**LAPORAN PROJECT SISTEM EMBEDDED**

**APLIKASI TIMESTATE MACHINE PADA VENDING MACHINE**



***Disusun Oleh :***

**Tondi Taufiq Akbar Nasution 1110171048**

**3 D4 ELEKTRONIKA B**

**PROGRAM STUDI TEKNIK ELEKTRONIKA**

**DEPARTEMEN ELEKTRO**

**POLITEKNIK ELEKTRONIKA NEGERI SURABAYA**

**2019**

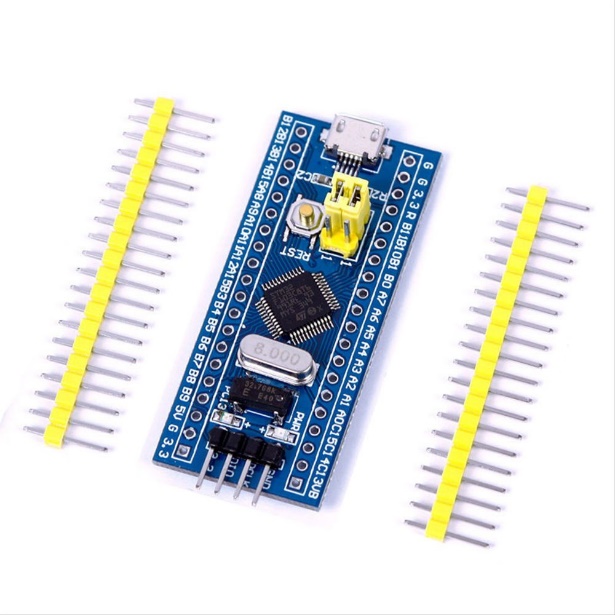
**LAPORAN PROJECT SISTEM EMBEDDED**

**APLIKASI TIMESTATE MACHINE PADA VENDING MACHINE**

1. **TUJUAN**
2. Mahasiswa dapat membuat program timestate machine dengan menggunakan ARM
3. Mahasiswa dapat mengimplementasikan program timestate machine menggunakan pemrograman berstruktur dan mekanisme time out dalam suatu alat.
4. **DASAR TEORI**

II.1 ARM (STM32F103C8T6)

Arsitektur ARM merupakan arsitektur [prosesor](https://id.wikipedia.org/wiki/Prosesor) 32-bit RISC yang dikembangkan oleh *ARM Limited*. Dikenal sebagai *Advanced RISC Machine* di mana sebelumnya dikenal sebagai *Acorn RISC Machine*. Pada awalnya merupakan *prosesor desktop* yang sekarang didominasi oleh keluarga x86. Namun desain yang sederhana membuat prosesor ARM cocok untuk aplikasi berdaya rendah. Hal ini membuat prosesor ARM mendominasi pasar *mobile electronic* dan *embedded system* di mana membutuhkan daya dan harga yang rendah.

 STM32 adalah  mikrokontroler  berbasis inti prosesor 32 bit RISC ARM Cortex-M7, Cortex-M4F, Cortex-M3, Cortex-M0+, dan Cortex-M0  dari STMicrelectronics. Mikrokontroler ini mempunyai frekuensi clock tinggi, umumnya berada pada kisaran 72MHz atau lebih.

