

LAPORAN AKHIR PRAKTIKUM
SISTEM MIKROPROSESOR DAN MIKROKONTROLER
PERCOBAAN I
ASSEMBLY I



Nama : Donny Mardiansyah
No. BP : 1611512021
Hari/Tanggal : Rabu / 17 Oktober 2018
Kelompok/Shift : 18/III
Anggota Kelompok : Fauzi Akbar
ASISTEN : M. Harits Al-Hakim

RESERVOAR
LAB
ASISTEN : M. Harits Al-Hakim

LABORATORIUM SISTEM EMBEDDED DAN ROBOTIKA
JURUSAN SISTEM KOMPUTER
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS
PADANG
2018

BAB I

PENDAHULUAN

1.1. Tujuan

1. mengenal dan memahami pemrograman ARM dengan menggunakan bahasa assembly
2. memahami tipe data dan register dalam prosesor ARM
3. memahami instruction set dalam prosesor ARM
4. memahami instruction set LDR (load register) atau STR (store register) memory dalam prosesur ARM

1.2. Landasan Teori

Arsitektur ARM merupakan arsitektur prosessor 32 bit RISC. Instruksi set konvensional (RISC) yang dikembangkan oleh ARM. Pada awalnya instruksi prosessor desktop yang sedangnya dibutuhkan oleh keluarga x86 namun desain yang seimbang membuat ARM cocok untuk aplikasi seluler. Sebaliknya prosesor ARM digunakan dalam teknologi listrik, elektronika rumah tangga seperti MP3, mobile phone, media player, dan lain sebagainya.

Arsitektur ARM juga memiliki fitur tambahan sebagai:

1. Instruksi yang menggunakan andora operasi optimisasi dan logika.
2. auto-increment dan auto-decrement addressing mode untuk pengoptimalkan loop program.
3. mengintip berjeda instruksi untuk mensimilasi throughput data.
4. penggunaan secara kondisional untuk mensimilasi algoritma keselusin.

Bahasa assembly merupakan bahasa terendah dari bahasa Pseudo program. Klasifikasi dari assembly adalah bahasa penitifull yang berkait dengan menggunakan program ini langsung tanpa

dalam komunikasi antara register. Bahasa assembly berbentuk -berbentuk
dari bentuk pada arsitektur prosesor yang digunakan seperti Instruksi
sat yang digunakan. Agar dapat diinterpretasi program yang ditulis
dalam bahasa assembly harus dicantumkan dalam bahasa mesin.

Pembahasan ini adalah soal SBC (Simple Board Computer) yang
menggunakan prosesor ARM. ARM merupakan prosesor dengan Instruction
Set Complex yang kurang dari 100 instruksi. RISC memiliki
bentuk sederhana dan mudah dibaca. Setiap satu interpretasi adalah
instruksi dapat diinterpretasi lebih cepat karena sistem RISC mempunyai
kelebihan fungsi dengan menggunakan cycle cycles kelebihannya adalah
keunggulan fungsionalitas seputar.

Bahasa assembly merupakan bahasa komputer yang kelebihannya
adalah bahasa mesin dengan bahasa yang mudah dibaca dan lainnya.

Sebagian besar assembly adalah program yang mencakup perulangan instruksi
dalam format bahasa Inggris sehingga mudah dibaca dan dipahami

Contoh:
MOV R0 #02h
MOV A #03h
ADD A, R0

Perintah pertama berfungsi menyalin proses pengisian
register R0 dengan data 02h. Perintah berikutnya berfungsi menyalin
menjalankan proses pengisian register A dengan data 03h
Kemudian proses penjumlahan data pada register A dengan data
pada register R0 dilakukan dengan perintah ADD A, R0
dengan menghasilkan data 05h disimpan pada register A.

Perintah dari MUL merupakan singkatan dari MUVO
Sebagian ADD merupakan perintah penjumlahan pada salah satu assembly

Bahasa masin adalah kumpulan dari kode biner yang merupakan instruksi yang bisa dijalankan oleh komputer. Adalam komputer hal ini disebut dalam kode halus. Sebagian besar instruksi disimpan dalam kode halus. Sedangkan algoritma dan program matematik sedangkan berada dalam assembly. Untuk mempermudah pengetahuan kode biner agar lebih mudah memahami kode biner untuk mengidentifikasi kode biner agar lebih mudah dimengerti dan mudah dalam penulisan. Program kode biner meliputi dua bagian yakni RIM dan RIO. Kode biner disebut operasi code dan perintah RIM & RIO. Kode biner ini merupakan suatu set perintah yang diberikan dalam bentuk instruksi operation untuk setiap perintah dapat dilihat dalam bentuk instruksi set.

