (0x192, 0x192, 0x192)); //S
  tft.fillRoundRect (10+(80*2), 285, 65, 55,10, (0x192, 0x192, 0x192)); //D
  tft.fillRoundRect (10+(80*3), 285, 65, 55,10, (0x192, 0x192, 0x192)); //F
  tft.fillRoundRect (10+(80*4), 285, 65, 55,10, (0x192, 0x192, 0x192)); //G
  tft.fillRoundRect (10+(80*5), 285, 65, 55,10, (0x192, 0x192, 0x192)); //H
  tft.fillRoundRect (10+(80*6), 285, 65, 55,10, (0x192, 0x192, 0x192)); //J
  tft.fillRoundRect (10+(80*7), 285, 65, 55,10, (0x192, 0x192, 0x192)); //K
  tft.fillRoundRect (10+(80*8), 285, 65, 55,10, (0x192, 0x192, 0x192)); //L
  tft.fillRoundRect (10+(80*9), 285, 61, 55,10, (0x192, 0x192, 0x192)); //.
  tft.fillRoundRect (90, 345, 65, 55,10, (0x192, 0x192, 0x192)); //Z
  tft.fillRoundRect (90+(80*1), 345, 65, 55,10, (0x192, 0x192, 0x192)); //X
  tft.fillRoundRect (90+(80*2), 345, 65, 55,10, (0x192, 0x192, 0x192)); //C
  tft.fillRoundRect (90+(80*3), 345, 65, 55,10, (0x192, 0x192, 0x192)); //V
  tft.fillRoundRect (90+(80*4), 345, 65, 55,10, (0x192, 0x192, 0x192)); //B
  tft.fillRoundRect (90+(80*5), 345, 65, 55,10, (0x192, 0x192, 0x192)); //N
  tft.fillRoundRect (90+(80*6), 345, 65, 55,10, (0x192, 0x192, 0x192)); //M
  tft.setTextColor(RA8875_WHITE);
  tft.setCursor(26,158);
  tft.print("0");
  tft.setCursor(26+(80*1),158);
  tft.print("1");
  tft.setCursor(26+(80*2),158);
  tft.print("2");
  tft.setCursor(26+(80*3),158);
  tft.print("3");
  tft.setCursor(26+(80*4),158);
}

```

```
tft.print("4");
tft.setCursor(26+(80*5),158);
tft.print("5");
tft.setCursor(26+(80*6),158);
tft.print("6");
tft.setCursor(26+(80*7),158);
tft.print("7");
tft.setCursor(26+(80*8),158);
tft.print("8");
tft.setCursor(26+(80*9),158);
tft.print("9");
tft.setCursor(26,217);
tft.print("Q");
tft.setCursor(26+(80*1),217);
tft.print("W");
tft.setCursor(26+(80*2),217);
tft.print("E");
tft.setCursor(26+(80*3),217);
tft.print("R");
tft.setCursor(26+(80*4),217);
tft.print("T");
tft.setCursor(26+(80*5),217);
tft.print("Y");
tft.setCursor(26+(80*6),217);
tft.print("U");
tft.setCursor(26+(80*7),217);
tft.print("I");
tft.setCursor(26+(80*8),217);
tft.print("O");
tft.setCursor(26+(80*9),217);
tft.print("P");
tft.setCursor(26,278);
tft.print("A");
tft.setCursor(26+(80*1),278);
tft.print("S");
tft.setCursor(26+(80*2),278);
tft.print("D");
tft.setCursor(26+(80*3),278);
tft.print("F");
tft.setCursor(26+(80*4),278);
tft.print("G");
tft.setCursor(26+(80*5),278);
tft.print("H");
tft.setCursor(26+(80*6),278);
tft.print("J");
tft.setCursor(26+(80*7),278);
tft.print("K");
tft.setCursor(26+(80*8),278);
tft.print("L");
tft.setCursor(26+(80*9),278);
tft.print(".");
tft.setCursor(107,337);
tft.print("Z");
tft.setCursor(107+(80*1),337);
tft.print("X");
tft.setCursor(107+(80*2),337);
tft.print("C");
tft.setCursor(107+(80*3),337);
tft.print("V");
tft.setCursor(107+(80*4),337);
tft.print("B");
tft.setCursor(107+(80*5),337);
tft.print("N");
```

```

tft.setCursor(107+(80*6),337);
tft.print("M");
tft.fillRect(90,405,548,50,10,RA8875_BLACK);//SPACEBAR
tft.fillRect(90+(80*7),345,140,55,10,RA8875_BLACK);//CLEAR
tft.setTextColor(RA8875_WHITE);
tft.setFontScale(2);
tft.setCursor(290,405);
tft.print("SPACEBAR");
tft.setFontScale(1);
tft.setCursor(145+(80*7),355);
tft.print("CLEAR");
tft.fillRect(665,370,35,5,RA8875_WHITE);//panah1
tft.fillTriangle(670,390,653,370,670,355,RA8875_WHITE);//panah2}
void drawMenuInputMatkul()
{ drawKeyboard();
  tft.setTextColor(RA8875_WHITE);
  tft.setFontScale(3);
  tft.setCursor(110,10);
  tft.print("Masukkan Mata Kuliah");
  tft.fillRect(130,80,470,60,RA8875_WHITE);///ISI NAMA
  tft.fillRect(620,80,80,60,RA8875_WHITE);/// ENTER
  ///tft.fillRect(625,85,70,50,RA8875_BLACK);/// ENTER
  tft.fillRect(630,100,40,20,RA8875_BLACK);//panah1
  tft.fillTriangle(690,110,650,85,650,135,RA8875_BLACK);//panah2}
void drawMenuDeadline()
{ tft.changeMode(GRAPHIC);
  tft.fillRect(RA8875_BLACK);
  tft.fillRect(0,10,10,450,RA8875_WHITE); //Bingkai putih left
  tft.fillRect(790,0,10,450,RA8875_WHITE); //Bingkai putih right
  tft.fillRect(0,0,800,30,RA8875_WHITE); //Bingkai putih up
  tft.fillRect(0,450,800,10,RA8875_WHITE); //Bingkai putih down
  tft.changeMode(TEXT);
  tft.setTextColor(RA8875_BLACK);
  tft.setFontScale(1);
  tft.setCursor(300,0);
  tft.print("DEADLINE SYSTEM");
  tft.fillRect(90,110,140,100,10,RA8875_WHITE);///jj/mm
  tft.fillRect(100,150,50,50,10,RA8875_RED);///jj
  tft.fillRect(170,150,50,50,10,RA8875_YELLOW);///mm
  tft.setTextColor(RA8875_BLACK);
  tft.setFontScale(3);
  tft.setCursor(145,140);
  tft.print(":");
  tft.setFontScale(1);
  tft.setCursor(102,115);
  tft.print("jj : mm");
  tft.fillRect(300,110,280,100,10,RA8875_WHITE);///hh/bb/tt
  tft.fillRect(310,150,50,50,10,RA8875_GREEN);///hh
  tft.fillRect(390,150,50,50,10,RA8875_BLUE);///bb
  tft.fillRect(470,150,100,50,10,RA8875_MAGENTA);///tt
  tft.setTextColor(RA8875_BLACK);
  tft.setFontScale(3);
  tft.setCursor(360,140);
  tft.print("/");
  tft.setCursor(440,140);
  tft.print("/");
  tft.setFontScale(1);
  tft.setCursor(305,115);
  tft.print(" hh / bb / tt");
  tft.setFontScale(2);
  tft.fillRect(140,300,80,50,10,RA8875_RED);///up
  tft.fillRect(300,300,80,50,10,RA8875_BLUE);///down
  tft.fillRect(460,300,80,50,10,RA8875_YELLOW);///set

```

```

tft.setTextColor(RA8875_BLACK);
tft.setFontScale(1);
tft.setCursor(158,305);
tft.print("UP");
tft.setCursor(308,305);
tft.print("DOWN");
tft.setCursor(475,305);
tft.print("SET");}
void Keyboard_Is_Running_box00()
{ Serial.println("key 00 ready");
  tft.changeMode(TEXT);
  tft.setTextColor(RA8875_BLACK);
  tft.setFontScale(2);
  if (tft.touchDetect())
    {tft.touchReadPixel (&tx, &ty);
     tx=800-tx;ty=480-ty;
     if (ty >= 165 && ty <= 230)//////////////////////////////////////// Baris Angka 0-
9
     {if ((tx > 10 && tx < 75))//////////////////////////////////////// Tombol 0
     { Nama_box00+='0';
       tft.changeMode(TEXT);
       tft.setCursor(132,85);
       tft.print(Nama_box00);
       delay (100);
       while(tft.touchDetect()){};}
     else if ((tx > 90 && tx < 145))//////////////////////////////////////// Tombol
1
     { Nama_box00+='1';
       tft.changeMode(TEXT);
       tft.setCursor(132,85);
       tft.print(Nama_box00);
       delay (100);
       while(tft.touchDetect()){};}
     else if ((tx > 170 && tx < 225))//////////////////////////////////////// Tombol
2
     { Nama_box00+='2';
       tft.changeMode(TEXT);
       tft.setCursor(132,85);
       tft.print(Nama_box00);
       delay (100);
       while(tft.touchDetect()){};}
     else if ((tx > 10+(80*3) && tx < 10+(80*3)+55))//////////////////////////////////////// Tombol
3
     { Nama_box00+='3';
       tft.changeMode(TEXT);
       tft.setCursor(132,85);
       tft.print(Nama_box00);
       delay (100);
       while(tft.touchDetect()){};}
     else if ((tx > 10+(80*4) && tx < 10+(80*4)+55))//////////////////////////////////////// Tombol
4
     { Nama_box00+='4';
       tft.changeMode(TEXT);
       tft.setCursor(132,85);
       tft.print(Nama_box00);
       delay (100);
       while(tft.touchDetect()){};}
     else if ((tx > 10+(80*5) && tx < 10+(80*5)+55))//////////////////////////////////////// Tombol
5
     { Nama_box00+='5';
       tft.changeMode(TEXT);
       tft.setCursor(132,85);
       tft.print(Nama_box00);

```



```

        delay (100);
        while(tft.touchDetect()){};}
else if ((tx > 10+(80*6) && tx < 10+(80*6)+55))//////////////////// Tombol
6
    { Nama_box00+='6';
      tft.changeMode(TEXT);
      tft.setCursor(132,85);
      tft.print(Nama_box00);
      delay (100);
      while(tft.touchDetect()){};}
else if ((tx > 10+(80*7) && tx < 10+(80*7)+55))//////////////////// Tombol
7
    { Nama_box00+='7';
      tft.changeMode(TEXT);
      tft.setCursor(132,85);
      tft.print(Nama_box00);
      delay (100);
      while(tft.touchDetect()){};}
else if ((tx > 10+(80*8) && tx < 10+(80*8)+55))//////////////////// Tombol
8
    { Nama_box00+='8';
      tft.changeMode(TEXT);
      tft.setCursor(132,85);
      tft.print(Nama_box00);
      delay (100);
      while(tft.touchDetect()){};}
else if ((tx > 10+(80*9) && tx < 10+(80*9)+55))//////////////////// Tombol
9
    { Nama_box00+='9';
      tft.changeMode(TEXT);
      tft.setCursor(132,85);
      tft.print(Nama_box00);
      delay (100);
      while(tft.touchDetect()){};}
}//////////////////// End Baris 0-
9
if (ty >= 240 && ty <= 290)//////////////////// Baris Huruf Q-
P
{if ((tx > 10 && tx < 75))//////////////////// Tombol Q
  { Nama_box00+='Q';
    tft.changeMode(TEXT);
    tft.setCursor(132,85);
    tft.print(Nama_box00);
    delay (100);
    while(tft.touchDetect()){};}
else if ((tx > 90 && tx < 145))//////////////////// Tombol
W
    { Nama_box00+='W';
      tft.changeMode(TEXT);
      tft.setCursor(132,85);
      tft.print(Nama_box00);
      delay (100);
      while(tft.touchDetect()){}; }
else if ((tx > 170 && tx < 225))//////////////////// Tombol
E
    { Nama_box00+='E';
      tft.changeMode(TEXT);
      tft.setCursor(132,85);
      tft.print(Nama_box00);
      delay (100);
      while(tft.touchDetect()){};}
else if ((tx > 10+(80*3) && tx < 10+(80*3)+55))//////////////////// Tombol
R

```