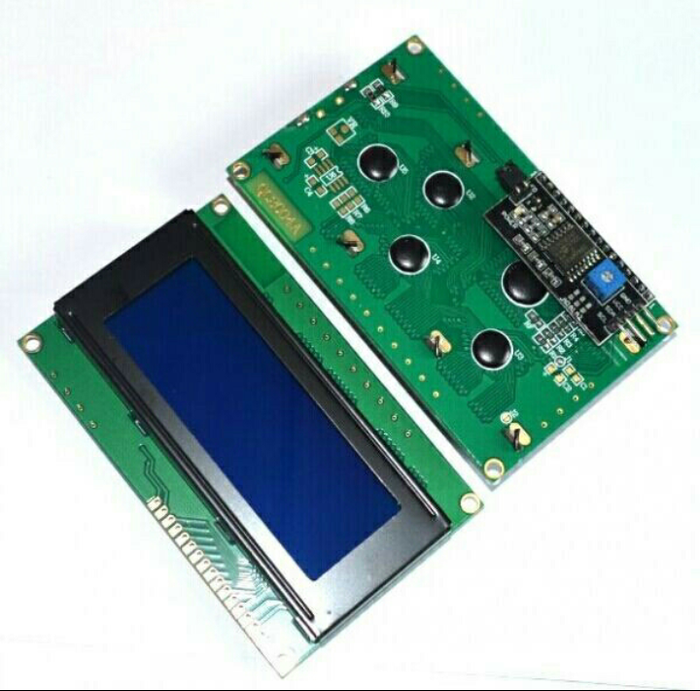
*Gambar II.1 STM32F103C8T6*

II.2 I2C LCD 20X4

LCD atau Liquid Crystal Display adalah suatu jenis media display (tampilaan) yang dibuat menggunakan teknologi *CMOS logic*yang bekerja dengan tidak menghasilkan cahaya namun memantulkan cahaya yang ada disekeliingnya terhadap *front-lit* atau mentransmisikan cahaya dari *back-lit*. LCD berfungsi sebagai penampil data baik dalam bentuk huruf, angka, karakter ataupun grafik.

Yang dimaksud dengan I2C LCD adalah modul LCD yang dikendalikan secara serial sinkron dengan protocol I2C/IIC (*Inter Integrated Circuit)* atau TWI (*Two Wire Interface).* Normalnya, modul LCD dikendalikan secara parallel baik untuk jalur data maupun kontrolnya. Namun, jalur parallel akan memakan banyak pin disisi controller (misal Arduino, Android, Komputer, dll). Setidaknya kita akan membutuhkan 6 atau 7 pin untuk mengendalikan sebuah modul LCD. Dengan demikian untuk sebuah controller yang “sibuk” dan harus, mengendalikan banyak I/O, menggunakan jalur parallel adalah solusi yang kurang tepat.

Arduino sudah mendukung protocol I2C/IIC. Di papan Arduino Uno, port I2C terletak pada pin A4 untuk jalur SDA (Serial Data) dan pin A5 untuk jalur SCL (Serial Clock), jangan lupa untuk menghubungkan jalur kabel ground antara Arduino dengan perangkat I2C client. Untuk sisi software, Arduino sudah cukup membantu kita bekerja dengan protocol ini melalui library ‘Wire.h’. Berikutnya, library ini akan dimanfaatkan untuk mengkonversi jalur parallel LCD menjadi jalur serial I2C. Kita dapat secara mudah melakukannya, namun agar lebih mudah kita dapat melakukannya dengan menggunakan library LiquidCrystal\_I2C.h (bersama dengan library LCD.h).



*Gambar II.2 Konfigurasi pin I2C LCD 20x4*

1. **ALAT DAN BAHAN**
2. Laptop yang sudah terinstall software Arduino IDE 1 buah
3. ARM (STM32F103C8T6) 1 buah
4. I2C LCD 20x4 1 buah
5. ST-Link 1 buah
6. USB TTL 1 buah
7. Push Button 4 buah
8. Kabel jumper secukupnya
9. Kertas Duplex 1 buah
10. Lem tembak 1 buah
11. Cutter 1 buah
12. Gunting 1 buah
13. **PROSEDUR**

**IV.I Pemrograman dan Penyetinggan Pada STM32CubeIDE**

1. Membuka aplikasi STM32CubeIDE pada PC
2. Memastikan USB TTL sudah terdeteksi oleh PC
3. Menuliskan program seperti dibawah ini :
   1. Program Pada task.c

