#### **BAB IV**

### PENGUJIAN DAN EVALUASI SISTEM

Pengujian dan evaluasi sistem pada tugas akhir ini meliputi perangkat keras elektonika dan perangkat lunak yang telah dibuat. Pengujian pada perangkat keras elektronika menggunakan program simulasi *Proteus ISIS 7 Professional* sedangkan pengujian pada perangkat lunak dilakukan dengan menggunakan program simulasi *ProView32 Version 3.3 (Franklin Sftware Inc)*.

Dari pengujian ini akan didapatkan data-data maupun bukti-bukti bahwa sistem yang telah dibuat dapat bekerja dengan baik. Berdasarkan data-data dan bukti-bukti tersebut, akan dapat dilakukan analisa terhadap proses kerja, yang nantinya dapat digunakan untuk menarik kesimpulan dari apa yang telah dijelaskan dalam tugas akhir ini.

#### 4.1. Pengujian Perangkat Keras Elektronika

Pengujian perangkat keras elektronika ini meliputi (modul mikrokontroler, komunikasi antar mikrokontroler, rangkaian driver motor MOC3021 dan rangkaian driver motor L298).

#### 4.1.1 Modul Mikrokontroler

### A. Tujuan

Tujuan dari pengujian rangkaian mikrokontroler ini adalah untuk mengetahui apakah rangkaian yang dibuat dapat berfungsi dengan baik. Dimana rangkaian mikrokontroler ini digunakan untuk mengatur dan mengendalikan semua proses produksi dan pengemasan sambal pecel.

## B. Peralatan yang Digunakan

- 1. Program simulasi Proteus ISIS 7 Professional.
- Rangkaian minimum sistem AT89C52 (proteus tidak menyediakan mikrokontroler AT89S52).
- 3. Catu daya 5 Volt.
- 4. Rangkaian LED sebanyak tuju buah (sebagai *display* atau output simulasi).

### C. Prosedur Pengujian

1. Minimum sistem dan LED dirancang seperti pada Gambar 4.1. berikut. Ke tuju buah LED di hubungkan pada *port* P3 (*pin* P3.0-P3.7).



Gambar 4.1. Rangkaian Pengujian Minimum Sistem

2. Membuat program sederhana untuk menyalakan lampu LED secara berjalan dengan menggunakan program ProView32. Setelah program dibuat

kemudian dilakukan debuger untuk membuat file.hex yang nantinya akan dimasukkan kedalam AT89C52.

Program sederhana pengujian minimum sistem:

	; file name:	tesmikro.	.asm					
	ORG	000H						
		MOV	A, #1111	1110 <b>B</b>				
	MULAI:	MOV ACALL RL SJMP	P3, DELAY A MULAI	A				
	DELAY:	MOV	R0,	#10H				
	DELAY1:	MOV	R1.	#0FFH				
	DELAY2:	MOV	R2.	#0H				
		DJNZ	R2.	\$				
		DJNZ	R1,	DELAY2				
		DJNZ	R0,	DELAY1				
		RET						
	END							
3.	File. <mark>hex</mark>	(tesmi	kro.hex	) dimasukkan k	kedalam	mikroko	ontroler	AT89C52
	dengan o	cara doi	ıble clic	k pada IC AT890	C52.			

Edit Component		? 🛛
Component <u>B</u> eference: Component <u>V</u> alue: PCB Package: Program File: ( Clock Frequency: Advanced Properties: Simulate Program Fetches <u>●</u> Other <u>P</u> roperties:	TESMIKR0       Hidden:         AT89C52       Hidden:         DfL40       ?        Proteus_TA_Edit\TESMIKR0.HEX       Hide All         12MHz       Hide All         No       ✓	QK Help Data Hidden <u>P</u> ins <u>C</u> ancel
Exclude from <u>S</u> imulation Exclude from PCB <u>L</u> ayout Edit <u>all</u> properties as text	Hide common pins	

Gambar 4.2. Cara Memasukkan File.hex Kedalam IC AT89C52

4. Tekan tombol play pada menu window untuk memulai simulasi dan lihat perubahan yang terjadi pada ke 7 LED (*display*).



Gambar 4.3. Tombol Play Untuk Memulai Simulasi

# D. Hasil Pengujian

Berikut ini adalah hasil pengujian dari modul mikrokontroler yang dilakukan dengan menggunakan program *proteus ISIS 7 professional*.



Gambar 4.4. Tampilan LED Menyala Secara Berjalan

Output LED (Display)	Address	Value	Keterangan
Port P3	0x0030	0b 11111110	Lampu LED menyala
( <i>pin</i> P3.0-P3.7)		0b 11111101	dari atas ke bawah atau
		0b 11111011	dari kanan ke kiri (RL),
		0b 11110111	kemudian kembali lagi
		0b 11101111	dari awal sampai
		0b 11011111	simulasi dihentikan
		0b 10111111	(diakhiri).
		0b 01111111	

Tabel 4.1 Hasil Pengujian Mikrokontroler

### E. Analisa

Dari hasil pengujian di atas dapat disimpulkan bahwa mikrokontroler AT89C52 dapat berfungsi dengan baik sesuai dengan perancangan yang dibuat.

### 4.1.2 Komunikasi Antar Mikrokontroler

#### A. Tujuan

Tujuan dari pengujian komunikasi antar mikrokontroler ini adalah untuk mengetahui apakah komunikasi yang dilakukan antara mikrokontroler pertama dengan mikrokontroler kedua dapat berjalan dengan baik.

### B. Peralatan yang Digunakan

- 1. Program simulasi Proteus ISIS 7 Professional.
- 2. Dua buah rangkaian minimum sistem AT89C52.
- 3. Tuju buah LED sebagai *disply* simulasi.
- 4. *Limit switch* sebagai inputan pengiriman data mikrokontroler kedua kepada mikrokontroler pertama.
- 5. Catu daya 5 Volt.

# C. Prosedur Pengujian

 Kedua buah rangkaian minimum sistem AT89C52 (mikrokontroler pertama dan kedua) serta ketuju buah LED dirangkai seperti pada Gambar 4.5 berikut. Komunikasi antar mikro menggunakan *port* P2 (*pin* P2.0-P2.7), inputan limit switch menggunakan *port* P1 (*pin* P1.0) sedangkan *output* simulasi (tuju buah LED) menggunakan *port* P3 (*pin* P3.0-P3.7).





 Membuat program komunikasi sederhana pada kedua buah mikrokontroler dengan menggunakan program ProView32. Setelah program dibuat kemudian dilakukan debuger untuk membuat file.hex yang nantinya akan dimasukkan kedalam kedua buah AT89C52.

Program sederhana komunikasi antar mikrokontroler:

; file name: teskomunikasimikro-1.asm



#### ; file name: teskomunikasimikro-2.asm

BIT	P1.0
P0, #00	H
P1, #00	H
P2, #00	)H
P3, #00	)H
	BIT P0, #00 P1, #00 P2, #00 P3, #00

JNB SAKLAR,\$ ;kirim data 11h ke mikro 1 MOV A,#11H MOV P2,A DELAY LCALL DELAY: MOV R0, #5H DELAY1: MOV #0FFH R1, DELAY2: MOV R2, #0H DJNZ R2, \$ DJNZ R1, DELAY2 DELAY1 DJNZ R0, RET END

 File.hex (teskomunikasimikro-1.hex dan teskomunikasimikro-2.hex) dimasukkan kedalam masing-masing mikrokontroler AT89C52 dengan cara seperti pada Gambar 4.2. prosedur pengujian modul mikrokontroler di atas.
 Tekan tombol play untuk memulai simulasi. Seperti pada Gambar 4.3. prosedur pengujian modul mikrokontroler di atas.

# D. Hasil Pengujian

MULAI:

Berikut ini adalah hasil pengujian komunikasi antar mikrokontroler yang dilakukan dengan menggunakan program *proteus ISIS 7 professional*.

• Kondisi sebelum terjadinya komunikasi (*limit switch* pada mikro 2 *pin* P1.0 belum ditekan atau mikro 2 belum mengirim data #11H ke mikro 1).