```

{ Nama_box00+='R';
  tft.changeMode(TEXT);
  tft.setCursor(132,85);
  tft.print(Nama_box00);
  delay(100);
  while(tft.touchDetect()){};}
else if ((tx > 10+(80*4) && tx < 10+(80*4)+55))//////////////////// Tombol
T
{ Nama_box00+='T';
  tft.changeMode(TEXT);
  tft.setCursor(132,85);
  tft.print(Nama_box00);
  delay(100);
  while(tft.touchDetect()){};}
else if ((tx > 10+(80*5) && tx < 10+(80*5)+55))//////////////////// Tombol
Y
{ Nama_box00+='Y';
  tft.changeMode(TEXT);
  tft.setCursor(132,85);
  tft.print(Nama_box00);
  delay(100);
  while(tft.touchDetect()){};}
else if ((tx > 10+(80*6) && tx < 10+(80*6)+55))//////////////////// Tombol
U
{ Nama_box00+='U';
  tft.changeMode(TEXT);
  tft.setCursor(132,85);
  tft.print(Nama_box00);
  delay(100);
  while(tft.touchDetect()){};}
else if ((tx > 10+(80*7) && tx < 10+(80*7)+55))//////////////////// Tombol
I
{ Nama_box00+='I';
  tft.changeMode(TEXT);
  tft.setCursor(132,85);
  tft.print(Nama_box00);
  delay(100);
  while(tft.touchDetect()){};}
else if ((tx > 10+(80*8) && tx < 10+(80*8)+55))//////////////////// Tombol
O
{ Nama_box00+='O';
  tft.changeMode(TEXT);
  tft.setCursor(132,85);
  tft.print(Nama_box00);
  delay(100);
  while(tft.touchDetect()){};}
else if ((tx > 10+(80*9) && tx < 10+(80*9)+55))//////////////////// Tombol P
{ Nama_box00+='P';
  tft.changeMode(TEXT);
  tft.setCursor(132,85);
  tft.print(Nama_box00);
  delay(100);
  while(tft.touchDetect()){};}
}//////////////////// End Baris 0-
9
if (ty >= 300 && ty <= 350)//////////////////// Baris Huruf A-
L
{if ((tx > 38 && tx < 93))//////////////////// Tombol A
{ Nama_box00+='A';
  tft.changeMode(TEXT);
  tft.setCursor(132,85);
  tft.print(Nama_box00);

```

```

    delay (100);
    while(tft.touchDetect()){};}
else if ((tx >38+(80*1) && tx < 38+(80*1)+55))//////////////////// Tombol
S
{ Nama_box00+='S';
  tft.changeMode(TEXT);
  tft.setCursor(132,85);
  tft.print(Nama_box00);
  delay (100);
  while(tft.touchDetect()){};}
else if ((tx >38+(80*2) && tx < 38+(80*2)+55))//////////////////// Tombol
D
{ Nama_box00+='D';
  tft.changeMode(TEXT);
  tft.setCursor(132,85);
  tft.print(Nama_box00);
  delay (100);
  while(tft.touchDetect()){};}
else if ((tx >38+(80*3) && tx < 38+(80*3)+55))//////////////////// Tombol
F
{ Nama_box00+='F';
  tft.changeMode(TEXT);
  tft.setCursor(132,85);
  tft.print(Nama_box00);
  delay (100);
  while(tft.touchDetect()){};}
else if ((tx >38+(80*4) && tx < 38+(80*4)+55))//////////////////// Tombol
G
{ Nama_box00+='G';
  tft.changeMode(TEXT);
  tft.setCursor(132,85);
  tft.print(Nama_box00);
  delay (100);
  while(tft.touchDetect()){};}
else if ((tx >38+(80*5) && tx < 38+(80*5)+55))//////////////////// Tombol
H
{ Nama_box00+='H';
  tft.changeMode(TEXT);
  tft.setCursor(132,85);
  tft.print(Nama_box00);
  delay (100);
  while(tft.touchDetect()){};}
else if ((tx >38+(80*6) && tx < 38+(80*6)+55))//////////////////// Tombol
J
{ Nama_box00+='J';
  tft.changeMode(TEXT);
  tft.setCursor(132,85);
  tft.print(Nama_box00);
  delay (100);
  while(tft.touchDetect()){};}
else if ((tx >38+(80*7) && tx < 38+(80*7)+55))//////////////////// Tombol
K
{ Nama_box00+='K';
  tft.changeMode(TEXT);
  tft.setCursor(132,85);
  tft.print(Nama_box00);
  delay (100);
  while(tft.touchDetect()){};}
else if ((tx >38+(80*8) && tx < 38+(80*8)+55))//////////////////// Tombol
L
{ Nama_box00+='L';
  tft.changeMode(TEXT);
  tft.setCursor(132,85);

```

```

    tft.print>Nama_box00);
    delay (100);
    while(tft.touchDetect()){};}
else if ((tx >38+(80*9) && tx < 38+(80*9)+55))//////////////////// Tombol
L
    {>Nama_box00+='.';
    tft.changeMode(TEXT);
    tft.setCursor(132,85);
    tft.print>Nama_box00);
    delay (100);
    while(tft.touchDetect()){};}
}//////////////////// End Baris A-
L
if (ty >= 350 && ty <= 410)//////////////////// Baris Huruf Z-
M
    {if ((tx > 90 && tx < 145))//////////////////// Tombol Z
    {>Nama_box00+='Z';
    tft.changeMode(TEXT);
    tft.setCursor(132,85);
    tft.print>Nama_box00);
    delay (100);
    while(tft.touchDetect()){};}
else if ((tx > 10+(80*2) && tx < 10+(80*2)+55))//////////////////// Tombol
X
    {>Nama_box00+='X';
    tft.changeMode(TEXT);
    tft.setCursor(132,85);
    tft.print>Nama_box00);
    delay (100);
    while(tft.touchDetect()){};}
else if ((tx > 10+(80*3) && tx < 10+(80*3)+55))//////////////////// Tombol
C
    {>Nama_box00+='C';
    tft.changeMode(TEXT);
    tft.setCursor(132,85);
    tft.print>Nama_box00);
    delay (100);
    while(tft.touchDetect()){};}
else if ((tx > 10+(80*4) && tx < 10+(80*4)+55))//////////////////// Tombol
V
    {>Nama_box00+='V';
    tft.changeMode(TEXT);
    tft.setCursor(132,85);
    tft.print>Nama_box00);
    delay (100);
    while(tft.touchDetect()){};}
else if ((tx > 10+(80*5) && tx < 10+(80*5)+55))//////////////////// Tombol
B
    {>Nama_box00+='B';
    tft.changeMode(TEXT);
    tft.setCursor(132,85);
    tft.print>Nama_box00);
    delay (100);
    while(tft.touchDetect()){};}
else if ((tx > 10+(80*6) && tx < 10+(80*6)+55))//////////////////// Tombol
N
    {>Nama_box00+='N';
    tft.changeMode(TEXT);
    tft.setCursor(132,85);
    tft.print>Nama_box00);
    delay (100);
    while(tft.touchDetect()){};}

```

```

else if ((tx > 10+(80*7) && tx < 10+(80*7)+55))//////////////////// Tombol
M
    { Nama_box00+='M';
      tft.changeMode(TEXT);
      tft.setCursor(132,85);
      tft.print(Nama_box00);
      delay (100);
      while(tft.touchDetect()){};}
}//////////////////// End Baris Z-
M
if (ty >= 415 && ty <= 455)//////////////////// Tombol SPACE
{if ((tx > 100 && tx < 665))
  { Nama_box00+=' ';
    tft.changeMode(TEXT);
    tft.setCursor(132,85);
    tft.print(Nama_box00);
    delay (100);
    while(tft.touchDetect()){};}}

if (ty >= 345 && ty <= 345+140)//////////////////// Tombol CLEAR
{if ((tx > 90+(80*7) && tx < 38+(80*9)+55))
  { tft.changeMode(TEXT);
    tft.setTextColor(RA8875_WHITE);
    tft.setCursor(132,85);
    tft.print(Nama_box00);
    delay (1000);
    Nama_box00="";
    while(tft.touchDetect()){};}}

if (ty >= 80 && ty <= 140)//////////////////// Tombol
ENTER
{if ((tx > 620 && tx < 700))
  { if (Box_Aktif=='0'){
    Nama_box00.toCharArray(_put00,10);
    EEPROM.writeBlock<char>(100,_put00,10);
    menu='A';
    drawMenuDeadline();
    delay (100);
    while(tft.touchDetect()){};}
    if (Box_Aktif=='1'){
    Nama_box00.toCharArray(_put01,10);
    EEPROM.writeBlock<char>(200,_put01,10);
    menu='B';
    drawMenuDeadline();
    delay (100);
    while(tft.touchDetect()){};}
    if (Box_Aktif=='2'){
    Nama_box00.toCharArray(_put02,10);
    EEPROM.writeBlock<char>(300,_put02,10);
    menu='C';
    drawMenuDeadline();
    delay (100);
    while(tft.touchDetect()){};}} }
}
void rtc_setup()//////////////////// setting RTC
{
  if (! rtc.begin()) {
    Serial.println("Couldn't find RTC");
    while (1);}
  if (rtc.lostPower()) {
    Serial.println("RTC lost power, lets set the time!");
    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));}
}
void cekAlarm_box00()

```