**#include** "main.h"

**#include** "MyTask.h"

**#include** "MyLCD.h"

**#include** "MyUART.h"

**#include** "MySeOS.h"

**unsigned** **char** **Button\_Continue**(**void**);

**unsigned** **char** **Button\_Abort**(**void**);

**unsigned** **char** **Button\_C1000**(**void**);

**unsigned** **char** **Button\_C500**(**void**);

**void** **Led\_Init**(**void**);

**void** **Led\_IndexStart**(**void**);

**void** **Led\_c500**(**void**);

**void** **Led\_Buffer**(**void**);

**void** **Led\_Abort**(**void**);

**void** **Led\_Continue**(**void**);

**#define** Continue HAL\_GPIO\_ReadPin(PbContinue\_GPIO\_Port, PbContinue\_Pin)==GPIO\_PIN\_RESET

**#define** Abort HAL\_GPIO\_ReadPin(PbAbort\_GPIO\_Port, PbAbort\_Pin)==GPIO\_PIN\_RESET

**#define** Pb1000 HAL\_GPIO\_ReadPin(PbC1000\_GPIO\_Port, PbC1000\_Pin)==GPIO\_PIN\_RESET

**#define** pb500 HAL\_GPIO\_ReadPin(PbC500\_GPIO\_Port, PbC500\_Pin)==GPIO\_PIN\_RESET

**#define** BuzzerOn HAL\_GPIO\_WritePin(Buzzer\_GPIO\_Port, Buzzer\_Pin, GPIO\_PIN\_SET)

**#define** BuzzerOff HAL\_GPIO\_WritePin(Buzzer\_GPIO\_Port, Buzzer\_Pin, GPIO\_PIN\_RESET)

**enum** state {*start*, *startdelay*, *finishdelay*, *koinkurang*, *kembali*, *buffer*, *index\_start*, *c500*, *dropStuff*, *dropCoin*, *batal*} Condition;

**volatile** uint32\_t timeout=0;

**volatile** uint32\_t i=0;

**volatile** uint8\_t state\_kembali=0;

**unsigned** **int** Coint\_500=0;

**unsigned** **int** Coint\_1000=0;

**volatile** uint64\_t Jml\_tunai = 0;

**volatile** uint16\_t Coin=0;

**volatile** uint16\_t Coin\_temp=0;

**volatile** uint64\_t Jml\_kembali=0;

**char** dis\_coint;

**\_Bool** Kembali500 =0;

**\_Bool** Kembali1000 =0;

**\_Bool** Abort500=0;

**\_Bool** Abort1000=0;

**\_Bool** AbortTotal=0;

**\_Bool** RunEnable =1;

**void** **MyTask\_Init**(**void**)

{

timeout=0;

Condition=*start*;

Led\_Init();

LCD\_Init();

}

**void** **MyTask\_Run**(**void**)

{

**switch**(Condition){

**case** *start*:

{

LCD\_SetCursor(0, 0);LCD\_Print("Vending Machine");

LCD\_SetCursor(0, 1);LCD\_Print("OKE OCE 2019");

Led\_Init();

Condition=*startdelay*;

**break**;

}

**case** *startdelay*:

{

BuzzerOn;

**if**(++timeout>250000){

LCD\_SetCursor(0, 2);LCD\_Print("Masukkan Coin :");

timeout=0;

BuzzerOff;

MyTask\_Display(RunEnable);

Condition=*index\_start*;

}

**break**;

}

**case** *c500*:

{

Led\_c500();

**if**(Button\_C500()){

Coin=5;

Coin\_temp+=5;

Jml\_tunai+=5;

MyTask\_Display(RunEnable);

Condition=*buffer*;

}

**if**(Button\_C1000()){

Coin=10;

Coin\_temp+=10;

Kembali500=!(Kembali500);

Condition=*kembali*;

}

**if**(Button\_Abort()){

Led\_Abort();

LCD\_Clear();

LCD\_SetCursor(0, 0);LCD\_Print("Transaksi gagal");

Condition=*batal*;

}

**if**(Button\_Continue()){

BuzzerOn;

LCD\_Clear();

LCD\_SetCursor(0, 1);LCD\_Print("Transaksi Gagal");

LCD\_SetCursor(0, 2);LCD\_Print("MAAF COIN KURANG");

Condition=*koinkurang*;

}

**break**;

}

**case** *koinkurang*:

{

**if**(++timeout>1000000){

timeout=0;

LCD\_Clear();

LCD\_SetCursor(0, 0);LCD\_Print("Vending Machine");

LCD\_SetCursor(0, 1);LCD\_Print("OKE OCE 2019");

LCD\_SetCursor(0, 2);LCD\_Print("Masukkan Coin:");