Program yang ditulis dalam bahasa assembly dibentuk dari tiga bagian, Operan 1, Operan 2, bantuan dan setengah. Program ini disebut sebagai program sumber (source code). Source code hanya bisa diolah langsung untuk dilakukan pada prosesot. Untuk dilakukan setengah program source code harus dikonversi dulu ke dalam bahasa mesin.

Source code dicontoh dalam program berikut contohnya: PC kecil atau buku source code dapat dikenakan variabel bahasa mesin disebut setengah. Hasil kerja assembled adalah program binary dan juga assembly listing. assembly listing merupakan hasil yang berasal dari program sumber, pada statis berisi sifat-sifat dari program sumber. Pada statis binary berisi binary yang dinyatakan halus dengan bahasa mesin.

BAB II

PROSEDUR KERJA

2.1. Alat dan Bahan

1. 1 unit Raspberry Pi
2. 1 unit SD card (min ~6GB) dengan OS Raspberry
3. 1 unit SD adaptor
4. 1 unit kabel UTP
5. 1 unit laptop
6. 1 unit kabel data USB to wanita USB

2.2. Cara kerja

Percobaan dimulai

1. Laptop dihidupkan.
2. SD card masukan slot SD Card memory.
3. raspberry pi dihidupkan.
4. laptop ditungguan ke raspberry pi menggunakan kabel UTP
5. komunikasi antara SSH ke raspberry pi melalui laptop dengan menggunakan command : "SSH Pi & C:\>pi>>p 22.

Percobaan 1A) registrasi dan aritmatika sederhana dalam Assembly

- a. Sebagian command dituliskan dengan menggunakan command :
(echo <nama assembly> > nama.fil6) & untuk pada
setiap barisnya dapat menggunakan command : echo < nama
assembly > >> <nama.fil6.s>
- b. Command : as -O <nama.fil6.o> <nama.fil6.s> dilakukan
diatasnya.
- c. Komando gcc -O <nama.fil6.o> <nama.fil6.s> dilakukan.
- d. command <nama.fil6> untuk mengakses program dilakukan.
- e. Program dijalankan.

2. Perbaikan IB: Otrosi load / start sekarang berhasil
- a. Buat program druktuhan dan disimpan dengan nama Latihan 3
- b. Program dimulai



Prosedur Karya (Screenshot)

Registar dan Antwoorden Soalnya

```
1  /* -- latihan1.s */
2  .global Main
3
4
5  main:
6      mov r1, #3          /* r1 = 3 */
7      mov r2, #4          /* r2 = 4 */
8      add r0, r1, r2      /* r0 = r1 + r2 */
9      bx lr
```

Ototani load / Roto Soalnya

```
1  /* -- latihan2.s */
2  .global main
3
4
5  main:
6      mov r0, #3          /* r0 = 3 */
7      mov r1, #4          /* r1 = 4 */
8      add r0, r0, r1      /* r0 = r0 + r1 */
9      bx lr
```