	110 P3.2/INT0 13 P3.2/INT0 13 P3.3/INT1 14 P3.3/INT1 16 P3.6/INT 16 P3.6/INT 17 P3.6/INT P3.7/RD AT\$9C52 (MIKF	P1.1/	T2EX = 2 P1.2 = 4 P1.3 = 6 P1.4 = 6 P1.6 = 7 P1.6 = 7 P1.7 = 8	
• • • • • D5 • • • • • •	Watch Window			X
• • • D6 🚰 🐂 🔤 • 🔥 •	Name	Address	Value	Previous Value
	P2	0×0020	0×00	O×FF
	P1	0×0010	0×00	O×FF
	P3	0×0030	0b00000000	Ob1111111

Gambar 4.6. Sebelum Terjadi Komunikasi.

• Kondisi Setelah terjadi komunikasi (*limit switch* pada mikro 2 *pin* P1.0 ditekan atau mikro 2 mengirim data #11H ke mikro 1). LED 4-bit pertama dan LED 4-bit kedua akan menyala dan mati secara bergantian.



Gambar 4.7. Setelah Terjadi Komunikasi

Tabel 4.2. Hasil Pengujian Komunikasi Antar Mikrokontroler

Limit Input		Value	Keterangan
switch	(Mikro 2)	(Output Mikro 1)	
Off 0x00		0b 11110000	LED 4-bit pertama (pin P3.0-P3.3)
			menyala dan LED 4-bit kedua (pin
			P3.4-P3.7) mati.
On	<b>0</b> x11	0 <mark>b 0</mark> 0001111	LED 4-bit pertama dan LED 4-bit
		0b 11110000	kedua menyala dan mati secara
		0b 00001111	bergantian. Hal ini akan terus
_		0b 11 <mark>110</mark> 000	dilakukan sampai simulasi dihentikan
		0b 00001111	atau limit switch dimatikan.

### E. Analisa

Dari hasil pengujian komunikasi antar mikrokontroler, dapat ditarik kesimpulan bahwa *output, input,* dan komunikasi kedua mikrokontroler dapat berjalan dengan baik sesuai dengan perancangan sistem.

### 4.1.3 Driver Motor MOC3021

### A. Tujuan

Tujuan dari pengujian driver MOC3021 ini adalah untuk mengetahui apakah rangkaian dapat mengendalikan motor AC dengan baik.

## B. Peralatan yang Digunakan

- 1. Program simulasi Proteus ISIS 7 Professional.
- 2. Rangkaian driver MOC3021.
- 3. Lampu AC (*proteus* tidak menyediakan motor AC)
- 4. Limit switch sebagai masukan 5 Volt pengganti inputan dari mikrokontroler
- 5. Catu daya 5 Volt.
- 6. Catu daya 220 Volt.



Gambar 4.8. Rangkaian Driver Motor MOC3021

- 2. Tekan tombol play pada menu window *proteus* untuk memulai simulasi.
- 3. Tekan *limit switch* untuk menghidupkan lampu (inputan 5 Volt sebagai masukan IC MOC3021).

### D. Hasil Pengujian

Tabel 4.3. Hasil Pengujian Driver Motor MOC3021

Limit switch (inputan 5 Volt)	Lampu AC pengganti motor AC
Pasif	Mati
Aktif	Menyala

### E. Analisa

Dari hasil pengujian driver motor MOC3021 dapat ditarik kesimpulan bahwa rangkaian dapat berjalan dengan baik sesuai dengan rancangan.

# 4.1.4 Driver Motor L298

## A. Tujuan

Tujuan dari pengujian ini adalah untuk mengetahui apakah rangkaian driver L298 dapat mengendalikan motor DC dengan baiak.

## B. Peralatan yang Digunakan

- 1. Program simulasi Proteus ISIS 7 Professional. A B A
- 2. Rangkaian driver L298.
- 3. Dua buah motor DC.
- 4. Dua buah *limit switch*.
- 5. Catu daya 5 Volt

## C. Prosedur Pengujian

 Driver L298, motor DC dan *limit switch* di rangkai seperti pada Gambar 4.9. berikut.



Gambar 4.9. Rangkaian Driver Motor L298

Tekan tombol play pada menu window *proteus* untuk memulai simulasi.
 Tekan *limit switch* untuk mengatur putaran motor (*limit switch* 1= putaran motor berlawanan dengan arah jarum jam dan *limit switc* 2= putaran motor searah dengan arah jarum jam).

## D. Hasil Pengujian

Taber 4.4. Hash Pengujian Driver Motor L298						
Limit switch	Limit switch	Respon Motor	Arah Putaran Motor			
(1)	(2)					
Pasif	Pasif	Diam	ADATA			
Aktif	Aktif	Diam	-			
Aktif	Pasif	Berputar	Berlawanan arah jarum jam			
Pasif	Aktif	Berputar	Searah jarum jam			

Tabel 4.4. Hasil Pengujian Driver Motor L298

### E. Analisa

Dari hasil pengujian terhadap rangkaian driver motor L298 di atas, maka dapat ditarik kesimpulan bahwa rangkaian dapat mengendalikan motor DC dengan baiak.

#### 4.2. Pengujian Perangkat Lunak

### A. Tujuan

Tujuan dari pengujian ini adalah untuk mengetahui apakah program (perangkat lunak) yang telah dibuat dapat menangani *input* dan *output* sesuai dengan cara kerja dari sistem yang di buat.

#### B. Peralatan

Peralatan yang dibutuhkan untuk melakukan pengujian terhadap sistem ini adalah: "Sebuah PC yang digunakan untuk menjalankan program simulasi serta program simulasi ProView32 yang digunakan untuk melakukan simulasi dari program *asembly* yang dibuat pada mikrokontroler pertama dan mikrokontroler kedua.

#### C. Prosedur Pengujian

Program mikrokontroler pertama dan mikrokontroler kedua dibuat dengan menggunakan bahasa *asembly* melalui program ProView32, kedua buah mikrokontroler diprogram secara terpisah dengan nama dan fungi yang berbeda. Program mikrokontroler pertama disimpan dengan nama file (*Produksi.asm*) yang diprogram untuk menangani proses-proses pada peralatan produksi sambal pecel, sedangkan mikrokontroler kedua disimpan dengan nama file (*Pengemasan.asm*) yang diprogram untuk menanagani semua proses yang terjadi pada mesin pengemasan sambal pecel. Setelah program selesai dibuat, langkah selanjutnya adalah menjalankan program untuk mengetahui sukses dan tidaknya terhadap program yang telah kita buat. Caranya adalah, pada *keyboard* tekan tombol [Ctrl] + [D] atau pada menu window pilih <u>D</u>ebag kemudian pilih Start, Gammbar 4.10. Menu *ProView32* untuk melakukan *debag*.



Gambar 4.10. Menu ProView32

Bila dalam pembuatan program ini terdapat kesalahan dalam penulisan *syntax*, prosedur atu yang lain maka pada window *message* akan ditampilkan pesan kesalahan-kesalahan dari program yang kita buat, tampilan pesan kesalahan dapat dilihat pada Gambar 4.11. berikut:

			and the second s	
d:\\mikro_001.asm				🖬 Message 👘 🕢 🗰
KG TOON				E- Kunning AS1 on d:\stikom\data ta_irfan\program_asm_mikro\mikro_391.asm
ENSOR RACANG	BIT PL.O			
ENSOR BUMBU	BIT P1.1			ERROR #46 IN LINE 56 OF mikro_UD_65m : UNDEFINED SYMBOL (PASS-2) MOTOR_AC_KACANG_AKT
ENSOR CAMPURAN	BIT P1.2		<b>H</b>	
ENSOR PENAMPUNGAN	BIT P1.3			
ENSOR KAL ATAS	BIT P1.4			
ENSOR KAL BAWAH	BIT P1.5			
OTOR_AC_KACANG	BIT PO.O			
OTOR_AC_CAMPURAN	BIT PO.1			
OTOR_DC_KAL_NAIK	BIT PO.2			
OTOR_DC_KAL_TURON	BIT PO.3			
JHP START				
OTOR_AC_KACANG_ART	F;			
SETB MOTOR_AC_K	CANG ; kac	ang dan bumbu		
OTOR AC MACANG MAT				
CLR MOTOR AC K	CANG : kac	ang dan hunhu		
and the second second		and a second second		
OTOR AC CAMPURAN AN	TIF:			
	and the second second			

Gambar 4.11. Pesan Kesalahan Program

Setelah program dilakukan pembetulan dari kesalahan-kesalahan yang ada, maka tampilan dari ProView adalah seperti Gambar 4.12. berikut.

