BAB III

RANCANG BANGUN

A. Tujuan Rancang Bangun

Penelitian ini bertujuan untuk dapat merancang *stand* kelistrikan, menentukan komponen yang akan digunakan pada pembuatan alat ini serta penempatan komponen - komponen yang ada didalamnya selain itu juga untuk membangun suatu sistem yang dapat menghidupkan dan memadamkan sistem *starter*, sistem penerangan (lampu utama dan lampu kecil) serta sistem *signal* (lampu *sign* dan *hazard*) dengan menggunakan sensor suara.

B. Waktu dan Tempat Rancang Bangun

Pelaksanaan penelitian ini dilakukan di Laboratorium Teknik Sepeda Motor dan Laboratorium Teknik Elektronika Industri, SMK Negeri 2 Kota Bekasi. Adapun kegiatan ini dilaksanakan pada bulan Juni 2015 sampai dengan selesai.

C. Alat Dan Bahan

Alat dan bahan yang digunakan pada rancang bangun ini adalah:

- 1. Sensor Suara EasyVR
- 2. Mikrokontroler
- 3. DI-Smart AVR 16 System
- 4. LCD Board 2x16

- 5. Relay
- 6. Connector USB to TTL
- 7. Connector Serial to TTL (ISP)
- 8. Laptop ASUS A46CB
- 9. *Stand* kelistrikan sistem penerangan (Lampu Utama dan Lampu Kecil) serta sistem signal (Lampu *Sign* dan *Hazard*)

10. Stand Engine Trainer

D. Metode Penelitian

Metode penelitian yang digunakan pada penelitian adalah metode eksperimen. Dimana metode eksperimen adalah melakukan suatu percobaan dengan dilakukannya pembuatan suatu alat yang dapat mengolah suara untuk menstarter mesin dan untuk menghidupkan peralatan kendaraan (Lampu Utama, Lampu Senja, Lampu *Sign* Kanan, Lampu *Sign* Kiri dan Lampu *Hazard*). Jalannya penelitian seperti digambarkan pada alur penelitian di bawah ini:



Gambar 3.1 Flowchart penelitian



Gambar 3.2 Flowchart Cara Kerja Alat Eksperimen

E. Perancangan Desain Sensor Suara

Blok diagram keseluruhan sistem yang dirancang dibagi menjadi beberapa bagian, yaitu blok modul sensor suara *EasyVR*, blok mikrokontroler utama, blok *LCD Board Display* dan blok Relay yang ditunjukkan dalam gambar 3.3.



Gambar 3.3 Diagram Blok Sistem

Fungsi masing - masing blok dalam gambar 3.3 adalah sebagai berikut :

1.) Blok Microphone

Terdapat *microphone* sebagai penangkap atau penerima sinyal suara yang selanjutnya akan diteruskan ke *EasyVR* Modul.

2.) Blok Modul Sensor Suara EasyVR

Pengendalian dilakukan melalui suara yang dibantu dengan modul *EasyVR* yang berfungsi sebagai sensor suara. Data – data suara akan di *Training* melalui komputer / laptop menggunakan aplikasi *EasyVR Commander ver.3.9.1*. Data – data yang sudah *disampling* akan dimasukkan kembali ke *database EasyVR*.

3.) Blok Mikrokontroler

Mikrokontroler yang digunakan adalah ATMega 8535 yang berfungsi untuk mengolah data dari modul *EasyVR* dan mengakses relay untuk mengendalikan pengontrol peralatan kendaraan (sistem starter, sistem penerangan dan sistem signal).

4.) Blok LCD Board

LCD Board yang digunakan yaitu sebagai variasi atau pemanis pada rancang bangun alat ini selain itu dapat berguna sebagai konfirmasi pengucapan.

5.) Blok Relay

Relay digunakan sebagai saklar elektrik yang akan dilakukan percobaan terhadap 3 sistem pada kendaraan, yaitu pada sistem *starter*, sistem penerangan dan sistem *signal*.