Latihan 3

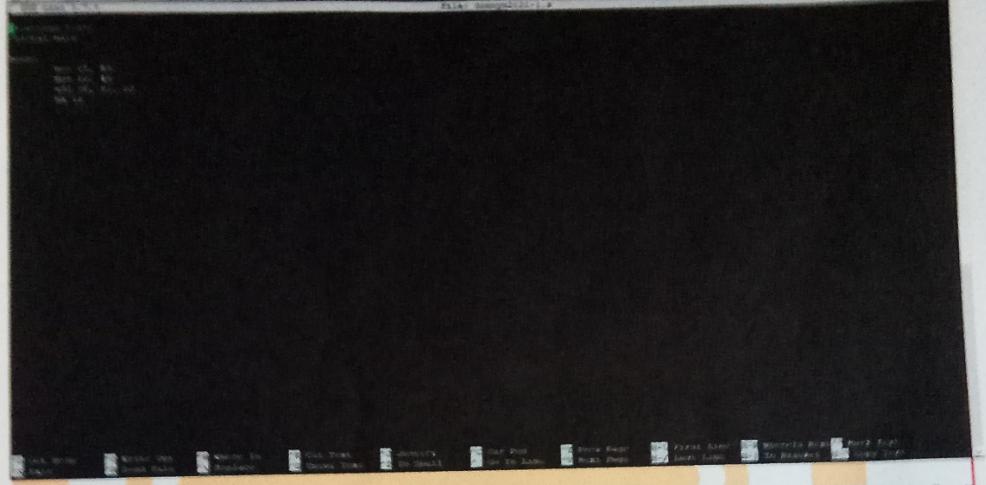
```
1  /* -- latihan3.s */
2
3  /* -- Data section */
4  .data
5
6  /* Ensure variable is 4-byte aligned */
7  .balign 4
8  /* Define storage for myvar1 */
9  myvar1:
10    /* Contents of myvar1 is just 4 bytes containing value '3' */
11    .word 3
12
13 /* Ensure variable is 4-byte aligned */
14 .balign 4
15 /* Define storage for myvar2 */
16 myvar2:
17    /* Contents of myvar2 is just 4 bytes containing value '4' */
18    .word 4
19
20 /* -- Code section */
21 .text
22
23 /* Ensure code is 4 byte aligned */
24 .balign 4
25 /* Global main */
26 main:
27     ldr r1, addr_of_myvar1 /* r1 = &myvar1 */
28     ldr r1, [r1]             /* r1 = *r1 */
29     ldr r2, addr_of_myvar2 /* r2 = &myvar2 */
30     ldr r2, [r2]             /* r2 = *r2 */
31     add r0, r1, r2          /* r0 = r1 + r2 */
32     bx lr
33
34 /* Labels needed to access data */
35 addr_of_myvar1 : .word myvar1
36 addr_of_myvar2 : .word myvar2
```

```
lctmon q
1  /* -- latihan4.s --
2
3  /* -- Data section --
4  .data
5
6  /* Ensure variable is 4-byte aligned */
7  .balign 4
8  /* Define storage for myvar1 */
9  myvar1:
10    /* Contents of myvar1 is just '3' */
11    .word 0
12
13  /* Ensure variable is 4-byte aligned */
14  .balign 4
15  /* Define storage for myvar2 */
16  myvar2:
17    /* Contents of myvar2 is just '4' */
18    .word 0
19
20  /* -- Code section --
21  .text
22
23  /* Ensure function section starts 4 byte aligned */
24  .balign 4
25  .global main
26  main:
27    ldr r1, addr_of_myvar1 /* r1 - &myvar1 */
28    mov r3, #3             /* r3 - 3 */
29    str r3, [r1]           /* *r1 - r3 */
30    ldr r2, addr_of_myvar2 /* r2 - &myvar2 */
31    mov r3, #4             /* r3 - 4 */
32    str r3, [r2]           /* *r2 - r3 */
33
34    /* Same instructions as above */
35    ldr r1, addr_of_myvar1 /* r1 - &myvar1 */
36    ldr r1, [r1]           /* r1 - *r1 */
37    ldr r2, addr_of_myvar2 /* r2 - &myvar2 */
38    ldr r2, [r2]           /* r2 - *r2 */
39    add r0, r1, r2
40    bx lr
41
42  /* Labels needed to access data */
43  addr_of_myvar1 : .word myvar1
44  addr_of_myvar2 : .word myvar2
```