CHG LOOM SENSOR JAAANG BIT P1.0	Message     Message     B. Running LSI on D/\STROM/Data TA_RFAN/PROGRAM_A     Running C/\FSN8IN/OH51V32.DLL on D/\STROM/Data TA	SM_MBROVMERO_RELASM
SENSOR_BUMPUPAN BIT P1.2		
BERNEY DALLANS AND STIT F1.3 NOTOR JC SACARS BIT F0.3 NOTOR JC SACARS BIT F0.3 NOTOR JC SACARS BIT F0.3 NOTOR JC SAL BARK BIT F0.3 STIT F0	Mark Register (M050 0):00:00         Constraint (M050 0):00:00         Constraint (M050 0):00:00         Notice         Notice	Andream Connection     Andream Connectio
Code (mikre_003)           Grees [Symbol         [Code           14 LUMP START           105         020127           LUMP START           107         SETB MOTOR AC KACANG ; kscang dan bumbu           108         MOTOR AC KACANG ; kscang dan bumbu           109         20127           20         CIR MOTOR AC KACANG ; kycang dan humbu           20         CIR MOTOR AC KACANG ; kycang dan humbu		A Connection No Connection No Connection No Connection No Connection No Connection No Connection No Connection No Connection No Connection

Gambar 4.12. Tampilan Program Yang Sudah Benar

# D. Hasil Pengujian

Untuk mengetahui hasil pengujian dari program yang telah kita buat, kita dapat melihat perubahan nilai yang terdapat pada *main register* atau perubahan warna indikator *view hardware* pada masing-masing port.

Dalam simulasi ini perubahan warna idikator pada *view hardware* mempunyai arti sama terhadap semua *input* dan *output*, warna hijau menunjukkan suatu keadaan aktif (hidup) sedangkan warna merah menunjukkan bahwa *pin* yang bersangkutan dalam kondisi pasif (mati).

Setelah program di jalankan (*debag*), kemudian kita tekan [F7] pada *keyboard* untuk mengamati perubahan yang terjadi secara *step by step* (langkah demi langkah) terhadap program yang dibuat.

### D.1. Hasil Pengujian Perangkat Lunak Mikrokontroler Pertama

Sebagai pengganti *inputan* dari sensor ke mikrokontroler dapat dilakukan dengan cara, c*lick* kiri di kotak link pada *view hardware* untuk memilih salah satu dari empat pilihan, (*No connection, Ground, Vcc,* atau *Net*).

Jika ingin mengaktifkan sensor, maka link pada *view hardware* dipilih *Vcc* dan apabila ingin megembalikan kondisi semula (sensor yang sudah aktif ingin dinonaktifkan) maka link pada *view hardware* dipindah ke *Ground*, begitu seterusnya. Cara pengaturan (link) *view hardware* seperti yang terlihat pada Gambar 4.13.

	😼 Port 1 (PENGEM	A 🔳 🗖 🔀	
S	0 Vcc LATCH 00 Ground No Conr No Conr No cou Groun Vcc Net 7 No Conr	STITU nection nection nd	

Gambar 4.13. Pengaturan View Hardware

Untuk mengetahui fungsi-fungsi dan pemakaian pin-pin mikrokontroloer

pertama tapat dilihat dalam Tabel 4.5. berikut.

	raber 4.5. rengaturan r in Wirklokontroler r ertaina					
Nomor Pin	Fungsi	Keterangan				
P1.0	Photodiode	Sensor Kacang				
P1.1	Photodiode	Sensor Bumbu				
P1.2	Photodiode	Sensor Campuran Kacang dan Bumbu				
P1.3	Photodiode	Sensor Sambal pada Penampungan				
P1.4	Limit switch	Sensor Batas Atas Kalibrasi				

Tabel 4.5. Pengaturan Pin Mikrokontroler Pertama

P1.5	Limit switch	Sensor Batas Bawah Kalibrasi
P0.0	Motor AC	Penggerak Gilingan Kacang dan Bumbu
P0.1	Motor AC	Penggerak Campuran Kacang dan Bumbu
P0.2-P0.3	Motor DC	Penggerak Naik Turun Kalibrasi
P2.0-P2.7	Koneksi	Komunikasi dengan Mikrokontroler Kedua

Seperti yang terlihat dalam tabel *Port* 1 (*pin* P1.0-*pin* P1.5) digunakan sebagai *port input* untuk sensor, Gammbar 4.14 berikut adalah tampilan ProView32 pengaturan *Port* 1 untuk *inputan*.



Gambar 4.14. Keterangan View Hardware Sebagai Inputan

Sedangkan Port 0 (pin P0.0-pin P0.3) digunakan sebagai port output untuk

motor AC dan motor DC. Pengaturannya seperti Gambar 4.15. berikut.



Gambar 4.15. Keterangan View Hardware Sebagai Outputan

Hasil dari pengujian program mikrokontroler pertama yang dilakukan secara *step by step* diuraikan dibawah ini:

• *Pin* P1.0 dan *Pin* P1.1 dihubungkan ke Vcc (sensor kacang dan bumbu aktif). Gambar 4.16. Tampilan ProView32 sensor kacang dan bumbu aktif.

📭 et (program _aso _mikm (prognikm _ok)(produksi.asm
CER SENSOR FACANO : NO SENSOR FACANO, CER SENSOR HUNDH CALL HUTUR_AC_FACAND, ASTIF
CEK_STRIOR_BUMBU : NEB SENSOR_BUMBU, CEK_SENSOR_KALANG
🚨 Code (produksi)
International partness pa
😴 Part 1 (PRODUK 🕞 🔀 😭 Part 0 (PRODUK 📮 🔀 🥌 Main Registers (PRODUKSI) 💦 🛛 🗶 🔛 Part 2 (PRODUK 📮 🗙
Open Number         Vacue         CPU         Bank         Data         Hardware         Open Number         Nu Connection           LATCH/00         No Connection         ACC         0730         RB         00         R0         00         PU         00         Nu Connection           Monomation         No Connection         ACC         0730         RB         00         PU         00         Nu Connection

• Aktifnya sensor kacang dan bumbu mengakibatkan *Pin* P0.0 motor AC aktif. Gambar 4.17. Proses penggilingan kacang dan bumbu.



Gambar 4.17. Penggilingan Kacang dan Bumbu

•

Pin P0.0 akan mati (penggilingan kacang dan bumbu selesai), apabila Pin
P1.0 dan Pin P1.1 mati (tidak ada kacang dan bumbu yang terdeteksi oleh sensor), link Pin P1.0 dan Pin P1.1di ground kan. Keadaan ini ditunjukkan pada Gambar 4.18.



Gambar 4.18. Penggilingan Kacang dan Bumbu Selesai

Matinya *Pin* P1.0 dan *Pin* P1.1 digunakan sebagai logika untuk menghidupkan *Pin* P0.1 (proses penyampuran kacang dan bumbu). Gambar 4.19. Tampilan ProView32 proses penyampuran kacang dan bumbu.