```

{DateTime now = rtc.now();
  if ((jam == bjam) && (menit == bmenit)&& (tanggal == btanggal)&& (bulan == bbulan))
    {servo_Box00.write(0);}
}
void cekAlarm_box01()
{ DateTime now = rtc.now();
  if ((jam == cjam) && (menit == cmenit)&& (tanggal == ctanggal)&& (bulan == cbulan))
    {servo_Box01.write(0);}
}
void cekAlarm_box02()
{DateTime now = rtc.now();
  if ((jam == djam) && (menit == dmenit)&& (tanggal == dtanggal)&& (bulan == dbulan))
    {servo_Box02.write(0);}
}
void run_alarm_box00()////////////////////////////////////Set Alarm
{ bjam = EEPROM.read(addrjj_box00);
  bmenit = EEPROM.read(addrmm_box00);
  btanggal = EEPROM.read(addrhh_box00);
  bbulan = EEPROM.read(addrbb_box00);
  btahun = EEPROM.read(addrtt_box00);
  DateTime now = rtc.now();
  if (millis() - waktulama >= interval)
  { waktulama = millis();
    if (state == 0)
      {state = 1; }
    else if (state == 1)
      {state = 0;}

    if (bjam < 10 && bmenit < 10) {
      sprintf (buff, "0%d:0%d", bjam, bmenit);}
    if (bjam < 10 && bmenit >= 10) {
      sprintf (buff, "0%d:%d", bjam, bmenit);}
    if (bjam >= 10 && bmenit < 10) {
      sprintf (buff, "%d:0%d", bjam, bmenit);}
    if (bjam >= 10 && bmenit >= 10) {
      sprintf (buff, "%d:%d", bjam, bmenit);}
    if (btanggal < 10 && bbulan < 10) {
      sprintf (buff1, "0%d/0%d/", btanggal, bbulan);}
    if (btanggal < 10 && bbulan >= 10) {
      sprintf (buff1, "0%d/%d/", btanggal, bbulan);}
    if (btanggal >= 10 && bbulan < 10) {
      sprintf (buff1, "%d/0%d/", btanggal, bbulan);}
    if (btanggal >= 10 && bbulan >= 10) {
      sprintf (buff1, "%d/%d/", btanggal, bbulan);}
    tft.setTextColor(RA8875_RED);
    tft.setFontScale(1);
    tft.setCursor(50, 290);
    tft.print(buff1);
    tft.print(atahun);
    tft.setCursor(90,320);
    tft.print(buff);}
}
void run_alarm_box01()////////////////////////////////////Set Alarm
{ cjam = EEPROM.read(addrjj_box01);
  cmenit = EEPROM.read(addrmm_box01);
  ctanggal = EEPROM.read(addrhh_box01);
  cbulan = EEPROM.read(addrbb_box01);
  ctahun = EEPROM.read(addrtt_box01);
  DateTime now = rtc.now();
  if (millis() - waktulama >= interval)
  {waktulama = millis();
    if (state == 0)
      {state = 1;}
  }
}

```

```

else if (state == 1)
{state = 0;}
if (cjam < 10 && cmenit < 10) {
    sprintf (buff, "0%d:0%d", cjam, cmenit);}
if (cjam < 10 && cmenit >= 10) {
    sprintf (buff, "0%d:%d", cjam, cmenit);}
if (cjam >= 10 && cmenit < 10) {
    sprintf (buff, "%d:0%d", cjam, cmenit);}
if (cjam >= 10 && cmenit >= 10) {
    sprintf (buff, "%d:%d", cjam, cmenit);}
if (ctanggal < 10 && cbulan < 10) {
    sprintf (buff1, "0%d/0%d/", ctanggal, cbulan);}
if (ctanggal < 10 && cbulan >= 10) {
    sprintf (buff1, "0%d/%d/", ctanggal, cbulan);}
if (ctanggal >= 10 && cbulan < 10) {
    sprintf (buff1, "%d/0%d/", ctanggal, cbulan);}
if (ctanggal >= 10 && cbulan >= 10) {
    sprintf (buff1, "%d/%d/", ctanggal, cbulan);}
tft.setTextColor(RA8875_YELLOW);
tft.setFontScale(1);
tft.setCursor(320, 290);
tft.print(buff1);
tft.print(atahun);
tft.setCursor(360,320);
tft.print(buff);}
}

void run_alarm_box02()////////////////////////////////////Set Alarm
{ djam = EEPROM.read(addrjj_box02);
  dmenit = EEPROM.read(addrmm_box02);
  dtanggal = EEPROM.read(addrhh_box02);
  dbulan = EEPROM.read(addrbb_box02);
  dtahun = EEPROM.read(addrtt_box02);
  DateTime now = rtc.now();
  if (millis() - waktulama >= interval)
  { waktulama = millis();
    if (state == 0)
    {state = 1;}
    else if (state == 1)
    { state = 0; }
    if (djam < 10 && dmenit < 10) {
        sprintf (buff, "0%d:0%d", djam, dmenit);}
    if (djam < 10 && dmenit >= 10) {
        sprintf (buff, "0%d:%d", djam, dmenit);}
    if (djam >= 10 && dmenit < 10) {
        sprintf (buff, "%d:0%d", djam, dmenit);}
    if (djam >= 10 && dmenit >= 10) {
        sprintf (buff, "%d:%d", djam, dmenit);}
    if (dtanggal < 10 && dbulan < 10) {
        sprintf (buff1, "0%d/0%d/", dtanggal, dbulan);}
    if (dtanggal < 10 && dbulan >= 10) {
        sprintf (buff1, "0%d/%d/", dtanggal, dbulan);}
    if (dtanggal >= 10 && dbulan < 10) {
        sprintf (buff1, "%d/0%d/", dtanggal, dbulan);}
    if (dtanggal >= 10 && dbulan >= 10) {
        sprintf (buff1, "%d/%d/", dtanggal, dbulan);}
    tft.setTextColor(RA8875_BLUE);
    tft.setFontScale(1);
    tft.setCursor(580, 290);
    tft.print(buff1);
    tft.print(atahun);
    tft.setCursor(620,320);
    tft.print(buff);}
}

```

```

void tamjam()////////////////////////////////////Tampilan Jam
{ if (ajam >= 24)ajam = 0;
  if (ajam == 255)ajam = 23;
  if (ajam < 10 ){sprintf (buff, "0%d", ajam);}
  if (ajam >= 10) {sprintf (buff, "%d", ajam);}
  tft.setTextColor(RA8875_BLACK);
  tft.setFontScale(2);
  tft.setCursor(102,150);
  tft.print(buff);}
void tamnit()////////////////////////////////////Tampilan Menit
{ if (amenit >= 59)amenit = 0;
  if (amenit == 255)amenit = 59;
  if (amenit < 10 ){sprintf (buff, "0%d", amenit);}
  if (amenit >= 10) {sprintf (buff, "%d", amenit);}
  tft.setTextColor(RA8875_BLACK);
  tft.setFontScale(2);
  tft.setCursor(170,150);
  tft.print(buff);}
void tamri()////////////////////////////////////Tampilan Hari
{
  if (atanggal >= 31)atanggal = 0;
  if (atanggal == 255) atanggal = 31;
  if (atanggal < 10 ){sprintf (buff, "0%d", atanggal);}
  if (atanggal >= 10) {sprintf (buff, "%d", atanggal);}
  tft.setTextColor(RA8875_BLACK);
  tft.setFontScale(2);
  tft.setCursor(312,150);
  tft.print(buff);
}
void tambul()////////////////////////////////////Tampilan Bulan
{if (abulan >= 12)abulan = 0;
  if (abulan == 255)abulan = 31;
  if (abulan < 10 ){sprintf (buff, "0%d", abulan);}
  if (abulan >= 10) {sprintf (buff, "%d", abulan);}
  if (abulan >= 12){abulan = 0;}
  tft.setTextColor(RA8875_BLACK);
  tft.setFontScale(2);
  tft.setCursor(392,150);
  tft.print(buff);}
void tamhun()////////////////////////////////////Tampilan Tahun
{if (atahun >= 2030)atahun = 2019;
  if (atahun <= 2019)atahun = 2019;
  tft.setTextColor(RA8875_BLACK);
  tft.setFontScale(2);
  tft.setCursor(472,150);
  tft.print(atahun);}
void Alarm_Is_Running_Box00 ()
{if(!tft.touchDetect())////////////////////////////////////jika tidak ditekan
  {tamjam();tamnit();tamri();tambul();tamhun();}
  if (tft.touchDetect())////////////////////////////////////jika ditekan
  { tft.touchReadPixel (&tx, &ty);
    tx=800-tx;ty=480-ty;
    if (ty >= 150 && ty <= 200)
    {if ((tx > 100 && tx < 150))////////////////////////////////PB jj
      {drawMenuDeadline();
        tamnit(); tamri();tambul();tamhun();
        up_Down='0';
        delay(100); while(tft.touchDetect()){};}
      if ((tx > 170 && tx < 220))////////////////////////////////PB mm
      {drawMenuDeadline();
        tamjam(); tamri();tambul();tamhun();
        up_Down='1';
        delay(100); while(tft.touchDetect()){};}
    }
}

```



```

        if (abulan == 255)abulan = 31;
        if (abulan < 10 ){sprintf (buff, "%d", abulan);}
        if (abulan >= 10) {sprintf (buff, "%d", abulan);}
        tft.fillRoundRect (390,150,50,50,10,RA8875_BLUE);
        tft.setTextColor(RA8875_BLACK);
        tft.setFontScale(2);
        tft.setCursor(392,150);
        tft.print(buff);delay(100);
        while (!tft.touchDetect()){delay(100);};
    }//down
} //up/down
} //dead=3
if(up_Down=='4')////////////////////up/down tahun
{tft.touchReadPixel (&tx, &ty);
tx=800-tx;ty=480-ty;
if (ty >= 300 && ty <= 350)
{if ((tx > 140 && tx < 220))////////////////////up
{
    atahun++;
    EEPROM.write(addrtt_box00, atahun);
    if (atahun >= 2030)atahun = 2019;
    tft.fillRoundRect (470,150,100,50,10,RA8875_MAGENTA);
    tft.setTextColor(RA8875_BLACK);
    tft.setFontScale(2);
    tft.setCursor(472,150);
    tft.print(atahun);delay(100);
    while (!tft.touchDetect()){delay(100);};
}
}
else if ((tx > 300 && tx < 380))////////////////////down
{
    atahun--;
    EEPROM.write(addrtt_box00, atahun);
    if (atahun <= 2019)atahun = 2019;
    tft.fillRoundRect (470,150,100,50,10,RA8875_MAGENTA);
    tft.setTextColor(RA8875_BLACK);
    tft.setFontScale(2);
    tft.setCursor(472,150);
    tft.print(atahun);delay(100);
    while (!tft.touchDetect()){delay(100);};
}
} //down
} //up/down
} //dead=4
if (ty >= 300 && ty <= 350)////////////////////PB SET
{if ((tx > 460 && tx < 550))
{
    servo_Box00.write(90);
    run_alarm_box00();
    menu='0';
    drawHomeScreen();
    delay(100); while(tft.touchDetect()){};
}
}
} //
} //void Alarn_Is_Running
void Alarm_Is_Running_Box01()
{
    if(!tft.touchDetect())////////////////////jika tidak ditekan
    {tamjam();tamnit();tamri();tambul();tamhun();}
    if (tft.touchDetect())////////////////////jika ditekan
    {tft.touchReadPixel (&tx, &ty);
    tx=800-tx;ty=480-ty;
    if (ty >= 150 && ty <= 200)
    {if ((tx > 100 && tx < 150))////////////////////PB jj
    {
        drawMenuDeadline();
        tamnit(); tamri();tambul();tamhun();
        up_Down='0';
        delay(100); while(tft.touchDetect()){}; }
    }
    if ((tx > 170 && tx < 220))////////////////////PB mm

```