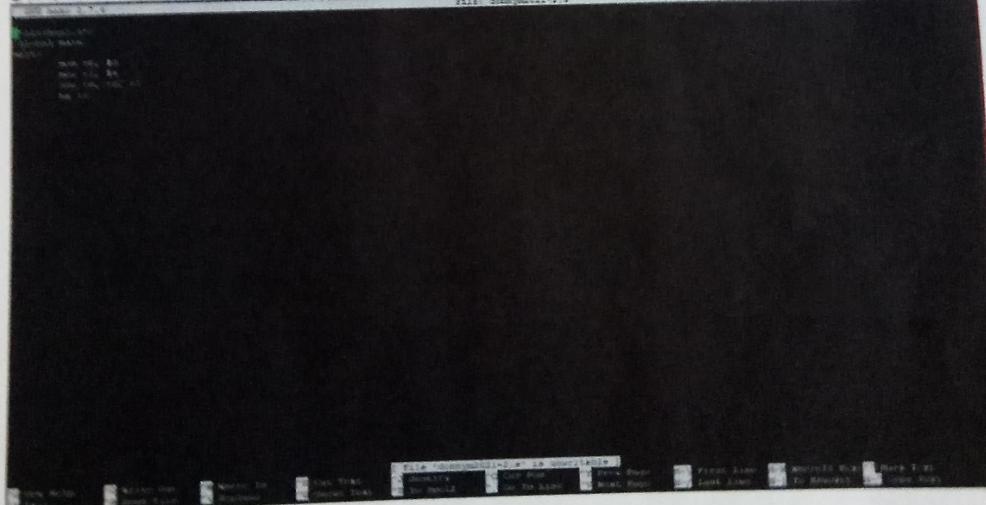
BAB III

DEMBAHASAN

1. Simpaku Perceptron 1



2. Simpaku Perceptron 2



Pada perceptron 1 dan 2 juga menggunakan Mod untuk
memindahkan nilai us register dan add untuk perceptron 2 register

A photograph of a computer monitor showing two terminal windows. The left terminal window displays a command-line interface with several lines of text output, including '2020-06-10T10:45:00Z' and '2020-06-10T10:45:00Z'. The right terminal window also displays a command-line interface with similar text output. A yellow sticky note is placed across the middle of the screen, with handwritten text '3. Create Revision 3' written on it.

g. Synax Percolation 4.

John Ferguson 3 direction revised IDR used about data

In dilemma Rondolalar ADD nilai register

Brian Kershaw q. dice que el personal de la fin. PTA viene reformar

Lipa dan dilakukan Perintah dan ADD nilai registrasi

Beda Persepsi & Sosial Toleransi yang didapat sekolah yang lain

Inter registraz hospit sema uerona hong sappada sedicut

Peterson 9.

Pada pendekan 3 den 1 dapat dituliskan berikut fungsi dan karakteristiknya

BAB IV ANALISA

Pada Praktikum mikroprosesor dan mikromerkurial

Untuk ini kita memerlukan percoalan Proses Register dan aktivitas sedangkan yang dilakukan dengan berfungsi menyimpan

Yang dibutuhkan pada software Putty yang nantinya berfungsi sebagai terminal untuk uraikan menggunakan sintax

Pada percoalan yang dilakukan ini dilakukan dengan menggunakan linguistik sesuai kriteria. Perkona - keru dibawa dalam apliasi

Putty Scolah itu merupakan IP address yang ingin diakses dan keru berakar dan perluan sudi terhadap dengan hasilnya

Pada wifinger mengecek ip pada Putty kaitan agar komunikasi dengan scruct. Untuk wasw ke terminal Putty memerlukan id manusia keru pi dan raspberry pi.

Pada percoalan yang pertama kita menggunakan Syntax

Yang berantai apabila nilai dari ti ditarikkan dengan nilai dari rz maka akan dilakukan nilai pada ko

dengan adanya potongan odd untuk Pemotongan dan

misal value mengakibat nilai yang ada. Scolah itu

ditentukan syntax as -o nama file -o nama file -> untuk

Penggunaan nama folder ataupun nome file yang digunakan

Scolah itu dicantum GCC -o nome file nome files

Scolah itu dijalankan dengan perintah ./nambah

Pada Perintah ketiga dengan instruksi operasi load/store
Salah satu assembly dilakukan dengan cara dan langkah berikut.
Bentuknya : 1. Jika 2. mempunyai modifikasi synax yang berbeda.
Yaitu dengan Perintah LDR dengan Penggunaan 2 variabel pada
hasil yang diambil ditulis pada tanda info register drama
dari Jarak 2 hasil yang berbeda pada tanda info register drama
dapat ditulis pada bagian 3. Pembahasan.