File Edit Search Project Tool View Debug Options Window Help         Image: Search Project Tool View Debug Options Window Help         Image: Search Project Tool View Debug Options Window Help         Image: Search Project Tool View Debug Options Window Help         Image: Search Project Tool View Debug Options Window Help         Image: Search Project Tool View Debug Options Window Help         Image: Search Project Tool View Debug Options Window Help         Image: Search Project Tool View Debug Options Window Help         Image: Search Project Tool View Debug Options Window Help         Image: Search Project Tool View Debug Options Window Help         Image: Search Project Tool View Debug Options Window Help         Image: Search Project Tool View Debug Options Window Help         Image: Search Project Tool View Debug Options Window Help         Image: Search Project Tool View Debug Options Window Help         Image: Search Project Tool View Debug Options Window Help         Image: Search Project All_Nalk:         Image: Search Project All_Nalk:         Image: Search Project All_TORDIN         Image: Search Project All_TORDIN         Image: Search Project All_TORDIN Acc CAD281         Image: Search Project All_TOP Cacce Camputant Option         Image: Search Project All_TOP Cacce Camputant Option         Image: Search Project All_Topic Cacce Project Alll Project Alll Project All Project Alll Project All Project All P	🧟 PV32	
Code (produkci)         Code (produkci)         CLR MOTOR_AC_CAMPURAN         CLR MOTOR_AC_CAMPURAN         CLR MOTOR_CAL_TURIN         CLR MOTOR_CAL_CAMPURAN         CLR MOTOR_CAL_CAMPURAN         CLR MOTOR_CAL_CAMPURAN         CLR MOTOR_CAL_CAMPURAN         CLR MOTOR_CAL_CAMPURAN         CLR MOTOR_CAL_CAMPURAN         CLR MOTOR_CAL_TURIN         CLR MOTOR_CAL_CAL_TURIN         CLR MOTOR_CAC_CAMPURAN         CLR MOTOR_CAC_CAMPURAN         CLR MOTOR_CAC_CAMPURAN         DIG:         MOTOR_CAC_CAMPURAN         DIG:         MOTOR_CAC_CAMPURAN         No Connection	File Edit Search Project Tool View Debug Options Window Help	
Image: Construction of the construc	en >x49 <u>&gt; &gt;</u>	
SETE NOTOR_AC_CAMPURAN SETE NOTOR_AC_CAMPURAN RET NOTOR_CAMPURAN_MATI: CLR NOTOR_AC_CAMPURAN RET NOTOR_CAMPURAN_MATI: CLR NOTOR_CAMPURAN RET NO Connection No Con	🕼 e: \ \produksi.asm	🔚 Port 1 (PRODUK 💶 🗖 🗙
EST     UATCHOOR     Ground       MUTOR, AC_CAMPURAN, MATI:     CLR     NOTOR, AC_CAMPURAN, MATI:       CLR     NOTOR, AC_CAMPURAN, MATI:     No Connection       RET     No Connection     No Connection       No Connection     No Connection     No Connection       No Connection     No Connection     No Connection       CLR     NOTOR_C KAL_TURUN     Image: Code     No Connection       Code (produksi)     Image: Code (produksi)     Image: Code (produksi)     Image: Code (produksi)       Image: Code (produksi)     Image: Code (produksi)     Image: Code (produksi)     Image: Code (produksi)       Image: Code (produksi)     Image: Code (produksi)     Image: Code (produksi)     Image: Code (produksi)       Image: Code (produksi)     Image: Code (produksi)     Image: Code (produksi)     Image: Code (produksi)       Image: Code (produksi)     Image: Code (produksi)     Image: Code (produksi)     Image: Code (produksi)       Image: Code (produksi)     Image: Code (produksi)     Image: Code (produksi)     Image: Code (produksi)       Image: Code (produksi)     Image: Code (produksi)     Image: Code (produksi)     Image: Code (produksi)       Image: Code (produksi)     Image: Code (produksi)     Image: Code (produksi)     Image: Code (produksi)       Image: Code (produksi)     Image: Code (produksi)     Image: Code (prod	SETB MOTOR_AC_CAMPURAN	o E Ground
NOTOR_AC_CAMPURAN_MATI: CLR_NOTOR_AC_CAMPURAN RET NOTOR_CALIBRASI_NAIK: SETE NUTOR_DC_KAL_MAIK CLR_NOTOR_DC_KAL_TURUN CLR_NOTOR_DC_KAL_TURUN CLR_NOTOR_DC_KAL_TURUN CLR_NOTOR_CC_KAL_TURUN CLR_NOTOR_AC_CAMPURAN DIG: MOTOR_AC_CAD281 SETE P0.1 Madress Symbol No Connection No Connection	RET	LATCH 00 Count
RD UD R.C. CARFORAN TATI:     IN Connection       CLR     NOTOR, AC_CAMPURAN       RET     No Connection       MOTOR, KALIBRASI, MAIK:     SETB       SETB     MOTOR, DC, KAL, MAIK       CLR     MOTOR, DC, KAL, TURUN       Code     Mnemonic       RH 26     SETB MOTOR, AC, CAMPURAN       D109:     MOTOR, AC, CAMPURAN       D109:     MOTOR, AC, CAMPURAN       D100:     MOTOR, AC, CAMPURAN       D1010:     CLR MOTOR, AC, CAMPURAN       D1010:     No Connection       No Connection     No Connection <t< td=""><td>HOTOD AC CANDIDAN NATE.</td><td>Exit Childen Composition</td></t<>	HOTOD AC CANDIDAN NATE.	Exit Childen Composition
RET HOTOR_KALIBRASI_NAIK: SETB HOTOR_DC_KAL_NAIK CLR HOTOR_CKAL_TURUN CLR HOTOR_CKAL_TURUN CLR HOTOR_DC_KAL_TURUN CLR HOTOR_CC_KAL_TURUN CLR HOTOR_CC_KAL_TURUN CLR HOTOR_CC_KAL_TURUN CLR HOTOR_CC_CAMPURAN DIG9: MOTOR_AC_CAMPURAN DIG9: MOTOR_AC_CAMPURAN DIG9: MOTOR_AC_CAMPURAN DIG9: MOTOR_AC_CAC261 CLR P0.1 HI 31 RET DIG0: MOTOR_AC_CAMPURAN DIG0: MOTOR_AC_CAC261 CLR P0.1 HI 31 RET DIG0: MOTOR_AC_CAC261 C	NUTUR_AC_CAMPURAN_MATT:	
HOTOR_KALIBRASI_NAIK:     No Connection       SETB     NOTOR_DC_KAL_TURUN       CLR     MOTOR_DC_KAL_TURUN       Code     (Mnemonic)       Code     (Mnemonic)       Code     Mnemonic       RH 205     SeTB       No Connection     No Connection	RET	
MOTOR_KALLERAST_MATK:         SETB       MOTOR_DC_KAL_TURUN         CLR       MOTOR_DC_KAL_TURUN         CLR       MOTOR_DC_KAL_TURUN         Code (produkci)       Image: Code (produkci)         Motor_Ac_CAD281       SETB MOTOR_AC_CAD281         RH 26       SETB MOTOR_AC_CAD281         SETB MOTOR_AC_CAD281       SETB P0.1         Integration       No Connection         No Connection       No Connection         No Connection <t< td=""><td></td><td></td></t<>		
SETB     MOTOR_DC_KAL_MAIK       CLR     MOTOR_DC_KAL_TURUN       CLR     No Connection       Code     Mnemonic       H# 265     SETB       MOTOR_AC_CAD201     SETB       SETE     MOTOR_AC_CAD201       SETE     MOTOR_AC_CAMPURAN       0109:     MOTOR_AC_CAD201       SETE     No Connection       No Connection     No Connection	MOTOR_KALIBRASI_NAIK:	No Lonnection
Code (produksi)  Code (	SETB MOTOR_DC_KAL_NAIK	No Connection
Code (produkci)  Code (mnemonic  R12 26 SETE MOTOR_AC_CAMPURAN  D109: MOTOR_AC_CAD281 SETE P0.1  R130 CLR MOTOR_AC_CAMPURAN D100: MOTOR_AC_CAMPURAN D	CLR MOTOR_DC_KAL_TORON	7 No Connection
Code (preduksi)     Code Mnemonic     Address Symbol Code Mnemonic     O     No Connection     No		-
Address Symbol Code Mnemonic 0 No Connection 177 26 SETE MOTOR_AC_CAMPURAN 0109: MOTOR_AC_CAD281 SETB P0.1 17003: 22 RET 17003: 22 RET 17003: CLR MOTOR_AC_CAMPURAN 010C: MOTOR_AC_CAMPURAN 010C: MOTOR_AC_CAMPURAN 010C: MOTOR_AC_CAMPURAN 010C: MOTOR_AC_CAMPURAN 010C: MOTOR_AC_CAC281 CLR P0.1 1707 PP00 No Connection No Connection	📶 Code (produksi) 📃 🗖 🔀	🕌 Port 0 (PRODUK 🖃 🗖 🗙
## 26     SETE MOTOR AC CAMPURAN       0109:     MOTOR AC CAD281       ## 27 RET     No Connection       1003:     22       ## 30 CLR MOTOR AC CA281     CLR P0.1       1000:     MOTOR AC CA281       0100:     MOTOR AC CA281       1010:     No Connection	Address Symbol Code Mnemonic	8 No Connection
0109:     MOTOR_AC_CAD281     SETB P0.1       ## 27 RET     22 RET       ## 30 CLR MOTOR_AC_CAD281     CLR P0.1       ## 31 RET     CLR P0.1       ## 31 RET     22 RET	##_26 SETB MOTOR_AC_CAMPURAN	LATCH 02 No Connection
Image: Construction     22     RET       010B:     22     RET       Image: Construction     No Connection       010C:     MOTOR_AC_CAC281     CLR P0.1       Image: Construction     No Connection       No Connection     No Connection       No Connection     No Connection       No Connection     No Connection	0109: MOTOR_AC_CAD281 SETB P0.1	No Connection
100:     CLR MOTOR AC CAMPURAN       100:     MOTOR AC CAMPURAN       100:     MOTOR AC CAC281       110:     Connection       110:     No Connection       110:     No Connection	##_27 RET	No Connection
DIOC:         MOTOR_AC_CA281         CLR P0.1           110F:         22         BET         No Connection	## 30 CLB MOTOR AC CAMPURAN	No Connection
## 31 RET No Connection	010C: MOTOR AC CAC281 CLR P0.1	PAD No Connection
010F 22 BET	##_31 RET	No Connection
7 No Connection	010E: 22 RET ⊻	7 No Connection