```

{ drawMenuDeadline();
  tamjam(); tamri();tambul();tamhun();
  up_Down='1';
  delay(100); while(tft.touchDetect()){};}
if ((tx > 310 && tx < 360))////////////////////////////////////PB hh
{ drawMenuDeadline();
  tamjam();tamnit(); tambul(); tamhun();
  up_Down='2';
  delay(100); while(tft.touchDetect()){};}
if ((tx > 390 && tx < 440))////////////////////////////////////PB bb
{ drawMenuDeadline();
  tamjam();tamnit(); tamri();tamhun();
  up_Down='3';
  delay(100); while(tft.touchDetect()){};}
if ((tx > 470 && tx < 520))////////////////////////////////////PB tt
{ drawMenuDeadline();
  tamjam();tamnit(); tamri(); tambul();
  up_Down='4';
  delay(100); while(tft.touchDetect()){};}
if(up_Down=='0')/////////////////////////////////up/down jam
{ tft.touchReadPixel (&tx, &ty);
  tx=800-tx;ty=480-ty;
  if (ty >= 300 && ty <= 350)
  { if ((tx > 140 && tx < 220))/////////////////////////////////up
    { ajam++;
      EEPROM.write(addrjj_box01, ajam);
      if (ajam >= 24)ajam = 0;
      if (ajam < 10 ){sprintf (buff, "0%d", ajam);}
      if (ajam >= 10) {sprintf (buff, "%d", ajam);}
      tft.fillRoundRect(100,150,50,50,10,RA8875_RED);
      tft.setTextColor(RA8875_BLACK);
      tft.setFontScale(2);
      tft.setCursor(102,150);
      tft.print(buff);delay(100);
      while (!tft.touchDetect()){delay(100);}}
    else if ((tx > 300 && tx < 380))/////////////////////////////////down
    { ajam--;
      EEPROM.write(addrjj_box01, ajam);
      if (ajam == 255)ajam = 23;
      if (ajam < 10 ){sprintf (buff, "0%d", ajam);}
      if (ajam >= 10) {sprintf (buff, "%d", ajam);}
      tft.fillRoundRect(100,150,50,50,10,RA8875_RED);
      tft.setTextColor(RA8875_BLACK);
      tft.setFontScale(2);
      tft.setCursor(102,150);
      tft.print(buff);delay(100);
      while (!tft.touchDetect()){delay(100);}}
    }//down
  }//up/down
} //dead=0
if(up_Down=='1')/////////////////////////////////up/down menit
{ tft.touchReadPixel (&tx, &ty);
  tx=800-tx;ty=480-ty;
  if (ty >= 300 && ty <= 350)
  { if ((tx > 140 && tx < 220))/////////////////////////////////up
    { amenit++;
      EEPROM.write(addrmm_box01, amenit);
      if (amenit >= 59)amenit = 0;
      if (amenit < 10 ){sprintf (buff, "0%d", amenit);}
      if (amenit >= 10) {sprintf (buff, "%d", amenit);}
      tft.fillRoundRect(170,150,50,50,10,RA8875_YELLOW);
      tft.setTextColor(RA8875_BLACK);
      tft.setFontScale(2);
    }
  }
}

```

```

        tft.setCursor(170,150);
        tft.print(buff);delay(100);
        while (!tft.touchDetect()){delay(100);};}
else if ((tx > 300 && tx < 380))////////////////////down
{ amenit--;
  EEPROM.write(addrmm_box01, amenit);
  if (amenit == 255)amenit = 59;
  if (amenit < 10 ){sprintf (buff, "0%d", amenit);}
  if (amenit >= 10) {sprintf (buff, "%d", amenit);}
  tft.fillRoundRect(170,150,50,50,10,RA8875_YELLOW);
  tft.setTextColor(RA8875_BLACK);
  tft.setFontScale(2);
  tft.setCursor(170,150);
  tft.print(buff);delay(100);
  while (!tft.touchDetect()){delay(100);};
} //down
} //up/down
} //dead=1
if(up_Down=='2')////////////////////up/down tanggal
{ tft.touchReadPixel (&tx, &ty);
  tx=800-tx;ty=480-ty;
  if (ty >= 300 && ty <= 350)
  { if ((tx > 140 && tx < 220))////////////////////up
  { atanggal++;
    EEPROM.write(addrhh_box01, atanggal);
    if (atanggal >= 31)atanggal = 0;
    if (atanggal < 10 ){sprintf (buff, "0%d", atanggal);}
    if (atanggal >= 10) {sprintf (buff, "%d", atanggal);}
    tft.fillRoundRect(310,150,50,50,10,RA8875_GREEN);
    tft.setTextColor(RA8875_BLACK);
    tft.setFontScale(2);
    tft.setCursor(312,150);
    tft.print(buff);delay(100);
    while (!tft.touchDetect()){delay(100);};}
else if ((tx > 300 && tx < 380))////////////////////down
{ atanggal--;
  EEPROM.write(addrhh_box01, atanggal);
  if (atanggal == 255) atanggal = 31;
  if (atanggal < 10 ){sprintf (buff, "0%d", atanggal);}
  if (atanggal >= 10) {sprintf (buff, "%d", atanggal);}
  tft.fillRoundRect(310,150,50,50,10,RA8875_GREEN);
  tft.setTextColor(RA8875_BLACK);
  tft.setFontScale(2);
  tft.setCursor(312,150);
  tft.print(buff);delay(100);
  while (!tft.touchDetect()){delay(100);};
} //down
} //up/down
} //dead=2
if(up_Down=='3')////////////////////up/down bulan
{ tft.touchReadPixel (&tx, &ty);
  tx=800-tx;ty=480-ty;
  if (ty >= 300 && ty <= 350)
  { if ((tx > 140 && tx < 220))////////////////////up
  { abulan++;
    EEPROM.write(addrbb_box01, abulan);
    if (abulan >= 12)abulan = 0;
    if (abulan < 10 ){sprintf (buff, "0%d", abulan);}
    if (abulan >= 10) {sprintf (buff, "%d", abulan);}
    tft.fillRoundRect(390,150,50,50,10,RA8875_BLUE);
    tft.setTextColor(RA8875_BLACK);
    tft.setFontScale(2);
    tft.setCursor(392,150);

```

```

        tft.print(buff);delay(100);
        while (!tft.touchDetect()){delay(100);};
    }
else if ((tx > 300 && tx < 380))/////////////////////////////////down
{
    abulan--;
    EEPROM.write(addrbb_box01, abulan);
    if (abulan == 255)abulan = 31;
    if (abulan < 10 ){sprintf (buff, "%d", abulan);}
    if (abulan >= 10) {sprintf (buff, "%d", abulan);}
    tft.fillRoundRect(390,150,50,50,10,RA8875_BLUE);
    tft.setTextColor(RA8875_BLACK);
    tft.setFontScale(2);
    tft.setCursor(392,150);
    tft.print(buff);delay(100);
    while (!tft.touchDetect()){delay(100);};
} //down
} //up/down
} //dead=3
if (up_Down=='4')/////////////////////////////////up/down tahun
{
    tft.touchReadPixel (&tx, &ty);
    tx=800-tx;ty=480-ty;
    if (ty >= 300 && ty <= 350)
    {
        if ((tx > 140 && tx < 220))/////////////////////////////////up
        {
            atahun++;
            EEPROM.write(addrtt_box01, atahun);
            if (atahun >= 2030)atahun = 2019;
            tft.fillRoundRect(470,150,100,50,10,RA8875_MAGENTA);
            tft.setTextColor(RA8875_BLACK);
            tft.setFontScale(2);
            tft.setCursor(472,150);
            tft.print(atahun);delay(100);
            while (!tft.touchDetect()){delay(100);};
        }
        else if ((tx > 300 && tx < 380))/////////////////////////////////down
        {
            atahun--;
            EEPROM.write(addrtt_box01, atahun);
            if (atahun <= 2019)atahun = 2019;
            tft.fillRoundRect(470,150,100,50,10,RA8875_MAGENTA);
            tft.setTextColor(RA8875_BLACK);
            tft.setFontScale(2);
            tft.setCursor(472,150);
            tft.print(atahun);delay(100);
            while (!tft.touchDetect()){delay(100);};
        } //down
    } //up/down
} //dead=4
if (ty >= 300 && ty <= 350)/////////////////////////////////PB SET
{
    if ((tx > 460 && tx < 550))
    { run_alarm_box01();
      menu='0';
      drawHomeScreen();
      delay(100); while(tft.touchDetect()){};
    }
}
if (ty >= 300 && ty <= 350)/////////////////////////////////PB SET
{
    if ((tx > 460 && tx < 550))

```



```

{
    ajam++;
    EEPROM.write(addrjj_box02, ajam);
    if (ajam >= 24)ajam = 0;
    if (ajam < 10 ){sprintf (buff, "%d", ajam);}
    if (ajam >= 10) {sprintf (buff, "%d", ajam);}
    tft.fillRoundRect(100,150,50,50,10,RA8875_RED);
    tft.setTextColor(RA8875_BLACK);
    tft.setFontScale(2);
    tft.setCursor(102,150);
    tft.print(buff);delay(100);
    while (!tft.touchDetect()){delay(100);};
}

else if ((tx > 300 && tx < 380))////////////////////down
{
    ajam--;
    EEPROM.write(addrjj_box02, ajam);
    if (ajam == 255)ajam = 23;
    if (ajam < 10 ){sprintf (buff, "%d", ajam);}
    if (ajam >= 10) {sprintf (buff, "%d", ajam);}
    tft.fillRoundRect(100,150,50,50,10,RA8875_RED);
    tft.setTextColor(RA8875_BLACK);
    tft.setFontScale(2);
    tft.setCursor(102,150);
    tft.print(buff);delay(100);
    while (!tft.touchDetect()){delay(100);};
} //down
} //up/down
} //dead=0
if(up_Down=='1')////////////////////up/down menit
{
    tft.touchReadPixel (&tx, &ty);
    tx=800-tx;ty=480-ty;
    if (ty >= 300 && ty <= 350)
    {
        if ((tx > 140 && tx < 220))////////////////////up
        {
            amenit++;
            EEPROM.write(addrmm_box02, amenit);
            if (amenit >= 59)amenit = 0;
            if (amenit < 10 ){sprintf (buff, "%d", amenit);}
            if (amenit >= 10) {sprintf (buff, "%d", amenit);}

            tft.fillRoundRect(170,150,50,50,10,RA8875_YELLOW);
            tft.setTextColor(RA8875_BLACK);
            tft.setFontScale(2);
            tft.setCursor(170,150);
            tft.print(buff);delay(100);
            while (!tft.touchDetect()){delay(100);};
        }
    }
    else if ((tx > 300 && tx < 380))////////////////////down
    {
        amenit--;
        EEPROM.write(addrmm_box02, amenit);
        if (amenit == 255)amenit = 59;
        if (amenit < 10 ){sprintf (buff, "%d", amenit);}
        if (amenit >= 10) {sprintf (buff, "%d", amenit);}
        tft.fillRoundRect(170,150,50,50,10,RA8875_YELLOW);
        tft.setTextColor(RA8875_BLACK);
        tft.setFontScale(2);
        tft.setCursor(170,150);
        tft.print(buff);delay(100);
    }
}

```



```

EEPROM.write(addrbb_box02, abulan);
if (abulan == 255)abulan = 31;
if (abulan < 10 ){sprintf (buff, "0%d", abulan);}
if (abulan >= 10) {sprintf (buff, "%d", abulan);}
tft.fillRoundRect (390,150,50,50,10,RA8875_BLUE);
tft.setTextColor (RA8875_BLACK);
tft.setFontScale (2);
tft.setCursor (392,150);
tft.print (buff);delay (100);
while (!tft.touchDetect ()) {delay (100);};
} //down
} //up/down
} //dead=3
if (up_Down=='4') //up/down tahun
{
tft.touchReadPixel (&tx, &ty);
tx=800-tx;ty=480-ty;
if (ty >= 300 && ty <= 350)
{
if ((tx > 140 && tx < 220)) //up
{
atahun++;
EEPROM.write(addrtt_box02, atahun);
if (atahun >= 2030)atahun = 2019;
tft.fillRoundRect (470,150,100,50,10,RA8875_MAGENTA);
tft.setTextColor (RA8875_BLACK);
tft.setFontScale (2);
tft.setCursor (472,150);
tft.print (atahun);delay (100);
while (!tft.touchDetect ()) {delay (100);};
}
else if ((tx > 300 && tx < 380)) //down
{
atahun--;
EEPROM.write(addrtt_box02, atahun);
if (atahun <= 2019)atahun = 2019;
tft.fillRoundRect (470,150,100,50,10,RA8875_MAGENTA);
tft.setTextColor (RA8875_BLACK);
tft.setFontScale (2);
tft.setCursor (472,150);
tft.print (atahun);delay (100);
while (!tft.touchDetect ()) {delay (100);};
} //down
} //up/down
} //dead=4
if (ty >= 300 && ty <= 350) //PB SET
{
if ((tx > 460 && tx < 550))
{ servo_Box00.write (90);
run_alarm_box02 ();
menu='0';
drawHomeScreen ();
delay (100); while (tft.touchDetect ()){};
}
}
} //touch
} //void Alarn_Is_Running
void deteksi_tugas00 ()
{ if (digitalRead (sensor_Box00)==LOW)
{ DateTime now = rtc.now ();
tugas00++;
Serial.println ("Deteksi 00");
if (tugas00==1)

```