Pada Perintah ketiga Program yang dilakukan lebih banyak
dengan Penggunaan Perintah STR, LDR, MOV, ADD, SUB
Untuk melakukan hal ini diperlukan sebanyak dua buah hasil yang
dapat ditulis pada bagian 3. Pembahasan.

Pada Perintah ketiga juga dilakukan pembahasan bagian drama
Yaitu GCR untuk Pengalihan Subprogram GCR untuk
mengakses hasil yang dilakukan operasi eksekusi dan gunanya dalam penggunaan
sebanyak 1bu dilakukan info register. Adapun dapat dituliskan
nilai yang besar pada bagian 3.2 drama pada bagian
yang digunakan untuk setiap variabel maka data, penggunaan
seperti berikut, dapat dituliskan nilai masing perintah
di bagian 3.2.

Pada hasil Program yang dilakukan pada Perintah yang
dimulai dapat dituliskan sebagaimana pada info register
dan pada dasarnya secara dapat dituliskan nilai awalnya
baik dari operasi yang dilakukan.

BAB V PENUTUP.

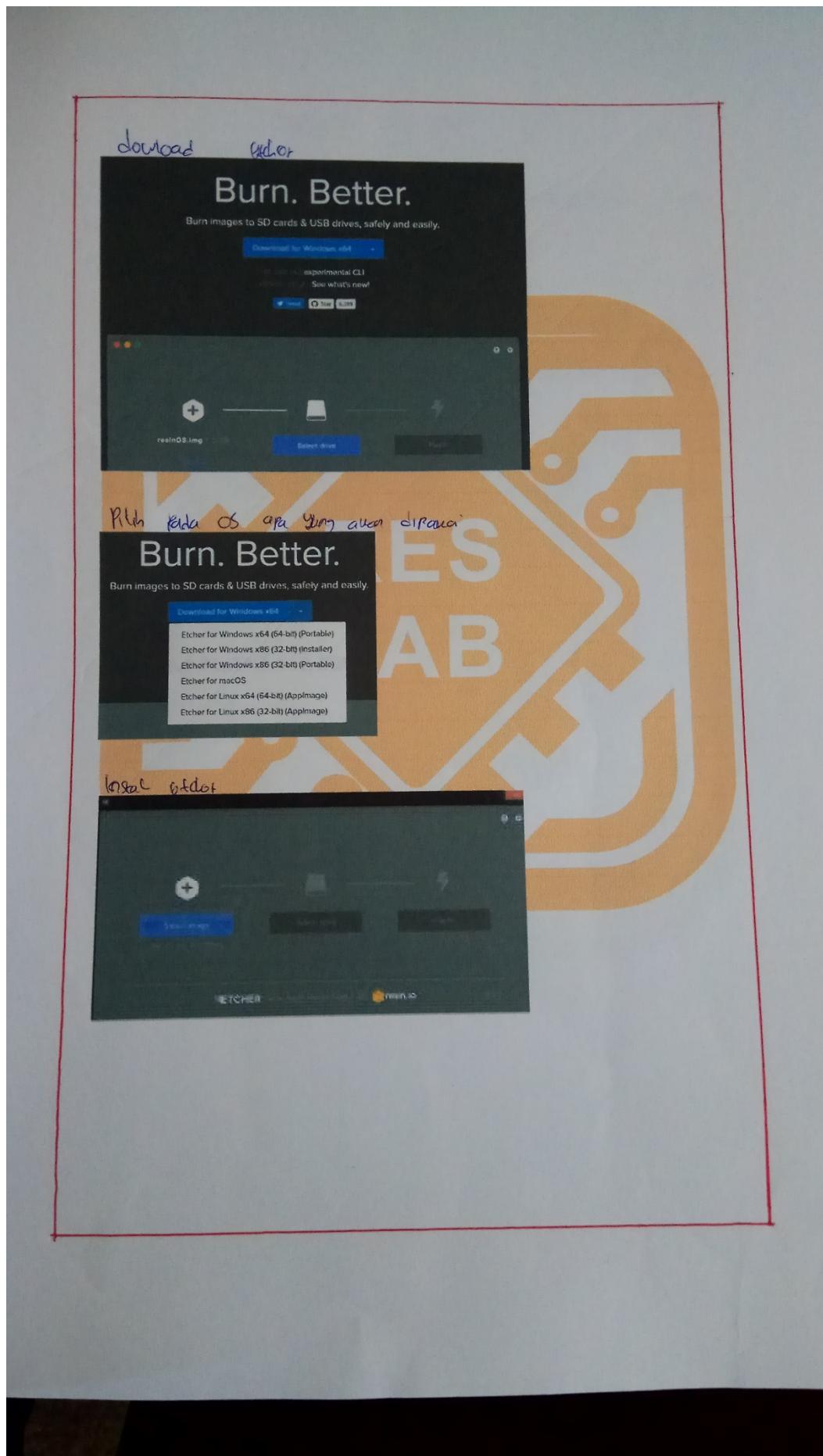