Gambar 4.19. Proses Penyampuran

•

*Pin* P1.2 adalah sensor penyampuran yang digunakan untuk mendeteksi hasil campuran (sambal) yang keluar dari proses penyampuran menuju ke tempat penampungan. *Pin* P1.2 akan menyala selama sambal masih keluar dari mesin penyampuran, begitu sebaliknya. Ketika *Pin* P1.2 mati maka program akan mematikan *Pin* P0.1 (proses penyampuran selesai).

 Sambal yang berada dalam tempat penampungan akan dideteksi oleh sensor penampungan melalui *Pin* P1.3 (sensor penampungan aktif). Sensor ini sangat penting mengingat proses pengemasan akan terus dilakukan selama *Pin* P1.3 ini masih aktif (masih ada sambal yang perlu dikemas). Gambar 4.20. Sensor penampungan mendeteksi sambal



Gambar 4.20. Sensor Penampungan Aktif

Sebelum dilakukan kalibrasi untuk proses pengisian, *port* 2 mikrokontroler pertama akan mengambil data dari mikrokontroler kedua (selama *port* 2 tidak bernilai 11H) maka proses pengisian tidak akan dilakukan, hal ini berarti kaleng yang berada pada proses pengemasan belum siap, begitu sebaliknya. Jika *port* 2 bernilai 11H maka proses kalibrasi akan dilakukan. Gambar 4.21. menunjukkan proses komunikasi dengan mikrokontroler kedua

🖉 PV32		
File Edit Search Project Tool View Debug Options Window	Help	
2	n 🖸 🖓 🔊 🖉 🖉 🐂 🚽 🚽	—J
C er\\progmikro_ok\produksi.asm		🚰 Port 1 (PRODUK 🔳 🗖 🔀
PROSES_PENCAMPURAN_SELESAI:	<u>^</u>	0 🕻 Ground
CALL MOTOR AC CAMPURAN MATI		LATCH 00 Ground
		Ground
CEK_SENSOR_PENAMPUNGAN:		
UND SENSOR_PENALPUNDAN, 4		No Connection
TUNGGU_KALENG:		No Connection
MOV A, P2 ; ambil data dari m	ikro-2	7 No Connection
KING AND	>	no comection
Main Registers (DRODUKSI)		Port 0 (PRODUK
CPU Bank Data Hardware		
PC 015C RB 00 @R0 00 P0 00	0 No Connection	U No Connection
ACC 00 R0 00 @R1 00 P1 00	LATCH II No Connection	No Connection
PSW 80 R1 00 @DPTR FF P2 11	No Connection	
DPTR0000 B3 00 X@B1 FF P3 00	No Connection	No Connection
B OD R4 OD SPX XX THLUDODO	PADE No Connection	PAD No Connection
C 1 R5 00 XAREA XX THL1 0000	No Connection	No Connection
EA 0 R6 00 Task XX THL2 AAAA	7 No Connection	7 🚺 No Connection
IE DO BZ DO TaskP W PCON DO		

Gambar 4.21. Komunikasi Dengan Mikrokontroler Kedua

*Pin* P0.2 (kalibrasi naik) akan aktif sampai menyentuh sensor atas kalibrasi (*pin* P1.4 aktif) lalu motor berhenti dan kemudian *pin* P0.3 (kalibrasi turun) akan aktif sampai sensor bawah kalibrasi *pin* P1.5 aktif motor berhenti. Gambar 4.22. Proses kalibrasi.

	PV32 File Edit Search Project Tool View Debug Options Window	v Help		
	22 > X II > <u>&gt; .</u>	<u></u>	—J	
	C e:\\progmikro_ok\produksi.asm		👺 Port 1 (PRODUK 🖃 🗖 🔀	
	MOV A, P2 ; ambil data dari : CJNE A, #11H, TUNGGU_KALENG	aikro-2	0 Ground	
	KALIBRASI_NAIK: CALL MOTOR_KALIBRASI_NAIK JNE SENSOR_KAL_ATAS, % ; loop sampai terd	eteksi sensor kal atas	Pen Vec	
	KALIBRASI_TURUN: CALL MOTOR_KALIBRASI_TURUN	=	No Connection	
	JNB SENSOR_KAL_BAWAH, ¢ ; loop sampai terd	eteksi sensor kal baw <mark>a</mark> h 😽 🔀	7 No Connection	
	🜌 Main Registers (PRODUKSI)	🚰 Port 2 (PRODUK 🖃 🗖 🔀	🖫 Port O (PRODUK 🖃 🗖 🔀	
	CPU         Bank         Data         Hardware           PC         0166         RB         00         QR0         00         P0         08	0 No Connection	0 No Connection	
	ACC 11 R0 00 @R1 00 P1 00 PSW 00 R1 00 @DPTR FF P2 11	No Connection	No Connection	
	SP         07         R2         00         X@R0         FF         P3         00           DPTR0000         R3         00         X@R1         FF         TCON         0A	No Connection	No Connection No Connection	
	B 00 R4 00 SPX XX THL00000	PAD No Connection	PAD No Connection	
	EA 0 R6 00 Task X THL2AAAA	7 No Connection	7 No Connection	CNLC
	Combor 1.2	Drosos Kalih		<b>SNIS</b>
	Gainbar 4.2.	2. FIUSES Kallol	as1	
• Setelah	proses kalibrasi diatas	selesai, maka	a proses selan	jutnya adalah
				, , ,
mikroko	ontroler pertama men	girim data 2	22H melalui	port 2 ke

proses ini ditunjukkan pada Gambar 4.23. berikut. BAYA

PV32     Fig. 5th Same Print Tell, View Other Online, Window	0.6		
	Mon 정 전 의 비장 I	1	
💼 e:\\progmikro_ok\produksi.asm		🚰 Port 1 (PRODUK 🖃 🗖 🗙	
CALL MOTOR_KALIBRASI_TURUN JNB SENSOR_KAL_BAWAH, % ; loop sampai terdet	teksi sensor kal bawah	0 Ground	
KALENG_SUDAH_TERISI: CALL MOTOR_KALIBRASI_STOP		Ground Vcc	
KIRIM_DATA_KE_MIKRO2: NOV F2, #22H			
JNB SENSOR_PENAMPUNGAN, KEMBALI_KEPROSES_AWAL			
< III.	>		
Main Registers (PRODUKSI)	👺 Port 2 (PRODUK 🔳 🗖 🔀	🚰 Port 0 (PRODUK 🔳 🗖 🗙	
CPU         Bank         Data         Hadware           PC         0171         R6         00         P0         00           ACC         11         R6         00         P1         00         P3           PSW         00         R1         00         @P1FF         P2         22           SP         07         R2         00         XeN1         FP         200           DPTFM0000         R3         00         XeN1         FF         P3         00           D         P1FM0000         R3         00         XeN1         FF         000           D         FF         500         XeN4         FF         T000         00           D         R4         00         SPX         XX         THL00000         C         0         R5         00         ARA         X         THL2         000         EA         EA         000         EA         000	LATCH 22 No Connection No Connection	0 No Connection LATCHIOO No Connection No Connection No Connection No Connection No Connection No Connection No Connection No Connection No Connection	

mikrokontroler kedua, sebagai informasi bahwa pengisian telah selesai

Gambar 4.23. Proses Pengisian Selesai Mikrokontroler Pertama Mengirim Data 22H melalui *port* 2 ke Mikrokontroler kedua

Setelah proses ini selesai maka program telah berejalan satukali alur proses.
 Selanjutnya program akan mengecek sensor penampungan apakah masih ada sambal yang perlu dikemas bila tidak maka proses akan selesai dan bila masih ada sambal dalam penampungan maka proses akan kembali ke proses yang menunjuk pada Gambar 4.20.