```

{ dataFile = SD.open("B00_bc1.txt", FILE_WRITE);
  if (dataFile)
  { Serial.println("opening B00_bc.txt");
    dataFile.print(tugas00);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  {Serial.println("error opening B00_bc.txt"); }
  delay(3000);
}
if(tugas00==2)
{ dataFile = SD.open("B00_bc2.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas00);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B00_bc2.txt"); }
  delay(3000);
}
if(tugas00==3)
{ dataFile = SD.open("B00_bc3.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas00);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else

```

```

    { Serial.println("error opening B00_bc3.txt"); }
    delay(3000);
}
if(tugas00==4)
{ dataFile = SD.open("B00_bc4.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas00);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B00_bc4.txt"); }
  delay(3000);
}
if(tugas00==5)
{ dataFile = SD.open("B00_bc5.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas00);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B00_bc5.txt"); }
  delay(3000);
}
if(tugas00==6)
{ dataFile = SD.open("B00_bc6.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas00);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
  }
}

```

```

        dataFile.println();
        dataFile.close(); }
    else
    { Serial.println("error opening B00_bc6.txt"); }
    delay(3000);
}
if(tugas00==7)
{ dataFile = SD.open("B00_bc7.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas00);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B00_bc7.txt"); }
  delay(3000);
}
if(tugas00==8)
{ dataFile = SD.open("B00_bc8.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas00);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B00_bc8.txt"); }
  delay(3000);
}
if(tugas00==9)
{ dataFile = SD.open("B00_bc9.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas00);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");

```

```

        dataFile.print(now.hour(), DEC);
        dataFile.print(':');
        dataFile.print(now.minute(), DEC);
        dataFile.println();
        dataFile.close(); }
    else
    { Serial.println("error opening B00_bc9.txt"); }
    delay(3000);
}
if(tugas00==10)
{ dataFile = SD.open("B00_bc10.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas00);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B00_bc0.txt"); }
  delay(3000);
}
if(tugas00==11)
{ dataFile = SD.open("B00_bc11.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas00);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B00_bc11.txt"); }
  delay(3000);
}
if(tugas00==12)
{ dataFile = SD.open("B00_bc12.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas00);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);

```

```

        dataFile.print('/');
        dataFile.print(now.year(), DEC);
        dataFile.print(" ");
        dataFile.print(now.hour(), DEC);
        dataFile.print(':');
        dataFile.print(now.minute(), DEC);
        dataFile.println();
        dataFile.close(); }
    else
    { Serial.println("error opening B00_bc12.txt"); }
    delay(3000);
}
if(tugas00==13)
{ dataFile = SD.open("B00_bc13.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas00);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B00_bc13.txt"); }
  delay(3000);
}
if(tugas00==14)
{ dataFile = SD.open("B00_bc14.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas00);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B00_bc14.txt"); }
  delay(3000);
}
if(tugas00==15)
{ dataFile = SD.open("B00_bc15.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas00);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');

```

```

        dataFile.print(now.day(), DEC);
        dataFile.print('/');
        dataFile.print(now.month(), DEC);
        dataFile.print('/');
        dataFile.print(now.year(), DEC);
        dataFile.print(" ");
        dataFile.print(now.hour(), DEC);
        dataFile.print(':');
        dataFile.print(now.minute(), DEC);
        dataFile.println();
        dataFile.close(); }
    else
    { Serial.println("error opening B00_bc15.txt"); }
    delay(3000);
}
if(tugas00==16)
{ dataFile = SD.open("B00_bc16.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas00);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B00_bc16.txt"); }
  delay(3000);
}
if(tugas00==17)
{ dataFile = SD.open("B00_bc17.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas00);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B00_bc17.txt"); }
  delay(3000);
}
if(tugas00==18)
{ dataFile = SD.open("B00_bc18.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas00);

```



```

        dataFile.print("|");
        dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
        dataFile.print(' ');
        dataFile.print(now.day(), DEC);
        dataFile.print('/');
        dataFile.print(now.month(), DEC);
        dataFile.print('/');
        dataFile.print(now.year(), DEC);
        dataFile.print(" ");
        dataFile.print(now.hour(), DEC);
        dataFile.print(':');
        dataFile.print(now.minute(), DEC);
        dataFile.println();
        dataFile.close(); }
    else
    { Serial.println("error opening B00_bc18.txt"); }
    delay(3000);
}
if(tugas00==19)
{ dataFile = SD.open("B00_bc19.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas00);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B00_bc19.txt"); }
  delay(3000);
}
if(tugas00==20)
{ dataFile = SD.open("B00_bc20.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas00);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B00_bc20.txt"); }
  delay(3000);
}
if(tugas00==21)

```

```

{ dataFile = SD.open("B00_bc21.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas00);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B00_bc21.txt"); }
  delay(3000);
}
if(tugas00==22)
{ dataFile = SD.open("B00_bc22.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas00);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B00_bc22.txt"); }
  delay(3000);
}
if(tugas00==23)
{ dataFile = SD.open("B00_bc23.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas00);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B00_bc23.txt"); }
}

```

```

    delay(3000);
}
if(tugas00==24)
{
    dataFile = SD.open("B00_bc24.txt", FILE_WRITE);
    if (dataFile)
    { dataFile.print(tugas00);
      dataFile.print("|");
      dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
      dataFile.print(' ');
      dataFile.print(now.day(), DEC);
      dataFile.print('/');
      dataFile.print(now.month(), DEC);
      dataFile.print('/');
      dataFile.print(now.year(), DEC);
      dataFile.print(" ");
      dataFile.print(now.hour(), DEC);
      dataFile.print(':');
      dataFile.print(now.minute(), DEC);
      dataFile.println();
      dataFile.close(); }
    else
    { Serial.println("error opening B00_bc24.txt"); }
    delay(3000);
}
if(tugas00==25)
{ dataFile = SD.open("B00_bc25.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas00);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  {Serial.println("error opening B00_bc25.txt"); }
  delay(3000);
}
if(tugas00==26)
{ dataFile = SD.open("B00_bc26.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas00);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);

```

```

        dataFile.println();
        dataFile.close(); }
    else
    { Serial.println("error opening B00_bc26.txt"); }
    delay(3000);
}
if(tugas00==27)
{ dataFile = SD.open("B00_bc27.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas00);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B00_bc27.txt"); }
  delay(3000);
}
if(tugas00==28)
{ dataFile = SD.open("B00_bc28.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas00);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B00_bc28.txt"); }
  delay(3000);
}
if(tugas00==29)
{ dataFile = SD.open("B00_bc29.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas00);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");

```

```

        dataFile.print(now.hour(), DEC);
        dataFile.print(':');
        dataFile.print(now.minute(), DEC);
        dataFile.println();
        dataFile.close(); }
    else
    { Serial.println("error opening B00_bc29.txt"); }
    delay(3000);
}
if(tugas00==30)
{ dataFile = SD.open("B00_bc30.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas00);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B00_bc30.txt"); }
  delay(3000);
}
if(tugas00>=30) {}
}
}

void B00_bc1()
{ dataFile = SD.open("B00_bc1.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}

void B00_bc2()
{ dataFile = SD.open("B00_bc2.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}

void B00_bc3()
{ dataFile = SD.open("B00_bc3.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}

void B00_bc4()
{ dataFile = SD.open("B00_bc4.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
}

```

```

void B00_bc5()
{  dataFile = SD.open("B00_bc5.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B00_bc6()
{  dataFile = SD.open("B00_bc6.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B00_bc7()
{  dataFile = SD.open("B00_bc7.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B00_bc8()
{  dataFile = SD.open("B00_bc8.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B00_bc9()
{  dataFile = SD.open("B00_bc9.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B00_bc10()
{  dataFile = SD.open("B00_bc10.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B00_bc11()
{  dataFile = SD.open("B00_bc11.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B00_bc12()
{  dataFile = SD.open("B00_bc12.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B00_bc13()
{  dataFile = SD.open("B00_bc13.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}

```

```

void B00_bc14()
{  dataFile = SD.open("B00_bc14.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B00_bc15()
{  dataFile = SD.open("B00_bc15.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B00_bc16()
{  dataFile = SD.open("B00_bc16.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B00_bc17()
{  dataFile = SD.open("B00_bc17.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B00_bc18()
{  dataFile = SD.open("B00_bc18.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B00_bc19()
{  dataFile = SD.open("B00_bc19.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B00_bc20()
{  dataFile = SD.open("B00_bc20.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B00_bc21()
{  dataFile = SD.open("B00_bc21.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B00_bc22()
{  dataFile = SD.open("B00_bc22.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
}

```

```

void B00_bc23()
{  dataFile = SD.open("B00_bc23.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B00_bc24()
{  dataFile = SD.open("B00_bc24.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B00_bc25()
{  dataFile = SD.open("B00_bc25.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B00_bc26()
{  dataFile = SD.open("B00_bc26.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B00_bc27()
{  dataFile = SD.open("B00_bc27.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B00_bc28()
{  dataFile = SD.open("B00_bc28.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B00_bc29()
{  dataFile = SD.open("B00_bc29.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B00_bc30()
{  dataFile = SD.open("B00_bc30.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void  drawDataTugasBox00  ()////////////////////////////////////
MENU 0
{ tft.changeMode(GRAPHIC);
  tft.fillScreen(RA8875_BLACK);
  tft.fillRect(0,10,10,450,RA8875_WHITE); //Bingkai putih left
  tft.fillRect(790,0,10,450,RA8875_WHITE); //Bingkai putih right
  tft.fillRect(0,0,800,30,RA8875_WHITE); //Bingkai putih up
}

```