MyTask\_Display(RunEnable);

BuzzerOff;

Condition=*c500*;

}

**break**;

}

**case** *buffer*:

{

Led\_Buffer();

**if**(Button\_C500()){

Coin=5;

Coin\_temp+=5;

Kembali500=!(Kembali500);

Condition=*kembali*;

}

**if**(Button\_C1000()){

Coin=10;

Coin\_temp+=10;

Kembali1000=!(Kembali1000);

Condition=*kembali*;

}

**if**(Button\_Continue()){

Condition=*dropStuff*;

}

**if**(Button\_Abort()){

Led\_Abort();

LCD\_Clear();

LCD\_SetCursor(0, 0);LCD\_Print("Transaksi gagal");

Condition=*batal*;

}

**break**;

}

**case** *dropCoin*:

{

**break**;

}

**case** *dropStuff*:

{

Led\_Continue();

LCD\_Clear();

LCD\_SetCursor(0, 1);LCD\_Print("Minuman Keluar");

LCD\_SetCursor(0, 2);

LCD\_Print("Uang Kembali");LCD\_SetCursor(13, 2);LCD\_PrintNum(Jml\_kembali);LCD\_Print("00");

Condition=*finishdelay*;

**break**;

}

**case** *index\_start*:

{

Led\_IndexStart();

**if**(Button\_C500()){

Coin=5;

Coin\_temp+=5;

Jml\_tunai+=5;

MyTask\_Display(RunEnable);

Condition=*c500*;

}

**if**(Button\_C1000()){

Coin=10;

Coin\_temp+=10;

Jml\_tunai+=10;

MyTask\_Display(RunEnable);

Condition=*buffer*;

}

**break**;

}

**case** *finishdelay*:

{

**if**(++timeout>1000000){

timeout=0;

Coin=0;

Coin\_temp=0;

Jml\_tunai=0;

Jml\_kembali=0;

LCD\_Clear();

Condition=*start*;

}

**break**;

}

**case** *kembali*:

{

**if**(Kembali500){

Jml\_tunai=10;

Jml\_kembali+=5;

Kembali500=!(Kembali500);

MyTask\_Display(RunEnable);

Condition=*buffer*;

}

**if**(Kembali1000){

Jml\_tunai=10;

Jml\_kembali+=10;

Kembali1000=!(Kembali1000);

MyTask\_Display(RunEnable);

Condition=*buffer*;

}

**break**;

}

**case** *batal*:

{

LCD\_SetCursor(0, 2);

LCD\_Print("Uang Kembali");LCD\_SetCursor(13, 2);LCD\_PrintNum(Coin\_temp);LCD\_Print("00");

**if**(++i>30){

i=0;

Condition=*finishdelay*;

}

**break**;

}

}

}

**void** **MyTask\_Error\_Handler**(**\_Bool** Enable, **char** \*pData)

{

}

**void** **MyTask\_Display**(**\_Bool** Enable)

{

**if**(Enable){

LCD\_SetCursor(15, 2);LCD\_PrintNum(Coin);LCD\_Print("00 ");

LCD\_SetCursor(0, 3);LCD\_Print("Total: ");LCD\_PrintNum(Jml\_tunai);LCD\_Print("00 ");LCD\_SetCursor(15, 3);LCD\_PrintNum(Jml\_kembali);LCD\_Print("00 ");

}

}

/\*Sensor\*/

**unsigned** **char** **Button\_Continue**(**void**)

{

**static** **unsigned** **char** bouncing=0xFF;

**unsigned** **char** Flagdetect=0;

**if**(Continue){

**if**(++timeout>500000){

BuzzerOn;

}

**else**{

bouncing=(bouncing<<1);

}

}

**else**{

bouncing=(bouncing<<1)|1;

timeout=0;

}

**if**(bouncing==0x03){

UART\_Print("Tranfer\_OK\n\r");

Flagdetect=1;

}

**return** Flagdetect;

}

**unsigned** **char** **Button\_Abort**(**void**)

{

**static** **unsigned** **char** bouncing=0xFF;

**unsigned** **char** Flagdetect=0;

**if**(Abort){

bouncing=(bouncing<<1);

