

Lampiran 1 Gambar dan Deskripsi Alat

Alat Penguji Kabel LAN Berbasis Mikrokontroler

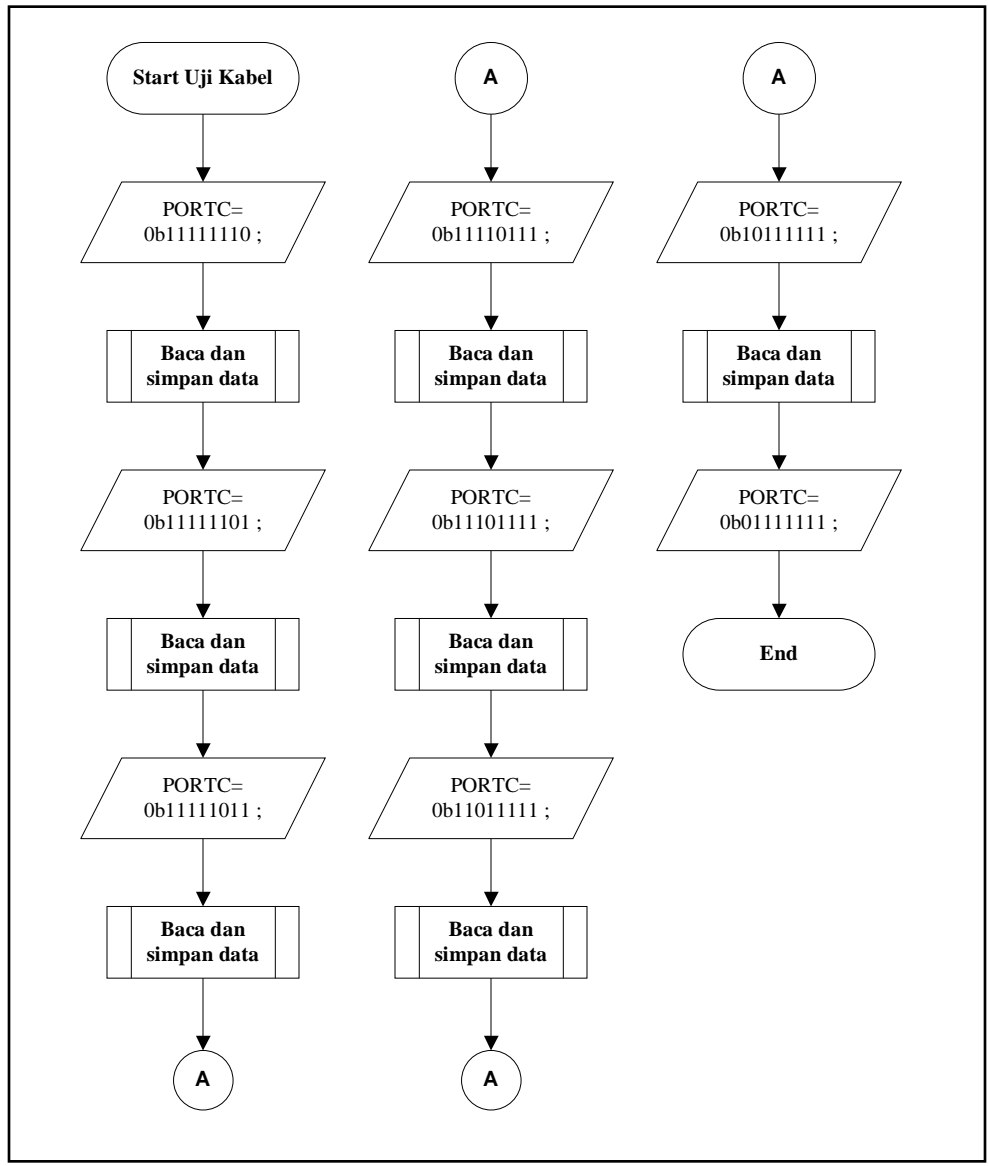