```
tft.fillRect(0,450,800,10,RA8875_WHITE); //Bingkai putih down
tft.changeMode(TEXT);
tft.setTextColor(RA8875_BLACK);
tft.setFontScale(1);
tft.setCursor(250,0);
tft.print("DATA TUGAS MASUK");
tft.setTextColor(RA8875_WHITE);
tft.setFontScale(0);
tft.setCursor(20,50);
B00_bc1();
tft.setCursor(20,50+(40*1));
B00_bc2();
tft.setCursor(20,50+(40*2));
B00_bc3();
tft.setCursor(20,50+(40*3));
B00_bc4();
tft.setCursor(20,50+(40*4));
B00_bc5();
tft.setCursor(20,50+(40*5));
B00_bc6();
tft.setCursor(20,50+(40*6));
B00_bc7();
tft.setCursor(20,50+(40*7));
B00_bc8();
tft.setCursor(20,50+(40*8));
B00_bc9();
tft.setCursor(20,50+(40*9));
B00_bc10();
tft.setCursor(280,50);
B00_bc11();
tft.setCursor(280,50+(40*1));
B00_bc12();
tft.setCursor(280,50+(40*2));
B00_bc13();
tft.setCursor(280,50+(40*3));
B00_bc14();
tft.setCursor(280,50+(40*4));
B00_bc15();
tft.setCursor(280,50+(40*5));
B00_bc16();
tft.setCursor(280,50+(40*6));
B00_bc17();
tft.setCursor(280,50+(40*7));
B00_bc18();
tft.setCursor(280,50+(40*8));
B00_bc19();
tft.setCursor(280,50+(40*9));
B00_bc20();
tft.setCursor(540,50);
B00_bc21();
tft.setCursor(540,50+(40*1));
B00_bc22();
tft.setCursor(540,50+(40*2));
B00_bc23();
tft.setCursor(540,50+(40*3));
B00_bc24();
tft.setCursor(540,50+(40*4));
B00_bc25();
tft.setCursor(540,50+(40*5));
B00_bc26();
tft.setCursor(540,50+(40*6));
B00_bc27();
tft.setCursor(540,50+(40*7));
```

```

B00_bc28();
tft.setCursor(540,50+(40*8));
B00_bc29();
tft.setCursor(540,50+(40*9));
B00_bc30();
tft.changeMode(GRAPHIC);
tft.fillRect(2,2,60,25,RA8875_BLACK);
tft.changeMode(TEXT);
tft.setTextColor(RA8875_WHITE);
tft.setFontScale(0);
tft.setCursor(7,5);
tft.print("LOGOUT");
}
void deteksi_tugas01()
{ if (digitalRead(sensor_Box00)==LOW)
  { DateTime now = rtc.now();
    tugas01++;
    Serial.println("Deteksi 00");
    if(tugas01==1)
    { dataFile = SD.open("B01_bc1.txt", FILE_WRITE);
      if (dataFile)
      { Serial.println("opening B01_bc.txt");
        dataFile.print(tugas01);
        dataFile.print("|");
        dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
        dataFile.print(' ');
        dataFile.print(now.day(), DEC);
        dataFile.print('/');
        dataFile.print(now.month(), DEC);
        dataFile.print('/');
        dataFile.print(now.year(), DEC);
        dataFile.print(" ");
        dataFile.print(now.hour(), DEC);
        dataFile.print(':');
        dataFile.print(now.minute(), DEC);
        dataFile.println();
        dataFile.close(); }
      else
      {Serial.println("error opening B01_bc.txt"); }
      delay(3000);
    }
    if(tugas01==2)
    { dataFile = SD.open("B01_bc2.txt", FILE_WRITE);
      if (dataFile)
      { dataFile.print(tugas01);
        dataFile.print("|");
        dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
        dataFile.print(' ');
        dataFile.print(now.day(), DEC);
        dataFile.print('/');
        dataFile.print(now.month(), DEC);
        dataFile.print('/');
        dataFile.print(now.year(), DEC);
        dataFile.print(" ");
        dataFile.print(now.hour(), DEC);
        dataFile.print(':');
        dataFile.print(now.minute(), DEC);
        dataFile.println();
        dataFile.close(); }
      else
      { Serial.println("error opening B01_bc2.txt"); }
      delay(3000);
    }
  }
}

```

```

if(tugas01==3)
{ dataFile = SD.open("B01_bc3.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close();      }
  else
  { Serial.println("error opening B01_bc3.txt"); }
  delay(3000);
}
if(tugas01==4)
{ dataFile = SD.open("B01_bc4.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close();  }
  else
  { Serial.println("error opening B01_bc4.txt"); }
  delay(3000);
}
if(tugas01==5)
{ dataFile = SD.open("B01_bc5.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else

```

```

    { Serial.println("error opening B01_bc5.txt"); }
    delay(3000);
}
if(tugas01==6)
{ dataFile = SD.open("B01_bc6.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B01_bc6.txt"); }
  delay(3000);
}
if(tugas01==7)
{ dataFile = SD.open("B01_bc7.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B01_bc7.txt"); }
  delay(3000);
}
if(tugas01==8)
{ dataFile = SD.open("B01_bc8.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);

```

```

        dataFile.println();
        dataFile.close(); }
    else
    { Serial.println("error opening B01_bc8.txt"); }
    delay(3000);
}
if(tugas01==9)
{ dataFile = SD.open("B01_bc9.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B01_bc9.txt"); }
  delay(3000);
}
if(tugas01==10)
{ dataFile = SD.open("B01_bc10.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B01_bc0.txt"); }
  delay(3000);
}
if(tugas01==11)
{ dataFile = SD.open("B01_bc11.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");

```

```

        dataFile.print(now.hour(), DEC);
        dataFile.print(':');
        dataFile.print(now.minute(), DEC);
        dataFile.println();
        dataFile.close(); }
    else
    { Serial.println("error opening B01_bc11.txt"); }
    delay(3000);
}
if(tugas01==12)
{ dataFile = SD.open("B01_bc12.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B01_bc12.txt"); }
  delay(3000);
}
if(tugas01==13)
{ dataFile = SD.open("B01_bc13.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B01_bc13.txt"); }
  delay(3000);
}
if(tugas01==14)
{ dataFile = SD.open("B01_bc14.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);

```

```

        dataFile.print('/');
        dataFile.print(now.year(), DEC);
        dataFile.print(" ");
        dataFile.print(now.hour(), DEC);
        dataFile.print(':');
        dataFile.print(now.minute(), DEC);
        dataFile.println();
        dataFile.close(); }
    else
    { Serial.println("error opening B01_bc14.txt"); }
    delay(3000);
}
if(tugas01==15)
{ dataFile = SD.open("B01_bc15.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B01_bc15.txt"); }
  delay(3000);
}
if(tugas01==16)
{ dataFile = SD.open("B01_bc16.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B01_bc16.txt"); }
  delay(3000);
}
if(tugas01==17)
{ dataFile = SD.open("B01_bc17.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');

```

```

        dataFile.print(now.day(), DEC);
        dataFile.print('/');
        dataFile.print(now.month(), DEC);
        dataFile.print('/');
        dataFile.print(now.year(), DEC);
        dataFile.print(" ");
        dataFile.print(now.hour(), DEC);
        dataFile.print(':');
        dataFile.print(now.minute(), DEC);
        dataFile.println();
        dataFile.close(); }
    else
    { Serial.println("error opening B01_bc17.txt"); }
    delay(3000);
}
if(tugas01==18)
{ dataFile = SD.open("B01_bc18.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B01_bc18.txt"); }
  delay(3000);
}
if(tugas01==19)
{ dataFile = SD.open("B01_bc19.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B01_bc19.txt"); }
  delay(3000);
}
if(tugas01==20)
{ dataFile = SD.open("B01_bc20.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);

```



```

        dataFile.print("|");
        dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
        dataFile.print(' ');
        dataFile.print(now.day(), DEC);
        dataFile.print('/');
        dataFile.print(now.month(), DEC);
        dataFile.print('/');
        dataFile.print(now.year(), DEC);
        dataFile.print(" ");
        dataFile.print(now.hour(), DEC);
        dataFile.print(':');
        dataFile.print(now.minute(), DEC);
        dataFile.println();
        dataFile.close(); }
    else
    { Serial.println("error opening B01_bc20.txt"); }
    delay(3000);
}
if(tugas01==21)
{ dataFile = SD.open("B01_bc21.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B01_bc21.txt"); }
  delay(3000);
}
if(tugas01==22)
{ dataFile = SD.open("B01_bc22.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B01_bc22.txt"); }
  delay(3000);
}
if(tugas01==23)

```

```

{ dataFile = SD.open("B01_bc23.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B01_bc23.txt"); }
  delay(3000);
}
if(tugas01==24)
{
  dataFile = SD.open("B01_bc24.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B01_bc24.txt"); }
  delay(3000);
}
if(tugas01==25)
{ dataFile = SD.open("B01_bc25.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else

```

```

        {Serial.println("error opening B01_bc25.txt"); }
        delay(3000);
    }
    if(tugas01==26)
    { dataFile = SD.open("B01_bc26.txt", FILE_WRITE);
      if (dataFile)
      { dataFile.print(tugas01);
        dataFile.print("|");
        dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
        dataFile.print(' ');
        dataFile.print(now.day(), DEC);
        dataFile.print('/');
        dataFile.print(now.month(), DEC);
        dataFile.print('/');
        dataFile.print(now.year(), DEC);
        dataFile.print(" ");
        dataFile.print(now.hour(), DEC);
        dataFile.print(':');
        dataFile.print(now.minute(), DEC);
        dataFile.println();
        dataFile.close(); }
      else
      { Serial.println("error opening B01_bc26.txt"); }
      delay(3000);
    }
    if(tugas01==27)
    { dataFile = SD.open("B01_bc27.txt", FILE_WRITE);
      if (dataFile)
      { dataFile.print(tugas01);
        dataFile.print("|");
        dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
        dataFile.print(' ');
        dataFile.print(now.day(), DEC);
        dataFile.print('/');
        dataFile.print(now.month(), DEC);
        dataFile.print('/');
        dataFile.print(now.year(), DEC);
        dataFile.print(" ");
        dataFile.print(now.hour(), DEC);
        dataFile.print(':');
        dataFile.print(now.minute(), DEC);
        dataFile.println();
        dataFile.close(); }
      else
      { Serial.println("error opening B01_bc27.txt"); }
      delay(3000);
    }
    if(tugas01==28)
    { dataFile = SD.open("B01_bc28.txt", FILE_WRITE);
      if (dataFile)
      { dataFile.print(tugas01);
        dataFile.print("|");
        dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
        dataFile.print(' ');
        dataFile.print(now.day(), DEC);
        dataFile.print('/');
        dataFile.print(now.month(), DEC);
        dataFile.print('/');
        dataFile.print(now.year(), DEC);
        dataFile.print(" ");
        dataFile.print(now.hour(), DEC);
        dataFile.print(':');
        dataFile.print(now.minute(), DEC);

```

```

        dataFile.println();
        dataFile.close(); }
    else
    { Serial.println("error opening B01_bc28.txt"); }
    delay(3000);
}
if(tugas01==29)
{ dataFile = SD.open("B01_bc29.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B01_bc29.txt"); }
  delay(3000);
}
if(tugas01==30)
{ dataFile = SD.open("B01_bc30.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B01_bc30.txt"); }
  delay(3000);
}
}
if(tugas01>=30) {}
}
}

void B01_bc1()
{ dataFile = SD.open("B01_bc1.txt");
  if (dataFile){
    while (dataFile.available())
    {tft.write(dataFile.read());}
    dataFile.close();}
}
void B01_bc2()
{ dataFile = SD.open("B01_bc2.txt");

```