}

**else**{

bouncing=(bouncing<<1)|1;

}

**if**(bouncing==0x03){

UART\_Print("Transfer\_Cancel\n\r");

BuzzerOff;

Flagdetect=1;

}

**return** Flagdetect;

}

**unsigned** **char** **Button\_C1000**(**void**)

{

**static** **unsigned** **char** bouncing=0xFF;

**unsigned** **char** Flagdetect=0;

**if**(Pb1000){

**if**(++timeout>500000){

BuzzerOn;

}

**else**{

bouncing=(bouncing<<1);

}

}

**else**{

bouncing=(bouncing<<1)|1;

timeout=0;

}

**if**(bouncing==0x03){

UART\_Print("1000\n\r");

Flagdetect=1;

}

**return** Flagdetect;

}

**unsigned** **char** **Button\_C500**(**void**)

{

**static** **unsigned** **char** bouncing=0xFF;

**unsigned** **char** Flagdetect=0;

**if**(pb500){

**if**(++timeout>500000){

BuzzerOn;

}

**else**{

bouncing=(bouncing<<1);

}

}

**else**{

bouncing=(bouncing<<1)|1;

timeout=0;

}

**if**(bouncing==0x03){

UART\_Print("500\n\r");

Flagdetect=1;

}

**return** Flagdetect;

}

/\*Led Configuration\*/

**void** **Led\_Init**(**void**)

{

HAL\_GPIO\_WritePin(Led\_0\_GPIO\_Port, Led\_0\_Pin, *GPIO\_PIN\_SET*);

HAL\_GPIO\_WritePin(Led\_1\_GPIO\_Port, Led\_1\_Pin, *GPIO\_PIN\_SET*);

HAL\_GPIO\_WritePin(Led\_2\_GPIO\_Port, Led\_2\_Pin, *GPIO\_PIN\_SET*);

HAL\_GPIO\_WritePin(Led\_3\_GPIO\_Port, Led\_3\_Pin, *GPIO\_PIN\_SET*);

}

**void** **Led\_IndexStart**(**void**)

{

HAL\_GPIO\_WritePin(Led\_0\_GPIO\_Port, Led\_0\_Pin, *GPIO\_PIN\_RESET*);

HAL\_GPIO\_WritePin(Led\_1\_GPIO\_Port, Led\_1\_Pin, *GPIO\_PIN\_SET*);

HAL\_GPIO\_WritePin(Led\_2\_GPIO\_Port, Led\_2\_Pin, *GPIO\_PIN\_SET*);

HAL\_GPIO\_WritePin(Led\_3\_GPIO\_Port, Led\_3\_Pin, *GPIO\_PIN\_SET*);

}

**void** **Led\_c500**(**void**)

{

HAL\_GPIO\_WritePin(Led\_0\_GPIO\_Port, Led\_0\_Pin, *GPIO\_PIN\_SET*);

HAL\_GPIO\_WritePin(Led\_1\_GPIO\_Port, Led\_1\_Pin, *GPIO\_PIN\_RESET*);

HAL\_GPIO\_WritePin(Led\_2\_GPIO\_Port, Led\_2\_Pin, *GPIO\_PIN\_SET*);

HAL\_GPIO\_WritePin(Led\_3\_GPIO\_Port, Led\_3\_Pin, *GPIO\_PIN\_SET*);

}

**void** **Led\_Buffer**(**void**)

{

HAL\_GPIO\_WritePin(Led\_0\_GPIO\_Port, Led\_0\_Pin, *GPIO\_PIN\_SET*);

HAL\_GPIO\_WritePin(Led\_1\_GPIO\_Port, Led\_1\_Pin, *GPIO\_PIN\_RESET*);

HAL\_GPIO\_WritePin(Led\_2\_GPIO\_Port, Led\_2\_Pin, *GPIO\_PIN\_RESET*);

HAL\_GPIO\_WritePin(Led\_3\_GPIO\_Port, Led\_3\_Pin, *GPIO\_PIN\_SET*);

}

**void** **Led\_Abort**(**void**)

{

HAL\_GPIO\_WritePin(Led\_0\_GPIO\_Port, Led\_0\_Pin, *GPIO\_PIN\_SET*);