### D.2. Hasil Pengujian Perangkat Lunak Mikrokontroler kedua

Pengujian yang dilakukan pada mikrokontroler kedua ini sama dengan pengujian yang dilakukan pada mikrokontroler pertama. Untuk mengetahui fungsi masing-masing *pin* pada mikrokontroler kedua dapat dilihat pada Tabel 4.6. berikut.

Nomor Pin	Fungsi	Keterangan
P1.0	Photodiode	Sensor Antrian Kaleng
P1.1	Photodiode	Sensor Jarak Antar Kaleng
P1.2	Photodiode	Sensor Counter Kaleng
P1.3	Photodiode	Sensor Isi Kaleng
P1.4	Limit switch	Sensor Tutup Kaleng
P1.5	Limit switch	Sensor Pendorong Tutup Kaleng
P1.6	Limit switch	Sensor Antrian Tutup Kaleng
P1.7	Limit switch	Sensor Hasil
P3.2	Limit switch	Sensor Dorong Kaleng
P0.0-P0.1	Motor DC	Pemutar Antrian Kaleng Kosong
P0.2-P0.3	Motor DC	Pendorong Maju Mundur Kaleng Kosong
P0.4-P0.5	Motor DC	Penggerak Conveyor
P0.6-P0.7	Motor DC	Penggerak Naik Turun Tutup Kaleng
P3.0-P3.1	Motor DC	Pemutar Antrian Tutup Kaleng
P2.0-P2.7	Koneksi	Komunikasi dengan Mikrokontroler Pertama

Tabel 4.6. Pengaturan Pin Mikrokontroler Kedua

Seperti yang terlihat dalam Tabel 4.2. *Port* 1 (*pin* P1.0-*pin* P1.7) dan *Port* 3 (*pin* P3.2)digunakan sebagai *port input* untuk sensor, Gammbar 4.24 berikut adalah tampilan ProView32 pengaturan *Port* 1 dan *Port* 3 untuk *inputan* sensor.



Tampilan view hardware port 0 den port 3 Untuk Output Pert 0 (PENGE ) X No Connection Pin P0.0 Pin P0.1 Perutar Antrian Kaleng Kosong 1 Pin P0.1 Perutar Antrian Kaleng Kosong 1 Perutar Antrian Kaleng Kosong 1	
Port 0 (PENGE	
ATCH/00 No Connection Pin P0.0 Pemutar Antrian Kaleng Kosong 0 Pin P0.1 Pemutar Antrian Kaleng Kosong 1 Pin P0.1 Pemutar Antrian Kaleng Kosong 1	_
LATCH[00 No Connection Pin P0.1 Pemutar Antrian Kaleng Kosong I	
D'- DO O D - M - M - M	1
PU.2 Pendorong Maju Kaleng Kosong	-
Pin P0.3 Pendorong Mundur Kaleng Koson	g
Pin P0.4 Penggerak Conveyor 0	٦
PAD No Connection Pin P0.5 Penggerak Conveyor 1	1
No Connection  Pin P0.6 Penggerak Naik Tutup Kaleng	1
7 No Connection Pin P0.7 Penggerak Turun Tutup Kaleng	1
Port 3 (PENGE 🔳 🗖 🗙 Tabel pengaturan pin mikrokontroler	
0 No Connection - Piri P3.0 Pemutar Antrian Tutup Kaleng	0
LATCH 00 No Connection Pin P3.1 Pemutar Antrian Tutup Kaleng	1
No Connection	
No Connection	
No Connection	
PAD No Connection	
No Connection	
7 F No Recorded	
/ No Connection	

Gambar 4.25. Keterangan View Hardware Untuk Motor

Hasil pengujian program mikrokontroler kedu yang dilakukan secara *step by step* diuraikan sebagai berikut.

• Pada saat mikrokontroler kedua aktif program pertamakali yang dijalankan adalah mengeset *Pin* P0.0 (memutar antrian kaleng), dan akan berhenti sampai *Pin* P1.0 aktif (terdeteksi sensor antrian kaleng). Gambar 4.26. dan Gambar 4.27. berikut ini menunjukkan proses antrian kaleng berputar sampai terdeteksi sensor dan berhenti.



Gambar 4.26. Antrian Kaleng Berputar

🗷 PV32	
File Edit Search Project Tool View Debug Options Window Help	
2	
💼 e:\\pengemasan.asm	👺 Port 1 (PENGE 🔳 🗖 🗙
CLR MOTOR_DC_ANTRIAN_KALENG1 CLR MOTOR_DC_ANTRIAN_KALENG2 PET MOTOR_DORONG_KALENG_NAJU: SETB MOTOR_DC_KALENG_MAJU CLR MOTOR_DC_KALENG_MUNDUR RET	D LATCHIOD Ho Connection No Connection No Connection No Connection No Connection
NOTOR_DORONG_KALENG_HUNDUR:	No Connection 7 No Connection
🛃 Code (pengemasan)	Port O (PENGE 🖃 🗖 🗙
Address         Symbol         Code         Mnemonic           ##         37         CLR MOTOR_DC_ANTRIAN_KALENG1	0 No Connection No Connection

Gambar 4.27. Antrian Kaleng Berhenti

Setelah *Pin* P0.0 (antrian kaleng) mati dan *Pin* P1.0 (sensor antrian kaleng)
aktif, selanjutnya *Pin* P0.2 (dorong kaleng aktif) bergerak maju sampai *Pin*P1.1 (sensor jarak) aktif, kemudian berhenti. Gambar 4.28. proses dorong kaleng ke *conveyor*.

🚅 PV32	
File Edit Search Project Tool View Debug Options Window Help	
2ª > X(() > <u>&gt;</u>	
🚛 e:\ \pengemasan.asm	🔄 Perit 1 (PENGE 🜉 🗖 🗙 🥖
STOP_ANTRIAN KALENG: CALL HOTOR_ANTRIAN_KALENG_MATI DORONG_KALENG_KZ_CONVEYOR: CALL HOTOR_DORONG_KALENG_HAJU JNB SENSOR_JARAK_KALENG, ( ; loop sampai s CEK_SENSOR_JARAK: JB SENSOR_JARAK_KALENG, ( ; loop sampai s (	0 Vec Vec No Connection No Connection
📕 Code (pengemasan)	👺 Port 0 (PENGE 🔳 🗖 🔀
Address Symbol Code Mnemonic	0 No Connection
##_123 JNB SENSOR_JARAK_KALENG, \$ ; loop sampai ser	LATCH 04 No Connection
0166: 3091FD JNB P1.1, 0166	No Connection
1169 CEK SENSOR 2091ED JB P1 1 CEK SENSOR	No Connection
##_129 CALL MOTOR_DORONG_KALENG_MUNDUR	No Connection
016C: PENDORONG_I3112 ACALL MOTOR_DORON	PAD No Connection
##_130 JNB SENSOR_DORONG_KALENG, \$ ; loop sampai s	No Connection
UTDE: JUDZED JNB P3.2, UTDE 💌	7 🕻 No Connection

Gambar 4.28. Dorong kaleng ke *conveyor* 

• *Pin* P1.1 (sensor jarak) aktif *Pin* P0.2 (dorong kaleng maju) mati dan kemudian *Pin* P0.3 (dorong kaleng mundur) bergerak ke belakang sampai

Pin P3.2 (sensor dorong kaleng) aktif, Pin P3.2 aktif menyebabkan Pin P0.3
mati dan Pin P0.4. (conveyor bergerak) aktif. Gambar 4.29. dan Gambar
4.30. berikut adalah proses yang menunjukkan pendorong kembali keposisi semula dan conveyor bergerak membawa kaleng kesatu.