```

    if (dataFile){
        while (dataFile.available())
            {tft.write(dataFile.read());}
        dataFile.close();}
}
void B01_bc3()
{ dataFile = SD.open("B01_bc3.txt");
  if (dataFile){
    while (dataFile.available())
        {tft.write(dataFile.read());}
    dataFile.close();}
}
void B01_bc4()
{ dataFile = SD.open("B01_bc4.txt");
  if (dataFile){
    while (dataFile.available())
        {tft.write(dataFile.read());}
    dataFile.close();}
}
void B01_bc5()
{ dataFile = SD.open("B01_bc5.txt");
  if (dataFile){
    while (dataFile.available())
        {tft.write(dataFile.read());}
    dataFile.close();}
}
void B01_bc6()
{ dataFile = SD.open("B01_bc6.txt");
  if (dataFile){
    while (dataFile.available())
        {tft.write(dataFile.read());}
    dataFile.close();}
}
void B01_bc7()
{ dataFile = SD.open("B01_bc7.txt");
  if (dataFile){
    while (dataFile.available())
        {tft.write(dataFile.read());}
    dataFile.close();}
}
void B01_bc8()
{ dataFile = SD.open("B01_bc8.txt");
  if (dataFile){
    while (dataFile.available())
        {tft.write(dataFile.read());}
    dataFile.close();}
}
void B01_bc9()
{ dataFile = SD.open("B01_bc9.txt");
  if (dataFile){
    while (dataFile.available())
        {tft.write(dataFile.read());}
    dataFile.close();}
}
void B01_bc10()
{ dataFile = SD.open("B01_bc10.txt");
  if (dataFile){
    while (dataFile.available())
        {tft.write(dataFile.read());}
    dataFile.close();}
}
void B01_bc11()
{ dataFile = SD.open("B01_bc11.txt");

```

```

    if (dataFile){
        while (dataFile.available())
            {tft.write(dataFile.read());}
        dataFile.close();}
}
void B01_bc12()
{ dataFile = SD.open("B01_bc12.txt");
  if (dataFile){
    while (dataFile.available())
        {tft.write(dataFile.read());}
    dataFile.close();}
}
void B01_bc13()
{ dataFile = SD.open("B01_bc13.txt");
  if (dataFile){
    while (dataFile.available())
        {tft.write(dataFile.read());}
    dataFile.close();}
}
void B01_bc14()
{ dataFile = SD.open("B01_bc14.txt");
  if (dataFile){
    while (dataFile.available())
        {tft.write(dataFile.read());}
    dataFile.close();}
}
void B01_bc15()
{ dataFile = SD.open("B01_bc15.txt");
  if (dataFile){
    while (dataFile.available())
        {tft.write(dataFile.read());}
    dataFile.close();}
}
void B01_bc16()
{ dataFile = SD.open("B01_bc16.txt");
  if (dataFile){
    while (dataFile.available())
        {tft.write(dataFile.read());}
    dataFile.close();}
}
void B01_bc17()
{ dataFile = SD.open("B01_bc17.txt");
  if (dataFile){
    while (dataFile.available())
        {tft.write(dataFile.read());}
    dataFile.close();}
}
void B01_bc18()
{ dataFile = SD.open("B01_bc18.txt");
  if (dataFile){
    while (dataFile.available())
        {tft.write(dataFile.read());}
    dataFile.close();}
}
void B01_bc19()
{ dataFile = SD.open("B01_bc19.txt");
  if (dataFile){
    while (dataFile.available())
        {tft.write(dataFile.read());}
    dataFile.close();}
}
void B01_bc20()
{ dataFile = SD.open("B01_bc20.txt");

```

```

    if (dataFile){
        while (dataFile.available())
            {tft.write(dataFile.read());}
        dataFile.close();}
}
void B01_bc21()
{ dataFile = SD.open("B01_bc21.txt");
  if (dataFile){
    while (dataFile.available())
        {tft.write(dataFile.read());}
    dataFile.close();}
}
void B01_bc22()
{ dataFile = SD.open("B01_bc22.txt");
  if (dataFile){
    while (dataFile.available())
        {tft.write(dataFile.read());}
    dataFile.close();}
}
void B01_bc23()
{ dataFile = SD.open("B01_bc23.txt");
  if (dataFile){
    while (dataFile.available())
        {tft.write(dataFile.read());}
    dataFile.close();}
}
void B01_bc24()
{ dataFile = SD.open("B01_bc24.txt");
  if (dataFile){
    while (dataFile.available())
        {tft.write(dataFile.read());}
    dataFile.close();}
}
void B01_bc25()
{ dataFile = SD.open("B01_bc25.txt");
  if (dataFile){
    while (dataFile.available())
        {tft.write(dataFile.read());}
    dataFile.close();}
}
void B01_bc26()
{ dataFile = SD.open("B01_bc26.txt");
  if (dataFile){
    while (dataFile.available())
        {tft.write(dataFile.read());}
    dataFile.close();}
}
void B01_bc27()
{ dataFile = SD.open("B01_bc27.txt");
  if (dataFile){
    while (dataFile.available())
        {tft.write(dataFile.read());}
    dataFile.close();}
}
void B01_bc28()
{ dataFile = SD.open("B01_bc28.txt");
  if (dataFile){
    while (dataFile.available())
        {tft.write(dataFile.read());}
    dataFile.close();}
}
void B01_bc29()
{ dataFile = SD.open("B01_bc29.txt");

```

```

    if (dataFile){
        while (dataFile.available())
            {tft.write(dataFile.read());}
        dataFile.close();}
}
void B01_bc30()
{ dataFile = SD.open("B01_bc30.txt");
  if (dataFile){
    while (dataFile.available())
        {tft.write(dataFile.read());}
    dataFile.close();}
}
void drawDataTugasBox00 ()//////////////////////////////////////
MENU 0
{ tft.changeMode(GRAPHIC);
  tft.fillScreen(RA8875_BLACK);
  tft.fillRect(0,10,10,450,RA8875_WHITE); //Bingkai putih left
  tft.fillRect(790,0,10,450,RA8875_WHITE); //Bingkai putih right
  tft.fillRect(0,0,800,30,RA8875_WHITE); //Bingkai putih up
  tft.fillRect(0,450,800,10,RA8875_WHITE); //Bingkai putih down
  tft.changeMode(TEXT);
  tft.setTextColor(RA8875_BLACK);
  tft.setFontScale(1);
  tft.setCursor(250,0);
  tft.print("DATA TUGAS MASUK");
  tft.setTextColor(RA8875_WHITE);
  tft.setFontScale(0);
  tft.setCursor(20,50);
  B01_bc1();
  tft.setCursor(20,50+(40*1));
  B01_bc2();
  tft.setCursor(20,50+(40*2));
  B01_bc3();
  tft.setCursor(20,50+(40*3));
  B01_bc4();
  tft.setCursor(20,50+(40*4));
  B01_bc5();
  tft.setCursor(20,50+(40*5));
  B01_bc6();
  tft.setCursor(20,50+(40*6));
  B01_bc7();
  tft.setCursor(20,50+(40*7));
  B01_bc8();
  tft.setCursor(20,50+(40*8));
  B01_bc9();
  tft.setCursor(20,50+(40*9));
  B01_bc10();
  tft.setCursor(280,50);
  B01_bc11();
  tft.setCursor(280,50+(40*1));
  B01_bc12();
  tft.setCursor(280,50+(40*2));
  B01_bc13();
  tft.setCursor(280,50+(40*3));
  B01_bc14();
  tft.setCursor(280,50+(40*4));
  B01_bc15();
  tft.setCursor(280,50+(40*5));
  B01_bc16();
  tft.setCursor(280,50+(40*6));
  B01_bc17();
  tft.setCursor(280,50+(40*7));
  B01_bc18();
}

```



```

tft.setCursor(280,50+(40*8));
B01_bc19();
tft.setCursor(280,50+(40*9));
B01_bc20();
tft.setCursor(540,50);
B01_bc21();
tft.setCursor(540,50+(40*1));
B01_bc22();
tft.setCursor(540,50+(40*2));
B01_bc23();
tft.setCursor(540,50+(40*3));
B01_bc24();
tft.setCursor(540,50+(40*4));
B01_bc25();
tft.setCursor(540,50+(40*5));
B01_bc26();
tft.setCursor(540,50+(40*6));
B01_bc27();
tft.setCursor(540,50+(40*7));
B01_bc28();
tft.setCursor(540,50+(40*8));
B01_bc29();
tft.setCursor(540,50+(40*9));
B01_bc30();
tft.changeMode(GRAPHIC);
tft.fillRect(2,2,60,25,RA8875_BLACK);
tft.changeMode(TEXT);
tft.setTextColor(RA8875_WHITE);
tft.setFontScale(0);
tft.setCursor(7,5);
tft.print("LOGOUT");
}
void deteksi_tugas01()
{ if (digitalRead(sensor_Box00)==LOW)
  { DateTime now = rtc.now();
    tugas01++;
    Serial.println("Deteksi 00");
    if(tugas01==1)
    { dataFile = SD.open("B02_bc1.txt", FILE_WRITE);
      if (dataFile)
      { Serial.println("opening B02_bc.txt");
        dataFile.print(tugas01);
        dataFile.print("|");
        dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
        dataFile.print(' ');
        dataFile.print(now.day(), DEC);
        dataFile.print('/');
        dataFile.print(now.month(), DEC);
        dataFile.print('/');
        dataFile.print(now.year(), DEC);
        dataFile.print(" ");
        dataFile.print(now.hour(), DEC);
        dataFile.print(':');
        dataFile.print(now.minute(), DEC);
        dataFile.println();
        dataFile.close(); }
      else
      {Serial.println("error opening B02_bc.txt"); }
      delay(3000);
    }
  }
  if(tugas01==2)
  { dataFile = SD.open("B02_bc2.txt", FILE_WRITE);
    if (dataFile)

```

```

    { dataFile.print(tugas01);
      dataFile.print("|");
      dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
      dataFile.print(' ');
      dataFile.print(now.day(), DEC);
      dataFile.print('/');
      dataFile.print(now.month(), DEC);
      dataFile.print('/');
      dataFile.print(now.year(), DEC);
      dataFile.print(" ");
      dataFile.print(now.hour(), DEC);
      dataFile.print(':');
      dataFile.print(now.minute(), DEC);
      dataFile.println();
      dataFile.close(); }
    else
    { Serial.println("error opening B02_bc2.txt"); }
    delay(3000);
  }
}
if(tugas01==3)
{ dataFile = SD.open("B02_bc3.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close();      }
  else
  { Serial.println("error opening B02_bc3.txt"); }
  delay(3000);
}
if(tugas01==4)
{ dataFile = SD.open("B02_bc4.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close();      }
  else
  { Serial.println("error opening B02_bc4.txt"); }
  delay(3000);
}
}

```

```

if(tugas01==5)
  { dataFile = SD.open("B02_bc5.txt", FILE_WRITE);
    if (dataFile)
      { dataFile.print(tugas01);
        dataFile.print("|");
        dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
        dataFile.print(' ');
        dataFile.print(now.day(), DEC);
        dataFile.print('/');
        dataFile.print(now.month(), DEC);
        dataFile.print('/');
        dataFile.print(now.year(), DEC);
        dataFile.print(" ");
        dataFile.print(now.hour(), DEC);
        dataFile.print(':');
        dataFile.print(now.minute(), DEC);
        dataFile.println();
        dataFile.close(); }
      else
        { Serial.println("error opening B02_bc5.txt"); }
        delay(3000);
    }
}
if(tugas01==6)
  { dataFile = SD.open("B02_bc6.txt", FILE_WRITE);
    if (dataFile)
      { dataFile.print(tugas01);
        dataFile.print("|");
        dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
        dataFile.print(' ');
        dataFile.print(now.day(), DEC);
        dataFile.print('/');
        dataFile.print(now.month(), DEC);
        dataFile.print('/');
        dataFile.print(now.year(), DEC);
        dataFile.print(" ");
        dataFile.print(now.hour(), DEC);
        dataFile.print(':');
        dataFile.print(now.minute(), DEC);
        dataFile.println();
        dataFile.close(); }
      else
        { Serial.println("error opening B02_bc6.txt"); }
        delay(3000);
    }
}
if(tugas01==7)
  { dataFile = SD.open("B02_bc7.txt", FILE_WRITE);
    if (dataFile)
      { dataFile.print(tugas01);
        dataFile.print("|");
        dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
        dataFile.print(' ');
        dataFile.print(now.day(), DEC);
        dataFile.print('/');
        dataFile.print(now.month(), DEC);
        dataFile.print('/');
        dataFile.print(now.year(), DEC);
        dataFile.print(" ");
        dataFile.print(now.hour(), DEC);
        dataFile.print(':');
        dataFile.print(now.minute(), DEC);
        dataFile.println();
        dataFile.close(); }
      else

```