F. Tahapan Pembuatan Alat

a. Memprogram Sensor Suara EasyVR

Tahapan awal dalam pekerjaan ini adalah men – *training* sensor suara *EasyVR*, pada proses *training* bertujuan untuk mengambil *sample*

suara yang akan disimpan didalam modul *EasyVR*. Pengambilan *sample* suara dilakukan dengan cara menghubungkan sensor *EasyVR* ke Laptop ASUS A46CB melalui *connector USB to TTL* serta dengan *software* yaitu *EasyVR Commander*.



Gambar 3.4 Tampilan Utama Software EasyVR Commander

EasyVR Commander sendiri merupakan *software original* yang telah disediakan dari *EasyVR* yang dapat di *download* dengan gratis pada website resmi.²⁸

Setelah masuk kedalam *software EasyVR Commander* Langkah selanjutnya dalam pemrograman sensor ini adalah dengan cara menambahkan perintah pada sensor suara *EasyVR* (*Add Command*). *Input* perintah suara sebanyak yang kita butuhkan seperti gambar 3.5.

²⁸ <u>http://www.veear.eu/downloads/</u> (diakses pada tanggal 18 mei 2015)

💰 Ea	syVR (Commander ·	v3.9.1		10		_						
File	Edit	Tools H	elp										
22	8 0	OM23	- 😤 💁	\mathbf{S}	?	R 💀 😪	🖉 🛞 🖣	s 🖪 🤞	J. 😲 🗜	- 👰 👻 🖥			
Group	o List				Add Cor	nmand (Shi	ft+INS)						
	Index	Description	Commands	*	Inde	x Label				1	Trained	Conflict	
Ģ	0	Trigger	0		\bigcirc	0					0	OK	
P	1	Group	12										
P	2	Group	1										
Þ	3	Group	0										
P	4	Group	0										
P	5	Group	0										
9	6	Group	0										
9	7	Group	0										
	8	Group	0	=									
	9	Group	0										
	10	Group	0										
9	11	Group	0										
9	12	Group	0										
2	13	Group	0										
2	14	Group	0										
2	15	Group	0										
	16	Password	0										
		SoundTa	1										
12	1	Wordset	8										
2	2	Wordset	0	Ŧ									
Ready							Connected	l to EasyVI	R 3 (Rev 0) o	on COM23			

Gambar 3.5 Menambahkan Perintah (Add Command)

Setelah menambahkan perintah (*Add Command*) kemudian *recording* atau memberikan *Input* suara (*Train Command*) dengan cara meng klik label yang sebelumnya kita buat kemudian klik tombol *Train Command* seperti gambar 3.6.

💰 Ei	asyVR C	ommander -	v3.9.1			-	1.2					×
File	Edit	Tools H	elp									
2	8 [c	OM23	- 😤 🚳	Ģ	9	2 🔊	ھ ا 🔊	🤹 i 🔊 i 🔊) 🖑 🛃 🧔 🖣			
Grou	p List				Grou	up 2 Coi	mm Train Con	mand (ENTER)				
	Index	Description	Commands	*		Index	Label			Trained	Conflict	
Þ	0	Trigger	0			0	ENGINE_STAF	۲T		0	ОК	
Ģ.	1	Group	12		∇		ENGINE_OFF			U	OK	
÷.	2	Group	12			2	HEAD_LAMP			0	OK	
Q.	3	Group	0			3	HEAD_LAMP_	DFF		0	OK	
Ģ.	4	Group	0			4	LAMPU_KECIL			0	OK	
Ģ.	5	Group	0			5	LAMPU_KECIL	_OFF		0	OK	
Ģ.	6	Group	0			6	SIGN_KANAN			0	OK	
	7	Group	0			7	SIGN_KANAN	OFF		0	OK	
Q.	8	Group	0	-		8	SIGN_KIRI			0	OK	
Ģ.	9	Group	0	-		9	SIGN_KIRI_OF	F		0	OK	
¢.	10	Group	0			10	HAZARD			0	OK	
÷.	11	Group	0			11	HAZARD_OFF			0	OK	
÷.	12	Group	0		1.		-					
÷.	13	Group	0									
Ó.	14	Group	0									
Ó.	15	Group	0									
÷.	16	Password	0									
6		SoundTa	1									
é	1	Wordset	8									
(2	Wordset	6	-								
Read	y						Co	onnected to EasyVF	3 (Rev 0) on COM2	3		

Gambar 3.6 *Recording/*memberikan input suara (*Train Command*)

Dalam proses *recording* akan memerlukan 2 kali penginputan suara, yang pertama sebagai suara pertama dan disimpan sebagai suara utama dan yang kedua merupakan konfirmasi terhadap suara yang pertama kali di input sebelumnya seperti gambar 3.7 dan 3.8.