e: \ \pengemasan.asm	Magnet 1 (PENGE 💶 🗖 🔀	🧏 Port 3 (PENGE 🗖 🗖
CEV SENSOD LADAY.	0 🕻 Ground	0 🚺 No Connection
JB SENSOR JARAK KALENG. S : loop sam	LATCH 00 Ground	LATCH 00 No Connection
	Ground	Vcc
PENDORONG_KALENG_MUNDUR:	Ground	No Connection
CALL MOTOR_DORONG_KALENG_MUNDUR	Ground	No Connection
JNB SENSOR_DORONG_KALENG, \$ ; 1000 Sampai	PAD No Connection	PAD No Connection
STOP PENDORONG KALENG:	No Connection	No Connection
CALL MOTOR_DORONG_KALENG_STOP	7 No Connection	7 🖡 No Connection
		•
🚨 Code (pengemasan)	🖌 🔛 Port O (PENGE 💶 🗖 🔀	🔛 Port 2 (PENGE 🔳 🗖
Address Symbol Code Mnemonic	0 No Connection	0 🖥 No Connection
##_130 JNB SENSOR_DORONG_KALENG, \$ ; loop sampai s	LATCH 08 No Connection	LATCH 00 No Connection
016E: 30B2FD JNB P3.2, 016E	No Connection	No Connection
133 CALL MOTOR DORONG KALENG STOP	No Connection	No Connection
## 136 CALL MOTOR CONVEYOR AKTIE	No Connection	No Connection
0173: JALANKAN CON311C ACALL MOTOR CONVE	PAD No Connection	PAD No Connection
##_137 JNB SENSOR_COUNTER_KALENG, \$ ; loop sampai	No Connection	No Connection
		7 E Ma Connection

2 PV32		
File Edit Search Project Tool View Debug Options Window Help		
2 B > X H S > 🔪		AYA
🕼 e: \ \pengemasan.asm	👺 Port 1 (PENGE 🔳 🗖 🔀	👺 Port 3 (PENGE 🔳 🗖 🔀
SETB MOTOR_DC_CONVEYOR1	0 📱 Ground	0 🚺 No Connection
RET	LATCH 00 Ground	LATCH 00 No Connection
	Ground	Vcc
MOTOR_CONVEYOR_MATI:	Ground	No Connection
CLR MOTOR DC CONVEYOR2	Ground	No Connection
RET	PAD No Connection	PAD No Connection
	No Connection	No Connection
MOTOR_DORONG_TUTUP_AKTIF:	7 🔓 No Connection	7 🛛 No Connection
< >>		
🜌 Code (pengemasan) 📃 🗖 🗙	🚰 Port 0 (PENGE 🖃 🗖 🗙	🚰 Port 2 (PENGE 💶 🗖 🔀
Address Symbol Code Mnemonic	0 🚺 No Connection	0 🟅 No Connection
##_57 SETB MOTOR_DC_CONVEYOR1	LATCH 10 No Connection	LATCH 00 No Connection
011C: MOTOR_CONVED284 SETB P0.4	No Connection	No Connection
##_58 CLR MOTOR_DC_CONVEYOR2	No Connection	No Connection
UTTE: U285 ULR PU.5	No Connection	No Connection
0120; 22 RET	PAD No Connection	PAD No Connection
##_62 CLR MOTOR_DC_CONVEYOR1	No Connection	No Connection
0121: MOTOR CONVEC284 CLR P0.4	7 🚺 No Connection	7 🚺 No Connection
	· · · · · · · · · · · · · · · · · · ·	

Gambar 4.30. Pendorong Berhenti dan *Conveyor* Bergerak

Pin P0.4 (conveyor) bergerak sampai Pin P1.2 (sensor counter) aktif dan mati kemudian conveyor berhenti. Setelah berhenti pin P1.2 akan melakukan perhitungan naik pada R6 (register R6) sekaligus melakukan pengecekan apakah R6 sudah sama dengan 4, kalau belum maka proses seperti semula dilakukan lagi sampai nilai dari register (R6 = 4). Gambar 4.31. berrikut menunjukkan tentang proses perhitungan naik jumlah kaleng.

File Edit Search Project Tool View Debug Options Window Help	
en >x44 <u>&gt; x 20</u> ,0,0	
🔹 e:\\pengemasan.asm	😨 Port 1 (PENGE 🗨 🗆 🗙 😨 Port 3 (PENGE 🗨 🗖
STOP_CONVEYOR_UNTUK_COUNTER: CALL MOTOR_CONVEYOR_MATI	0 Ground 0 No Connection LATCH 00 Ground LATCH 00 No Connection
HITUNG_KALENG: INC R6 ; tambahkan ju CJNE R6, #4, AKTIFKAN_ANTRIAN_KALENG =	Ground Ground Ground Ground Ground Ground No Connection No Connection
JALANKAN_CONVEYOR_UNTUK_PENGISIAN: CALL_NOTOR_CONVEYOR_AKTIF	Main Registers (PENGEMASAN)     CPU Bank Data Hardware
🛣 Code (pengemasan)	PC 0181 RB 00 @R0 00 P0 00 ACC 00 R0 00 @R1 00 P1 00
Address Symbol Code Mnemonic ## 146 INC R6 ; tambahkan jumlah kaleng kosong 017D: HITUNG KALENOE INC R6 ## 147 CJNE R6, #4, AKTIFKAN_ANTRIAN_KALENG 017E: BE04DC CJNE R6,#04,AKTIFKAI ## 150 CALL MOTOR CONVEYOR AKTIF 0181: JALANKAN COBILIC ACALL MOTOR CONVE ## 151 JNB SENSOB ISI KAI ENG. 9: Joon sampai sensor	PSW         00         R1         00         cellor HFF         P2         00         ction           SP         07         R2         00         X@R0         FF         P3         00         ction           DPTF[0000         R3         00         X@R1         FF         TCON         0A         ction           B         00         R4         00         SPX         XX         THL0         0000         ction           C         0         R5         00         XAREA         THL1         0000         ction           EA         0         R6         04         Task         XX         THL2         AAAA         ction           IE         00         R7         00         Task         XX         PCON         00         ction

Gambar 4.31. Hitung Register (R6) Sampai Bernilai 4

• Bila R6 sama dengan 4, berarti sudah ada empat buah kaleng yang siap untuk dilakukan pengisian. *Conveyor* bergerak sampai *pin* P1.3 (sensor isi kaleng) aktif, ketika *pin* P1.3 aktif mikrokontroler kedua mengirim data 11H ke mikrokontroler pertama melalui *port* 2 untuk menginformasikan bahwa kaleng siap diisi (MOV P2, #11 H). Setelah kaleng terisi mikrokontroler pertama mengirim data 22H ke mikrolontroler kedua sebagai informasi pengisian telah selesai.

Kemudian pin P0.4 (conveyor) aktif untuk mebawa kaleng yang sudah terisi keproses penutupan kaleng sampai pin P1.4 (sensor tutup kaleng) aktif. Proses-proses tersebut dapat dilihat pada Gambar 4.32. dan Gambar 4.33.



No Connection No Connection No Connection 22 
 SP
 03
 R1
 00
 Corr
 Corr</t No Connection No Connection PAD No Connection PAD No Connection CLR MC No Connection No Connection 7 🖡 No Connection 7 No Connection

Gambar 4.33. Pengisian Selesai dan Conveyor Bergerak

0121:

Pin P1.4 (sensor tutup kaleng) aktif conveyor tetap bergerak sampai pin
 P1.4 mati. Setelah pin P1.4 mati baru conveyor berhenti dan pin P0.6 (dorong tutup kaleng) akan aktif. Gambar 4.34. prooses penutupan kaleng.