HAL\_GPIO\_WritePin(Led\_1\_GPIO\_Port, Led\_1\_Pin, *GPIO\_PIN\_SET*);

HAL\_GPIO\_WritePin(Led\_2\_GPIO\_Port, Led\_2\_Pin, *GPIO\_PIN\_SET*);

HAL\_GPIO\_WritePin(Led\_3\_GPIO\_Port, Led\_3\_Pin, *GPIO\_PIN\_RESET*);

}

**void** **Led\_Continue**(**void**)

{

HAL\_GPIO\_WritePin(Led\_0\_GPIO\_Port, Led\_0\_Pin, *GPIO\_PIN\_RESET*);

HAL\_GPIO\_WritePin(Led\_1\_GPIO\_Port, Led\_1\_Pin, *GPIO\_PIN\_SET*);

HAL\_GPIO\_WritePin(Led\_2\_GPIO\_Port, Led\_2\_Pin, *GPIO\_PIN\_RESET*);

HAL\_GPIO\_WritePin(Led\_3\_GPIO\_Port, Led\_3\_Pin, *GPIO\_PIN\_SET*);

}

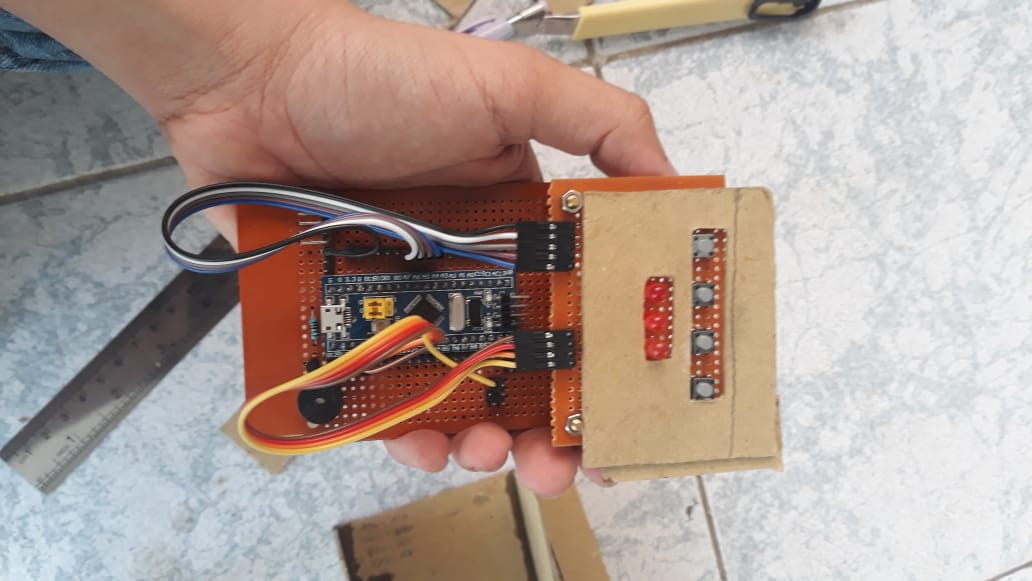
1. Melakukan proses build dan memastikan tidak terjadi error pada program
2. Mem-build dan meng-upload program
3. Mencoba program dan segera melakukan proses.