File	Edit	Tools H	elp			
3	800	OM23	- 🛛 🗞	Ģ	ءَ الجا 🕲 🛯 🧔 🔊 😓 🖳 💭 🖓 🗐	•
Grou	p List				Group 2 Command List	
	Index	Description	Commands	-	Index Label	Trained Conflict
Ģ	0	Trigger	0		0 ENGINE_START	Оок
Ģ	1	Group	12			🗸 ок
	2	Group	12		Command Training	0 OK
Ç,	3	Group	0			0 OK
Ģ	4	Group	0		Command 0 of Group 2	0 OK
Ç,	5	Group	0		Regional Start - Phase 1/2	0 OK
Ç,	6	Group	0			0 OK
Ç,	7	Group	0		Press the "Phase 1" button and say the word	0 OK
Ģ.	8	Group	0	=	corresponding to the command "ENGINE_START" within	0 OK
ø	9	Group	0		5 seconds.	0 OK
ø	10	Group	0			0 OK
Ģ	11	Group	0		Phase 1 Cancel	0 OK
Ģ	12	Group	0			
Ģ	13	Group	0			
P	14	Group	0			
Ģ	15	Group	0			
P	16	Password	0			
Ģ		SoundTa	1			
Ģ	1	Wordset	8			
P	2	Wordset	6	-		

Gambar 3.7 Trainning Command Phase 1

ile	Edit	Tools H	elp							
ξ.	8 0	OM23	- 🖹 🌆	9	$\left \begin{array}{c} \mathbf{Q} \\ \mathbf{Q} \end{array} \right $	🕺 🗟 🧟	🖉 👒 😽 🔊	4 🖑 🛵 🖗	-	_
ou	p List				Group 2	Command Li	st			
	Index	Description	Commands	-	Ind	ex Label			Trained C	onflict
	0	Trigger	0		\bigcirc	0 ENGINE_	START		(1)	OK
6	1	Group	12		0	4 ENGINE	OFF.		0	OK
6	2	Group	12		Comman	d Training	-		0	OK
6	3	Group	0						0	OK
6	4	Group	0			Command 0	of Group 2		0	OK
6	5	Group	0		503	"ENGINE_STA	RT" - Phase 2/2		0	OK
6	6	Group	0						0	OK
6	7	Group	0		Press th	e "Phase 2" bu	itton and say the word		0	OK
	8	Group	0		correspo	nding to the o	command "ENGINE_STAF	tT" within	0	OK
6	9	Group	0	-	5 secon	ls.			0	OK
6	10	Group	0			-			0	OK
6	11	Group	0				Phase 2	Cancel	0	OK
6	12	Group	0							
6	13	Group	0			_				
6	14	Group	0							
6	15	Group	0							
١.	16	Password	0							
١.		SoundTa	1							
	1	Wordset	8							
h	2	Wordset	6	-						

Gambar 3.8 Trainning Command Phase 2

Suara yang sudah terdaftar akan masuk ke *database* sensor, *Trained 2* dan *Conflict OK*, maksud dari pernyataan tersebut adalah *Input* Suara sudah terlatih 2 kali dan tidak ada konflik atau suara yang sama antara perintah 1 dengan perintah yang lainnya. Lakukan proses *trainning command* ini ke semua label yang sebelumnya kita buat (*Add Command*) hingga selesai. Seluruh *Trainning Command* yang berhasil yaitu seluruh label *Trained 2* dan *Conflict OK* Untuk Lebih jelasnya dapat dilihat pada gambar 3.9 dan 3.10.