1. KESIMPULAN

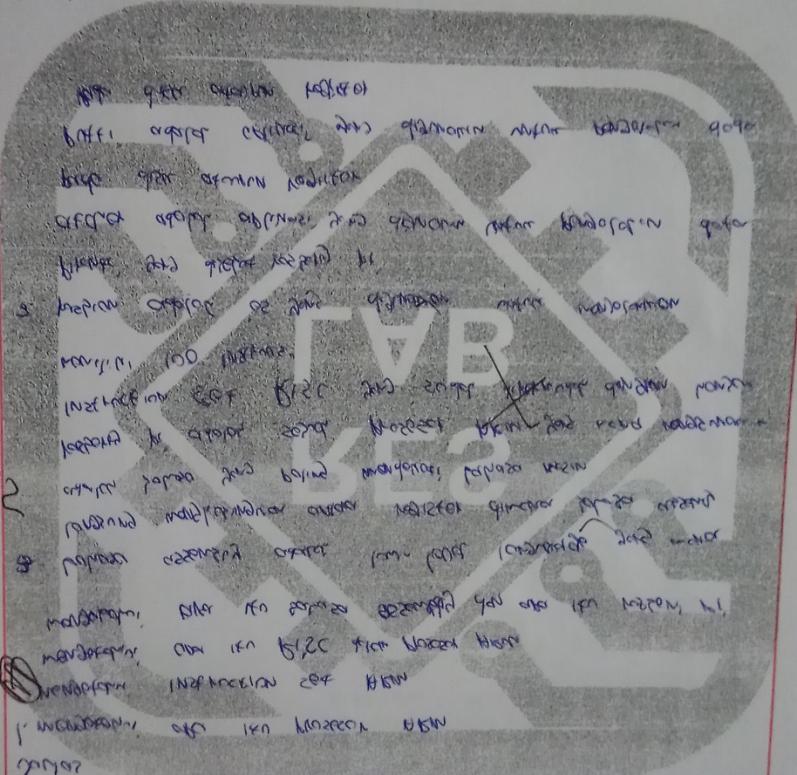
1. Perintah STR digunakan untuk memindahkan nilai dari register ke alokasi memori yang ditentukan oleh register destination.
2. Perintah LDR digunakan untuk mendapatkan nilai dari register memory yang ditentukan oleh register destination.
3. Langkah awal dalam program low-level adalah dengan menginisiasi register dan memasukkan nilai ke dalamnya.
4. raspberry Pi adalah sistem operasi yang menggunakan protokol Alan Turing.

2. SARAN

1. Praktikum ditujukan untuk pelajaran praktik.
2. Praktikum ditujukan kepada siswa-siswi untuk mendapat keselarasan.
3. Praktikum ditujukan untuk diperbandingkan dengan teori dan hasil dalam pelajaran praktikum.





45 15/10 { 52 } 

gründen müssen
wenn sie nicht
eine Organisation haben möchten ist, dass:

105 15/10 { 52 }

→ Gründen kann
nicht mehr als fünf Personen
haben und diese müssen alle gleichzeitig