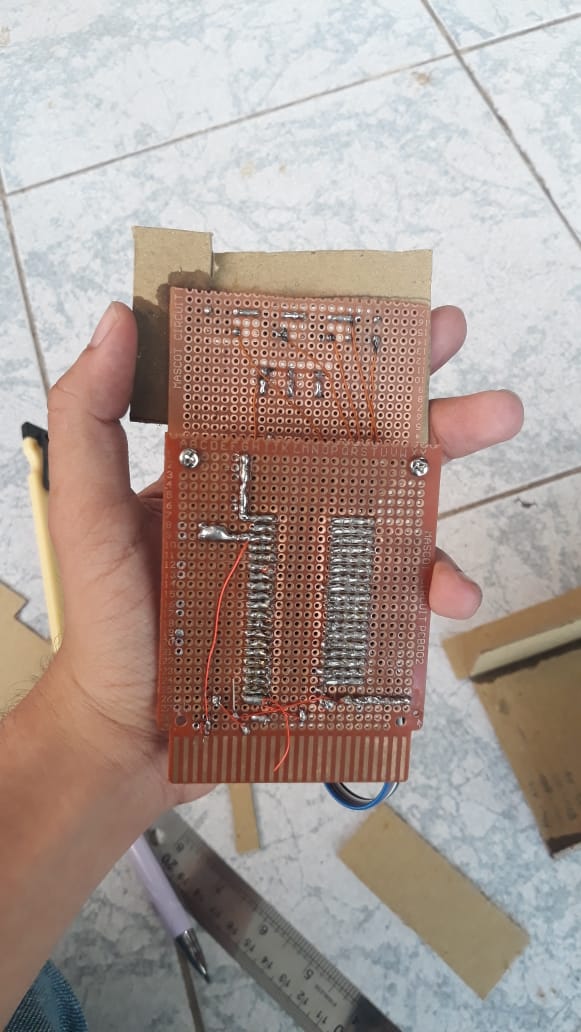
**IV.III Mekanik**

1. Menyiapkan alat dan bahan
2. Membuat rancangan untuk box yang akan diisi dengan rangkaian hardware dengan ukuran sesuai yang diinginkan.

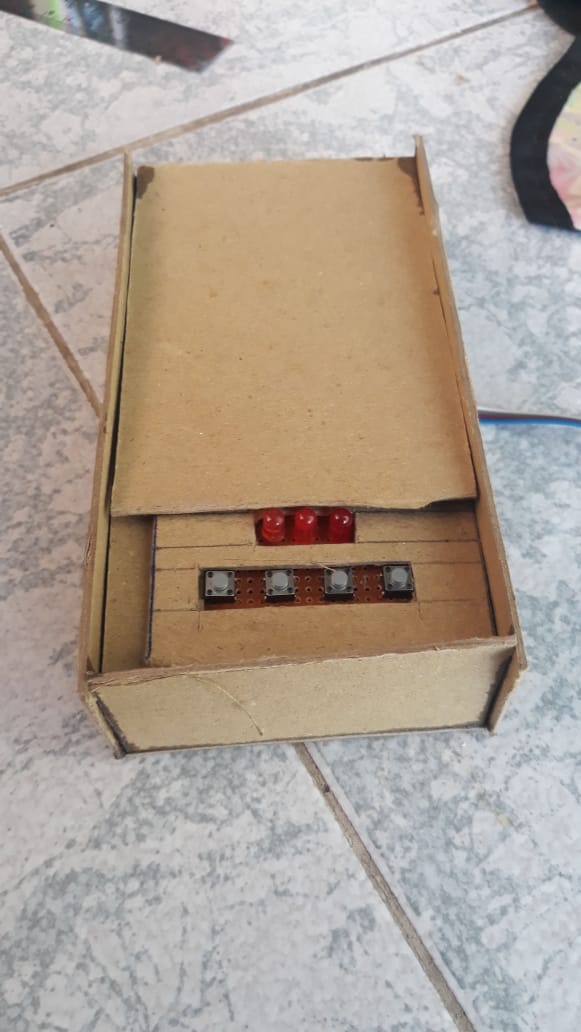
Berikut ini gambar langkah-langkah pembuatan box :

*Gambar 1. Kertas yang sudah siap dirangkai untuk membentuk box*









*Gambar 2. Hardware yang sudah dikemas dalam box*

1. **ANALISA DATA**

Pada project kali ini tentang aplikasi timestate machine with arm pada vending machine dapat dianalisa bahwa ketika kita memasukkan nilai 500 maka state akan menuju s500 dan ketika memasukkan nilai 500 lagi maka state akan menuju s1000 , dan ketika memasukkan nilai 1000 maka state akan menuju s1000 , ketika coin melebihi 1000 maka terdapat kembalian dan coin dapat diproses untuk pembelian sebuah minuman pada vending machine dengan arm dan ditampilkan pada layar LCD.

**KESIMPULAN**

Timestate machine adalah sebuah metode (dalam proses pemrograman) untuk menyederhanakan sebuah permasalahan yang runtut