💰 Ea	asyVR C	Commander	- v3.9.1						
File	Edit	Tools H	elp						
2.	8 C	OM23	- 😤 🍕	,	,	2 🔦	👰 🛃 🐉 🚳 🚳 🕼 🕵 🕵	•	
Group List Group 2 Command List									
	Index	Description	Commands	-		Index	Label	Trained	Conflict
P	0	Trigger	0			0	ENGINE_START	2	OK
P	1	Group	12			1	ENGINE_OFF	U	OK
	2	Group	12		$\left \bigcirc \right $	2	HEAD_LAMP	0	OK
9	3	Group	0			3	HEAD_LAMP_OFF	0	OK
9	4	Group	0			4	LAMPU_KECIL	0	OK
9	5	Group	0			5	LAMPU_KECIL_OFF	0	OK
9	6	Group	0		$\left \right\rangle$	6	SIGN_KANAN	0	OK
Þ	7	Group	0			7	SIGN_KANAN_OFF	0	OK
Þ	8	Group	0	=		8	SIGN_KIRI	0	OK
Þ	9	Group	0			9	SIGN_KIRI_OFF	0	ОК
Þ	10	Group	0		$\left \right\rangle$	10	HAZARD	0	OK
Þ	11	Group	0			11	HAZARD_OFF	0	OK
9	12	Group	0						
9	13	Group	0						
9	14	Group	0						
2	15	Group	0						
	16	Password	0						
2		SoundTa	1						
2	1	Wordset	8						
2	2	Wordset	6	Ŧ					
Ready	/						Connected to EasyVR 3 (Rev 0) on COM	123	

Gambar 3.9 Train Command Suara Terdaftar



Gambar 3.10 Train Command Seluruh Perintah Suara terdaftar

Sample suara yang akan digunakan adalah sebanyak sebelas kata dengan perintah yang berbeda. Pengambilan sample suara dilakukan sebanyak dua kali dengan kondisi ideal atau tidak adanya noise, variasi kata dan intonasi disetiap pengucapan relatif sama menghasilkan kesuksesan tinggi dalam pengambilan sample dengan tidak adanya error. Kegagalan menerima variasi suara kedua dengan variasi pengucapan suara pertama akan menimbulkan kegagalan. Oleh karena itu, agar EasyVR dapat berfungsi dengan baik, dibutuhkan variasi suara yang relatif sama dengan sample.

b. Memprogram Mikrokontroler

Tahapan proses yang terdapat pada sistem ini meliputi proses pengolahan data dari modul *EasyVR* ke Mikrokontroler dan proses pengontrolan *relay*. Semua proses tersebut dilakukan oleh perangkat lunak yang terdapat dalam mikrokontroler. Perangkat lunak ini tersusun dari instruksi-instruksi yang membentuk sebuah *listing* program atau *source code*. Semua instruksi program disusun secara terstruktur dalam beberapa subrutin yang secara khusus menangani fungsi tertentu. Dalam pemrograman mikrokontroler dibuat menggunakan *software Code Vision AVR* (CV AVR). Menggunakan bahasa pemrograman yaitu bahasa pemrograman C.

AK CodeVisionAVK	Close
To ca - Li a fan fan fan fan fan Lo a li b b c / b b c / b b c / y li o + o + o o l b b c / b li	
inger and inger a construction of a solution of	
	🕅 Ra Circho 👷 🗷 🕅
Image: Construction of the second s	
	♥ 卒 🛛

Gambar 3.11 Tampilan Utama Software CVAVR

CV AVR merupakan salah satu dari beberapa *software* yang berfungsi sebagai pemprograman mikrokontroler. Dalam proses pekerjaan ini yang dilakukan yaitu memprogram penulisan pada *LCD Board* dan mengatur ke aktifan relay dengan perintah serta memberikan jeda waktu terhadap kebutuhan relay tertentu.