10

106 geben können heißt es

mindestens eine Person muss mindestens 18 Jahre

alte sein um zu gründen

und es darf nur eine Person sein die die Organisation

gründet und darf nicht

mindestens 100 Minuten

Informationen geben darf und darf nicht mehr als

100 Minuten pro Tag geben

Informationen geben darf und darf nicht mehr als

100 Minuten pro Tag geben

Informationen geben darf und darf nicht mehr als

100 Minuten pro Tag geben

Informationen geben darf und darf nicht mehr als

100 Minuten pro Tag geben

Informationen geben darf und darf nicht mehr als

100 Minuten pro Tag geben

Informationen geben darf und darf nicht mehr als

100 Minuten pro Tag geben

Informationen geben darf und darf nicht mehr als

100 Minuten pro Tag geben

Informationen geben darf und darf nicht mehr als

100 Minuten pro Tag geben

Informationen geben darf und darf nicht mehr als

100 Minuten pro Tag geben

Informationen geben darf und darf nicht mehr als

100 Minuten pro Tag geben

Informationen geben darf und darf nicht mehr als

100 Minuten pro Tag geben

Informationen geben darf und darf nicht mehr als

100 Minuten pro Tag geben

10

15

15

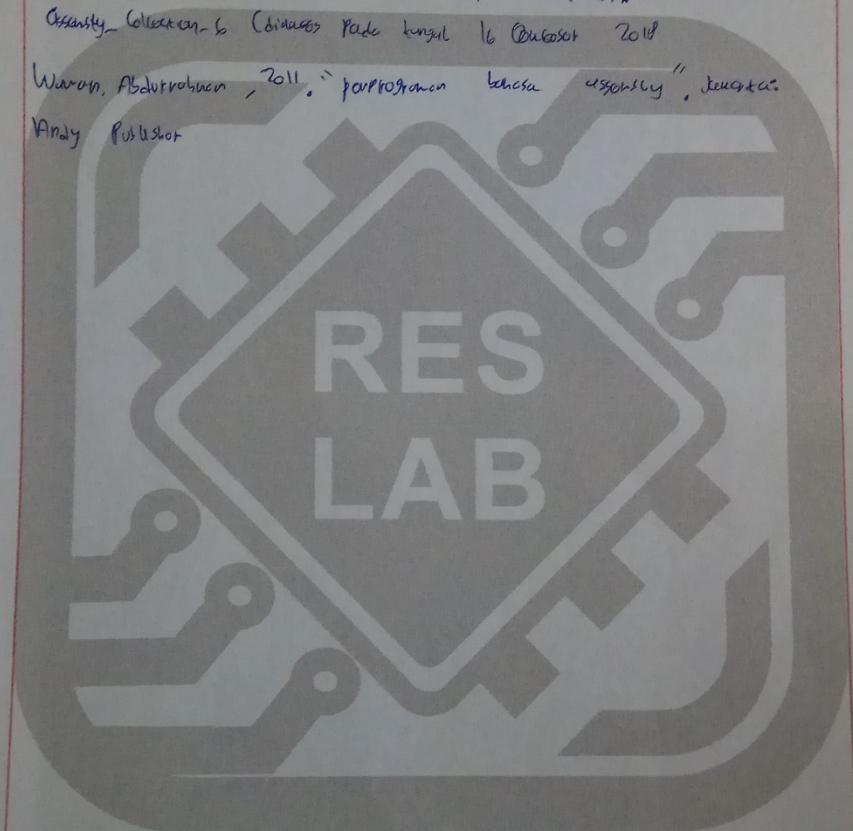
2. 10

DAFTAR PUSTAKA

Dinus, "Assembly" <http://dinus.ac.id/> > ajar_bahasa - assembly diases
Pada tanggal 16 Oktober 2018

Twinl & Co, "Assembly Collection", <http://www.Twinl.co/2011/03/20/>
Assembly Collection-6 (diases pada tanggal 16 Oktober 2018)

Wawan, Abdurrahman, 2011. "perprograman bahasa assembly". Jakarta:
Andy Publisher



LAPORAN AWAL PRAKTIKUM
SISTEM MIKROPROSESOR DAN MIKROKONTROLLER
MODUL I
ASSEMBLY I

10
25
20
10
9
77



Nama : Donny Mardiansyah
No. BP : 1611512021
Hari/Tanggal : Kamis/ 17 Oktober 2018
Kelompok/Shift : 18/I
Anggota Kelompok : Fauzi Akbar

RES
LAB

LABORATORIUM SISTEM EMBEDDED DAN ROBOTIKA
JURUSAN SISTEM KOMPUTER
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS
PADANG

2018