```

    { Serial.println("error opening B02_bc7.txt"); }
    delay(3000);
}
if(tugas01==8)
{ dataFile = SD.open("B02_bc8.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B02_bc8.txt"); }
  delay(3000);
}
if(tugas01==9)
{ dataFile = SD.open("B02_bc9.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B02_bc9.txt"); }
  delay(3000);
}
if(tugas01==10)
{ dataFile = SD.open("B02_bc10.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);

```

```

        dataFile.println();
        dataFile.close(); }
    else
    { Serial.println("error opening B02_bc0.txt"); }
    delay(3000);
}
if(tugas01==11)
{ dataFile = SD.open("B02_bc11.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B02_bc11.txt"); }
  delay(3000);
}
if(tugas01==12)
{ dataFile = SD.open("B02_bc12.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B02_bc12.txt"); }
  delay(3000);
}
if(tugas01==13)
{ dataFile = SD.open("B02_bc13.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");

```

```

        dataFile.print(now.hour(), DEC);
        dataFile.print(':');
        dataFile.print(now.minute(), DEC);
        dataFile.println();
        dataFile.close(); }
    else
    { Serial.println("error opening B02_bc13.txt"); }
    delay(3000);
}
if(tugas01==14)
{ dataFile = SD.open("B02_bc14.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B02_bc14.txt"); }
  delay(3000);
}
if(tugas01==15)
{ dataFile = SD.open("B02_bc15.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B02_bc15.txt"); }
  delay(3000);
}
if(tugas01==16)
{ dataFile = SD.open("B02_bc16.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);

```

```

        dataFile.print('/');
        dataFile.print(now.year(), DEC);
        dataFile.print(" ");
        dataFile.print(now.hour(), DEC);
        dataFile.print(':');
        dataFile.print(now.minute(), DEC);
        dataFile.println();
        dataFile.close(); }
    else
    { Serial.println("error opening B02_bc16.txt"); }
    delay(3000);
}
if(tugas01==17)
{ dataFile = SD.open("B02_bc17.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B02_bc17.txt"); }
  delay(3000);
}
if(tugas01==18)
{ dataFile = SD.open("B02_bc18.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B02_bc18.txt"); }
  delay(3000);
}
if(tugas01==19)
{ dataFile = SD.open("B02_bc19.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');

```

```

    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
else
{ Serial.println("error opening B02_bc19.txt"); }
delay(3000);
}
if(tugas01==20)
{ dataFile = SD.open("B02_bc20.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B02_bc20.txt"); }
  delay(3000);
}
if(tugas01==21)
{ dataFile = SD.open("B02_bc21.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B02_bc21.txt"); }
  delay(3000);
}
if(tugas01==22)
{ dataFile = SD.open("B02_bc22.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);

```



```

        dataFile.print("|");
        dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
        dataFile.print(' ');
        dataFile.print(now.day(), DEC);
        dataFile.print('/');
        dataFile.print(now.month(), DEC);
        dataFile.print('/');
        dataFile.print(now.year(), DEC);
        dataFile.print(" ");
        dataFile.print(now.hour(), DEC);
        dataFile.print(':');
        dataFile.print(now.minute(), DEC);
        dataFile.println();
        dataFile.close(); }
    else
    { Serial.println("error opening B02_bc22.txt"); }
    delay(3000);
}
if(tugas01==23)
{ dataFile = SD.open("B02_bc23.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B02_bc23.txt"); }
  delay(3000);
}
if(tugas01==24)
{
  dataFile = SD.open("B02_bc24.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B02_bc24.txt"); }
  delay(3000);
}
}

```

```

if(tugas01==25)
{ dataFile = SD.open("B02_bc25.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  {Serial.println("error opening B02_bc25.txt"); }
  delay(3000);
}
if(tugas01==26)
{ dataFile = SD.open("B02_bc26.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B02_bc26.txt"); }
  delay(3000);
}
if(tugas01==27)
{ dataFile = SD.open("B02_bc27.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else

```

```

    { Serial.println("error opening B02_bc27.txt"); }
    delay(3000);
}
if(tugas01==28)
{ dataFile = SD.open("B02_bc28.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B02_bc28.txt"); }
  delay(3000);
}
if(tugas01==29)
{ dataFile = SD.open("B02_bc29.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);
    dataFile.println();
    dataFile.close(); }
  else
  { Serial.println("error opening B02_bc29.txt"); }
  delay(3000);
}
if(tugas01==30)
{ dataFile = SD.open("B02_bc30.txt", FILE_WRITE);
  if (dataFile)
  { dataFile.print(tugas01);
    dataFile.print("|");
    dataFile.print(daysOfTheWeek[now.dayOfTheWeek()]);
    dataFile.print(' ');
    dataFile.print(now.day(), DEC);
    dataFile.print('/');
    dataFile.print(now.month(), DEC);
    dataFile.print('/');
    dataFile.print(now.year(), DEC);
    dataFile.print(" ");
    dataFile.print(now.hour(), DEC);
    dataFile.print(':');
    dataFile.print(now.minute(), DEC);

```

```

        dataFile.println();
        dataFile.close(); }
    else
    { Serial.println("error opening B02_bc30.txt"); }
    delay(3000);
}
if(tugas01>=30) {}
}
}

void B02_bc1()
{ dataFile = SD.open("B02_bc1.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}

void B02_bc2()
{ dataFile = SD.open("B02_bc2.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}

void B02_bc3()
{ dataFile = SD.open("B02_bc3.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}

void B02_bc4()
{ dataFile = SD.open("B02_bc4.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}

void B02_bc5()
{ dataFile = SD.open("B02_bc5.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}

void B02_bc6()
{ dataFile = SD.open("B02_bc6.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}

void B02_bc7()
{ dataFile = SD.open("B02_bc7.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}

void B02_bc8()
{ dataFile = SD.open("B02_bc8.txt");
  if (dataFile){
    while (dataFile.available())

```

```

        {tft.write(dataFile.read());}
        dataFile.close();}
    }
void B02_bc9()
{  dataFile = SD.open("B02_bc9.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B02_bc10()
{  dataFile = SD.open("B02_bc10.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B02_bc11()
{  dataFile = SD.open("B02_bc11.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B02_bc12()
{  dataFile = SD.open("B02_bc12.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B02_bc13()
{  dataFile = SD.open("B02_bc13.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B02_bc14()
{  dataFile = SD.open("B02_bc14.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B02_bc15()
{  dataFile = SD.open("B02_bc15.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B02_bc16()
{  dataFile = SD.open("B02_bc16.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B02_bc17()
{  dataFile = SD.open("B02_bc17.txt");
  if (dataFile){
    while (dataFile.available())

```

```

        {tft.write(dataFile.read());}
        dataFile.close();}
}
void B02_bc18()
{ dataFile = SD.open("B02_bc18.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B02_bc19()
{ dataFile = SD.open("B02_bc19.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B02_bc20()
{ dataFile = SD.open("B02_bc20.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B02_bc21()
{ dataFile = SD.open("B02_bc21.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B02_bc22()
{ dataFile = SD.open("B02_bc22.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B02_bc23()
{ dataFile = SD.open("B02_bc23.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B02_bc24()
{ dataFile = SD.open("B02_bc24.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B02_bc25()
{ dataFile = SD.open("B02_bc25.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B02_bc26()
{ dataFile = SD.open("B02_bc26.txt");
  if (dataFile){
    while (dataFile.available())

```

```

        {tft.write(dataFile.read());}
        dataFile.close();}
    }
void B02_bc27()
{  dataFile = SD.open("B02_bc27.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B02_bc28()
{  dataFile = SD.open("B02_bc28.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B02_bc29()
{  dataFile = SD.open("B02_bc29.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
void B02_bc30()
{  dataFile = SD.open("B02_bc30.txt");
  if (dataFile){
    while (dataFile.available())
      {tft.write(dataFile.read());}
    dataFile.close();}
}
}
void drawDataTugasBox00  ()//////////////////////////////////////
MENU 0
{ tft.changeMode (GRAPHIC);
  tft.fillScreen(RA8875_BLACK);
  tft.fillRect(0,10,10,450,RA8875_WHITE); //Bingkai putih left
  tft.fillRect(790,0,10,450,RA8875_WHITE); //Bingkai putih right
  tft.fillRect(0,0,800,30,RA8875_WHITE); //Bingkai putih up
  tft.fillRect(0,450,800,10,RA8875_WHITE); //Bingkai putih down
  tft.changeMode (TEXT);
  tft.setTextColor(RA8875_BLACK);
  tft.setFontScale(1);
  tft.setCursor(250,0);
  tft.print("DATA TUGAS MASUK");
  tft.setTextColor(RA8875_WHITE);
  tft.setFontScale(0);
  tft.setCursor(20,50);
  B02_bc1();
  tft.setCursor(20,50+(40*1));
  B02_bc2();
  tft.setCursor(20,50+(40*2));
  B02_bc3();
  tft.setCursor(20,50+(40*3));
  B02_bc4();
  tft.setCursor(20,50+(40*4));
  B02_bc5();
  tft.setCursor(20,50+(40*5));
  B02_bc6();
  tft.setCursor(20,50+(40*6));
  B02_bc7();
  tft.setCursor(20,50+(40*7));
  B02_bc8();
  tft.setCursor(20,50+(40*8));
}

```

```
B02_bc9();
tft.setCursor(20,50+(40*9));
B02_bc10();
tft.setCursor(280,50);
B02_bc11();
tft.setCursor(280,50+(40*1));
B02_bc12();
tft.setCursor(280,50+(40*2));
B02_bc13();
tft.setCursor(280,50+(40*3));
B02_bc14();
tft.setCursor(280,50+(40*4));
B02_bc15();
tft.setCursor(280,50+(40*5));
B02_bc16();
tft.setCursor(280,50+(40*6));
B02_bc17();
tft.setCursor(280,50+(40*7));
B02_bc18();
tft.setCursor(280,50+(40*8));
B02_bc19();
tft.setCursor(280,50+(40*9));
B02_bc20();
tft.setCursor(540,50);
B02_bc21();
tft.setCursor(540,50+(40*1));
B02_bc22();
tft.setCursor(540,50+(40*2));
B02_bc23();
tft.setCursor(540,50+(40*3));
B02_bc24();
tft.setCursor(540,50+(40*4));
B02_bc25();
tft.setCursor(540,50+(40*5));
B02_bc26();
tft.setCursor(540,50+(40*6));
B02_bc27();
tft.setCursor(540,50+(40*7));
B02_bc28();
tft.setCursor(540,50+(40*8));
B02_bc29();
tft.setCursor(540,50+(40*9));
B02_bc30();
tft.changeMode(GRAPHIC);
tft.fillRect(2,2,60,25,RA8875_BLACK);
}
```