D:\S	crip	t Sweet\RANCANG BANGUN PENGONTROL PERALATAN KENDARAAN DEI	NGAN SENSOR SUARA\all about this script\Aplikasi_Pemrograman\PROGRAM\2.c
	ote	s/ 2.c 🔀	
37		unsigned char v:	
38		putchar('');	
39		y=getchar();	
40		if(y=='A'){lcd_gotoxy(0,1);lcd_putsf("ENGINE START	"); PORTC.1=1; delay_ms(1000); PORTC.0=1; delay_ms(3000); PORTC.0=0; }
41		if(y=='B'){lcd_gotoxy(0,1);lcd_putsf("ENGINE OFF	");PORTC.1=0;} //Relay 2 OFF KONTAK MESIN
42		if(y=='C'){lcd_gotoxy(0,1);lcd_putsf("HEAD_LAMP_ON	");PORTC.2=1;} //Relay 3 ON
43		if(y=='D'){lcd_gotoxy(0,1);lcd_putsf("HEAD_LAMP_OFF	");PORTC.2=0;} //Relay 3 OFF
44		if(y=='E'){lcd_gotoxy(0,1);lcd_putsf("LAMPU_KECIL	");PORTC.3=1;} //Relay 4 ON
45		if(y=='F'){lcd_gotoxy(0,1);lcd_putsf("LAMPU KECIL OFF	");PORTC.3=0;} //Relay 4 OFF
46		if(y=='G'){lcd_gotoxy(0,1);lcd_putsf("SIGN_KANAN_ON	");PORTC.4=1;PORTC.5=0;} //Relay 5 ON
47		if(y=='H'){lcd_gotoxy(0,1);lcd_putsf("SIGN_KANAN_OFF	");PORTC.4=0;} //Relay 5 OFF
48		if(y=='I'){lcd_gotoxy(0,1);lcd_putsf("SIGN_KIRI_ON	");PORTC.5=1;PORTC.4=0;} //Relay 6 OFF
49		if (y=='J') {lcd gotoxy(0,1); lcd putsf("SIGN KIRI OFF	");PORTC.5=0;} //Relay 6 OFF
50		if(y=='K'){lcd_gotoxy(0,1);lcd_putsf("HAZARD	");PORTC.4=1;PORTC.5=1;} //Relay 5,6 HAZARD
51		if(y=='L'){lcd_gotoxy(0,1);lcd_putsf("HAZARD OFF	");PORTC.4=0;PORTC.5=0;} //Relay 5,6 HAZARD
52			
53		return;	
54	۱L	}	
55			
56	P	void voice_start()	
57	申	{	
58		unsigned char x;	
59		<pre>lcd_gotoxy(0,1);lcd_putsf("SPEAK NOW!! ");</pre>	
60		<pre>putchar('d');</pre>	
61		<pre>putchar('B');</pre>	
62		x=getchar();	
63		if (x=='r') {lcd_gotoxy(0,1); lcd_putsr("command ok	");delay_ms(1000);action();}
64		In (x e) (icu_gotoxy(0,i);icd_putsi("command error),return,}
66		3	-
1	<u> </u>	12	•

Gambar 3.12 Program pada Mikrokontroler

Program pada mikrokontroler seperti pada gambar diatas dapat dijelaskan sebagai contoh bahwa jika simbol A hidup maka LCD akan hidup dan menunjukkan tulisan *ENGINE START* kemudian Relay 2 sebagai kontak (ACC) hidup lalu 1 detik kemudian Relay 1 yang menghubungkan ke *starter* akan hidup dan dalam waktu 3 detik Relay 1 akan mati dan Relay 2 akan hidup hingga *command* selanjutnya di perintahkan. Jika simbol B hidup maka LCD akan hidup dan menunjukkan tulisan *ENGINE OFF* kemudian Relay 2 sebagai kontak (ACC) akan OFF. Jika simbol C hidup maka LCD akan hidup dan menunjukkan tulisan *HEAD LAMP ON* kemudian Relay 3 sebagai kontak ke lampu utama akan *ON*. Jika simbol D hidup maka LCD akan hidup 3 sebagai kontak ke lampu utama akan *OFF*. Jika simbol E hidup maka LCD akan LCD akan hidup dan menunjukkan tulisan *HEAD LAMP OFF* kemudian Relay 3 sebagai kontak ke lampu utama akan *OFF*. Jika simbol E hidup maka LCD akan hidup dan menunjukkan tulisan *HEAD LAMP OFF* kemudian Relay 3 sebagai kontak ke lampu utama akan *OFF*. Jika simbol E hidup maka LCD akan hidup dan menunjukkan tulisan *HEAD LAMP OFF* kemudian Relay 3 sebagai kontak ke lampu utama akan *OFF*. Jika simbol E hidup maka