			1	
			)	
📕 e: \ \pengemasan.asm		🛛 🔀 🥵 Port 1 (	PENGE 💶 🗖 🔀	🚰 Port 3 (PENGE 🔳 🗖
* SETB MOTOR_DC_TUTUP1		0]	Ground	n I No Connection
CLR MOTOR_DC_TUTUP2		LATCHIO	Ground	LATCH 00 No Connection
RE1			Ground	Ground
MOTOR DORONG TUTUP MATI:			Ground	
CLR MOTOR_DC_TUTUP1			Ground	
CLR MOTOR_DC_TUTUP2		PAD	No Connection	PADE No Connection
* RET			No Connection	No Connection
MOTOR ANTRIAN TUTUP AKTIF:		7	No Connection	7 No Connection
<		>	no connection	
Code (pengemasan)		🗙 🔛 Port 0 (l	PENGE 🔳 🗖 🔀	🔚 Port 2 (PENGE 🔳 🗖
Address Symbol Code	Mnemonic	0	No Connection	0 📱 No Connection
## 67 SETB MOTOR DC TUTUP1		LATCH 40	No Connection	LATCH 22 🖡 No Connection
0126: MOTOR_DOROD286	SETB P0.6		No Connection	No Connection
##_68 CLR MOTOR_DC_TUTUP2	01 0 00 7		No Connection	No Connection
0128: C287	CLR PU.7	+ 99 4	No Connection	No Connection
012A: 22	RET	PAD	No Connection	PAD No Connection
##_72 CLR MOTOR_DC_TUTUP1			No Connection	No Connection
	CLD D0 C			7

Gambar 4.34. Proses Penutupan Kaleng Pertama

Ketika pendorong tutup kaleng mendorong tutup ke kaleng maka *pin* P1.5
(sensor pendorong tutup) akan aktif, aktifnya *Pin* P1.5 menyebabkan *Pin*P3.0 (motor antrian tutup kaleng) akan aktif (berputar 180 °) untuk
menyiapkan tutup yang kedua. Putaran 180 ° ini dihentikan atau diatur oleh
sensor antrian tutup kaleng *Pin* P1.6 (sensor antrian tutup kaleng). Proses
putaran antrian tutup kaleng dapat dilihat pada Gambar 4.35. berikut.

🗶 PV32									
File Edit Search Project Tool View Debug Options Window Help									
212 - XIII - X. 222									
🕼 e:\\pengemasan.asm	🚰 Port 1 (PENGE 💶 🗖 🔀	🔛 Port 3 (PENGE 🖃 🗖 🔀							
SETB HOTOR_DC_ANTELAN_TUTUP1     CLR HOTOR_DC_ANTELAN_TUTUP2      BEN     HOTOR_ANTELAN_TUTUP NATI:     CLR HOTOR_DC_ANTELAN_TUTUP1	0 Ground LATCH 00 Ground Ground Ground Ground Ground	0 No Connection LATCH[0] No Connection Ground No Connection							
CLR HOTOR_DC_ANTRIAN_TUTUP2 RET RESET_KALENG:	Vcc No Connection 7 No Connection	No Connection No Connection 7 No Connection							
Address Symbol Code Mnemonic	Port U (PENGE C X	0 No Connection							
##         77         SETB MOTOR_DC_ANTRIAN_TUTUPI           0130:         MOTOR_ANTRIAD2B0         SETB P3.0           ##         78         CLR MOTOR_DC_ANTRIAN_TUTUP2           0132:         C2B1         CLR P3.1           ##         79         RT           0134:         22         RET           ##         82         CLR MOTOR DC_ANTRIAN_TUTUP1           0135:         MOTOR ANTRIACTUTUP1	LATCHION No Connection No Connection No Connection No Connection No Connection No Connection No Connection No Connection	LATCH 22 No Connection No Connection No Connection No Connection No Connection No Connection No Connection No Connection							

Gambar 4.35. Proses Antrian Tutup Kaleng

Ketika *pin* P1.6 (sensor antrian tutup) aktif maka akan menghentikan *pin*P3.0 (motor antrian tutup kaleng) dan sekaligus akan melakukan perhitungan naik pada register R7 (INC R7) (CJNE R7, # 4) untuk mengetahui berapa kaleng yang sudah dilakukan penutupan. Jika R7 belum sama dengan 4, maka proses akan dilakukan seperti semula (program kembali ke proses yang menghasilkan Gambar 4.34. ) sampai keempat kaleng trtutup semua. Proses perhitungan register R7 dapat dilihat pada Gambar 4.36. berikut.

🖉 PV32								
File Edit Search Project Tool View Debug Options Window Help								
2 B > X ( ( G > ), 2000 II <b>b 6 6 6 6 7 7</b> 7 7 7 7 7 7 7 7 7 7 7 7 7								
🗗 e: \ \pengemasan. asm	🙀 Port 1 (PENGE 💶 🗖 🔀	👺 Port 3 (PENGE 🗖 🗖 🗙						
JALANKAN_ANTRIAN_TUTUP: CALL HOTOR_ANTRIAN_TUTUP_AKTIF JUNG SENSOR_ANTRIAN_TUTUP, { ; loop samp STOP_ANTRIAN_TUTUP; CALL HOTOR_ANTRIAN_TUTUP_HATI HITUNG_KALENG_ISI: INC R7 CONC R7, #4, JALANKAN_CONVEYOK_UNTUK_TUTUP_KALENK	0 Ground Ground Ground Ground Ground Ground Ground Ground Connetion	0 No Connection ATCH[00] No Connection No Connection No Connection No Connection No Connection No Connection No Connection						
📕 Code (pengem 📕 Main Registers (PENGEMASAN) 📃 🗖 🗙	🚰 Port O (PENGE 🔳 🗖 🔀	🔚 Port 2 (PENGE 💶 🗖 🗙						
Address         Symbol         CPU         Bank         Data         Hardware           ## 188         INC R7         PC         [01A9]         RB         [00]         @R10         00         P0         00           01A8:         HITUN         ACC         22         R0         00         @R10         DP1         00           ## 189         CINE         FSW         80         R1         00         @P17         00           ## 189         CINE         FSW         80         R1         00         @P07         100           01A9:         SP         07         R2         100         @P07         R1         00           11A9:         CALL N         DPTF0000         R3         00         X@R1         FF         TCON         0A           01AC:         JALAN         DO         R4         00         SPX         XM20000         AREA         X         THL100000           01AE:         EA         0         R6         104         Task-         X         THL2/AAAA           F         F00         R4         Totak         XX         THL2/AAAAA         F	0 No Connection No Connection	0 No Connection No Connection No Connection No Connection No Connection No Connection No Connection No Connection No Connection No Connection						

### Gambar 2.36. Proses Penghitungan Kaleng Yang Ditutup

Proses terakhir Gambar 4.37. Setelah R7 sama dengan 4, maka selanjutnya keempat buah kaleng (yang sudah terisi sambal dan telah diberi tutup) dibawa *conveyor* untuk dilakukan penghitungan mundur (DEC R7) (CJNE R7,#0) oleh *Pin* P1.7 (sensor hasil) untuk benar-benar memastikan bahwa hasil produksi sama dengan kaleng yang disiapkan pada awal proses.

<b>PV32</b>												
File Edit	Searc	h Proj	ect To	ol Vi	iew	Debug	Optior	ns Wir	ndow	Help		
2	9	*11-1						<b>6</b>	RS	GO	<b>,@</b> ,@	
ert.	.\penį	gemas	an.asn	n	11	N S 1		τU	т	BĮ		
STOP_CONVEYOR_UNTUK_PACKING:												
HITUNG_HASIL_PRODUKSI:												
DEC R7 CJNE R7, #0, JALANKAN_CONVEYOR_UNTUK_PACKING												
	KEMBA	LI_KE	PROSE	.s_au	JAL:	S U	R	A B	Α	ΥΑ	_	
•	LJ	MP S	TART								<b>~</b>	
🜌 Code (pen 🌌 Main Registers (PENGEMASAN)												
Address	CPU		Bank	<	Data		Hardwa	are	_			
## 203	2 DE	PC [	01BA	RB	00	@R0	00	P0	00			
01B6:	HIT	ACC	22	R0	00	@R1	00	P1	00			
## 203	3 CJI	PSW	00	R1	00	@DPTR	FF	P2	22			
01B7:		SP	07	R2	00	X@R0	FF	P3	00			
##_20	6 LJI	DPTR	0000	R3	00	X@R1	FF	TCON	0A			
01BA:	KEI	В	00	R4	00	SPX	$\times$	THLO	0000			
		C	0	R5	00	XAREA	$\times$	THL1	0000			
UTDE.		EA 	0	R6	04	Task	$ \times $	THL2				
		IE	00	R7	00	TaskP	$\times$	PCON	00			

Gambar 4.37. Proses Terakhir Pengecekan Hasil Produksi