Relay 4 sebagai kontak ke lampu kecil akan ON. Jika simbol F hidup maka LCD akan hidup dan menunjukkan tulisan LAMPU KECIL OFF kemudian Relay 4 sebagai kontak ke lampu kecil akan OFF. Jika simbol G hidup maka LCD akan hidup dan menunjukkan tulisan SIGN KANAN ON kemudian Relay 5 sebagai kontak ke sign kanan akan ON. Jika simbol H hidup maka LCD akan hidup dan menunjukkan tulisan SIGN KANAN OFF kemudian Relay 5 sebagai kontak ke sign kanan akan OFF. Jika simbol I hidup maka LCD akan hidup dan menunjukkan tulisan SIGN KIRI ON kemudian Relay 6 sebagai kontak ke sign kiri akan ON. Jika simbol J hidup maka LCD akan hidup dan menunjukkan tulisan SIGN KIRI OFF kemudian Relay 6 sebagai kontak ke sign kiri akan OFF. Jika simbol K hidup maka LCD akan hidup dan menunjukkan tulisan HAZARD kemudian Relay 5 & 6 sebagai kontak ke hazard akan ON. Jika simbol L hidup maka LCD akan hidup dan menunjukkan tulisan HAZARD OFF kemudian Relay 5 & 6 sebagai kontak ke hazard akan OFF.

Berikut merupakan tabel yang berisikan perintah suara, simbol pada mikrokontroler serta relay yang digunakan :

NO	PERINTAH	SIMBOL PADA	RELAY
NO.	SUARA	MIKROKONTROLER	AKTIF
1	ENGINE START	А	1
2	ENGINE OFF	В	2
3	HEAD LAMP	С	3
4	HEAD LAMP OFF	D	3
5	LAMPU KECIL	Е	4
6	LAMPU KECIL OFF	F	4
7	SIGN KANAN	G	5
8	SIGN KANAN OFF	Н	5
9	SIGN KIRI	Ι	6
10	SIGN KIRI OFF	J	6
11	HAZARD	K	5&6
12	HAZARD OFF	L	5&6

Tabel 3.1. Perintah Suara, Simbol dan Relay

c. Memprogram LCD

Alphanumeric LCD merupakan sebuah Liquid Crystal Display yang terdiri dari segmen – segmen yang berfungsi untuk menampilkan karakter, baik berupa angka, simbol maupun huruf. Display yang akan dibahas adalah display LCD 2 x 16, artinya LCD memiliki 2 baris dan 16 karakter. Berikut merupakan cara pemrograman LCD 2 x 16 :

- 1. Buka Code Vision AVR
- 2. Buatlah *project* baru dengan cara berikut

Pilih File – New – Project – Ok – Code Wizard Yes

 Lakukan pengaturan Alphanumeric LCD, pastikan character/line adalah 16 karena dalam pembuatan alat ini menggunakan LCD 2 x 16 untuk lebih jelasnya dapat dilihat pada gambar dibawah 3.13 dibawah ini.

Eile Program Edit Help									
Di Coo R 🙀 <table-row> 🍪 🗈 🖻 📔</table-row>									
USART Analog Comparator ADC SPI									
USART Analog Comparator ADC SPI									
I2C 1 Wire TWI (I2C)									
Bit-Banged Project Information									
Chip Ports External IRQ Timers									
Alphanumeric LCD									
Enable Alphanumeric LCD Support									
Characters/Line: 16 -									
RS PORTC V Bit: 0 V									
RD PORTC V Bit 1 V									
EN PORTC V Bit: 2 V									
D4 PORTC V Bit: 4 V									
D5 PORTC V Bit: 5 V									
D6 PORTC V Bit: 6 V									
D7 PORTC ▼ Bit: 7 ▼									

Gambar 3.13 Pengaturan awal Alphanumeric LCD

4. Simpan project pada direktori yang diinginkan.

Pilih File – generate save and exit

Dalam penyimpanan ini akan menyimpan 3 file berbeda agar lebih mempermudah simpan semua *file* dengan nama yang sama.

-0-	∯ Co	odeWiza	rdAVR	- untitled.cwp							
	<u>F</u> ile	ile <u>Program</u> <u>E</u> dit <u>H</u> elp									
1	<u>B</u> (🛐 🔯 Pre <u>v</u> iew									
ľ		🔯 <u>G</u> er	am Preview								
1	05/	Ger Ger									
l		Bit-Bange	ed	Project Inforr							
l	Ch	ip F	orts	External IRQ	Timers						

Gambar 3.14 Penyimpanan File

Save C Compi	er Source File	🕵 Save C Compi	ler Project File		Save D:\Script	Sweet\RANCAN	IG BANGUN PENGONTROL PERAL	ATAN KENDARAAN E	DEN
Save in:	PROGRAM	Save in:	PROGRAM		Save in:	DROGRAM	•	G 🤌 📂 🛄 -	
Recent Places	Name Eve Linker Jitit Obj 1 2 Program LCD	Recent Places	Name Exe Linker List Obj 1 2		Recent Places	Name Exe Linker List Obj 1 2	~	Date modified 16/09/2015 23:24 16/09/2015 23:24 16/09/2015 23:24 16/09/2015 23:24 10/09/2015 23:24 10/09/2015 11:21	Type File folder File folder File folder File folder CodeWiza
Network	File name: Program LCD C Save as type: C Compiler files (*.c)	Network	 ✓ File name: Save as type: 	III Program LCD Project files (*.prj)	Network	File name: Save as type:	III Program LCD CodeWizardAVR project files (*.cw	• p) •	Save Cancel

Gambar 3.15 Penyimpanan 3 File

5. Ketikkan program berikut



Gambar 3.16 Pemrograman LCD menggunakan CV AVR

6. Lakukan kompilasi

Pilih Project – build all

Jika terjadi kesalahan penulisan program maka *software* akan menampilkan pesan *error*. Perbaiki program tersebut.

- 7. Simpan project
- d. Pembuatan Trainer sistem penerangan dan sistem signal

Trainer ini berfungsi sebagai dudukan atau tempat untuk seluruh rangkaian baik sumber arus (baterai), sensor suara, mikrokontroler, relay serta *LCD Board*. Selain itu *output* atau beban seperti lampu utama, lampu kecil, serta lampu *sign* juga ada di trainer ini. Dalam pembuatan *trainer* ini menggunakan bahan besi siku dengan ketebalan 3mm dan akrilik dengan ukuran 35 cm x 20 cm dengan ketebalan 5 mm. Berikut merupakan langkah kerja dari pembuatan *trainer* kelistrikan sistem *signal* dan sistem penerangan :

> Plat siku yang akan digunakan sebagai kerangka di potong sesuai dengan kebutuhan yaitu dudukan bawah 35cm x 20cm, dengan ketinggian 100cm dan *holder* akrilik 35cm x 20cm dengan ketebalan plat siku 3mm, setelah dipotong kemudian di sambungkan dengan cara dilas. Setelah itu dilanjutkan dengan pemasangan roda pada setiap sisi dudukan.



Gambar 3.17 Besi Siku yang Sudah di Potong dan di Las

- Melakukan penggerindaan di setiap bagian yang telah dilas. Penggerindaan paling banyak dilakukan pada bagian sudut kerangka. Hal ini dilakukan agar bagian hasil pengelasan terlihat lebih rapi.
- Melakukan proses pendempulan pada setiap bagian yang telah digerinda.



Gambar 3.18 Stand Kelistrikan Setelah di Las

- 5. Melakukan proses pengecatan dasar trainer kelistrikan.
- 6. Mengecat dan melakukan *finishing* pada *trainer* kelistrikan.
- 7. Memasang lampu utama dan lampu kecil pada trainer.
- 8. Memasang lampu sign kanan dan kiri pada trainer.
- 9. Memasang akrilik pada trainer.
- 10. Memasang sensor suara pada akrilik.
- 11. Memasang mikrokontroler pada DI-Smart AVR 16 System.
- 12. Memasang DI-Smart AVR 16 System pada akrilik.
- 13. Memasang LCD board pada akrilik.
- 14. Memasang *relay board* pada akrilik.
- 15. Merangkai seluruh komponen semua sistem.