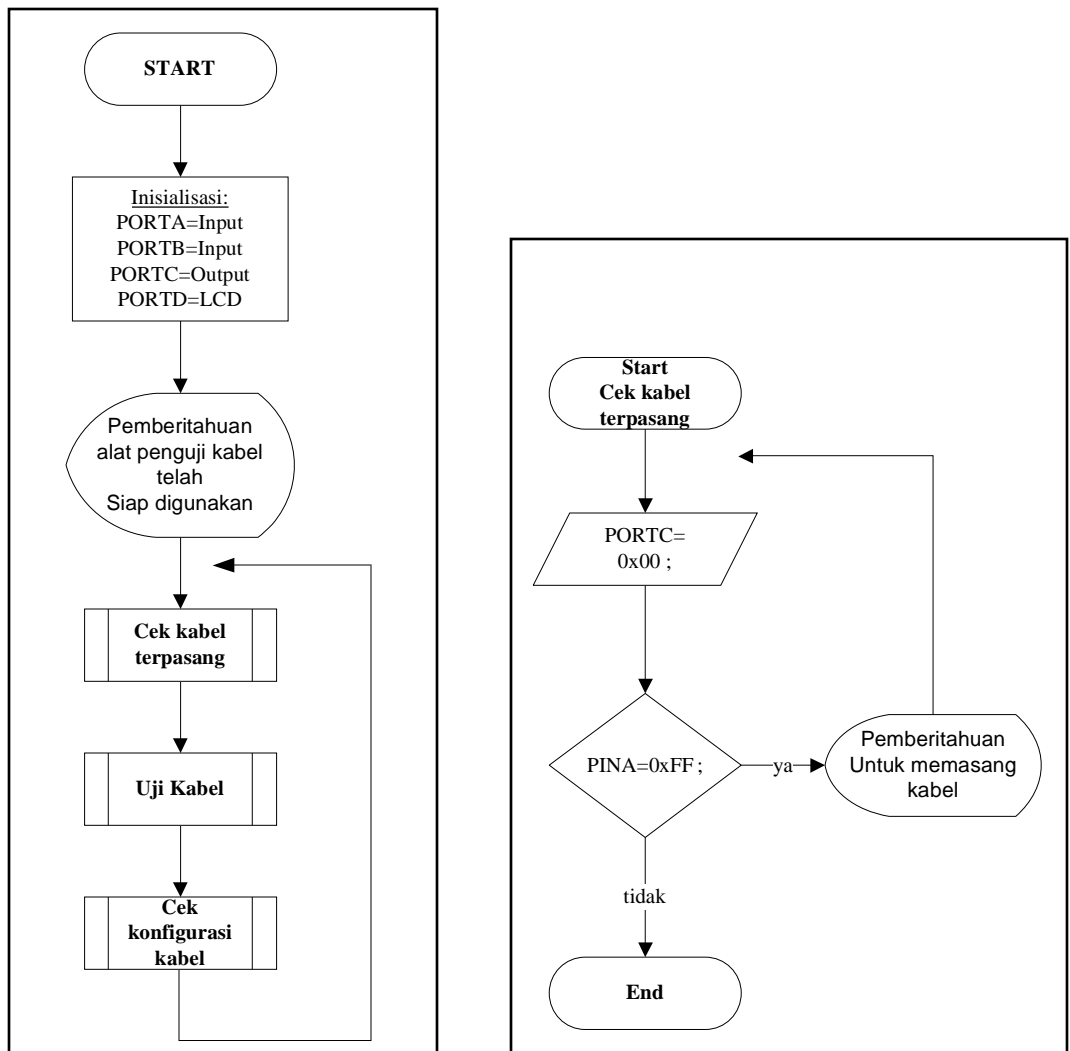


Gambar alat penguji kabel LAN Berbasis Mikrokontroler

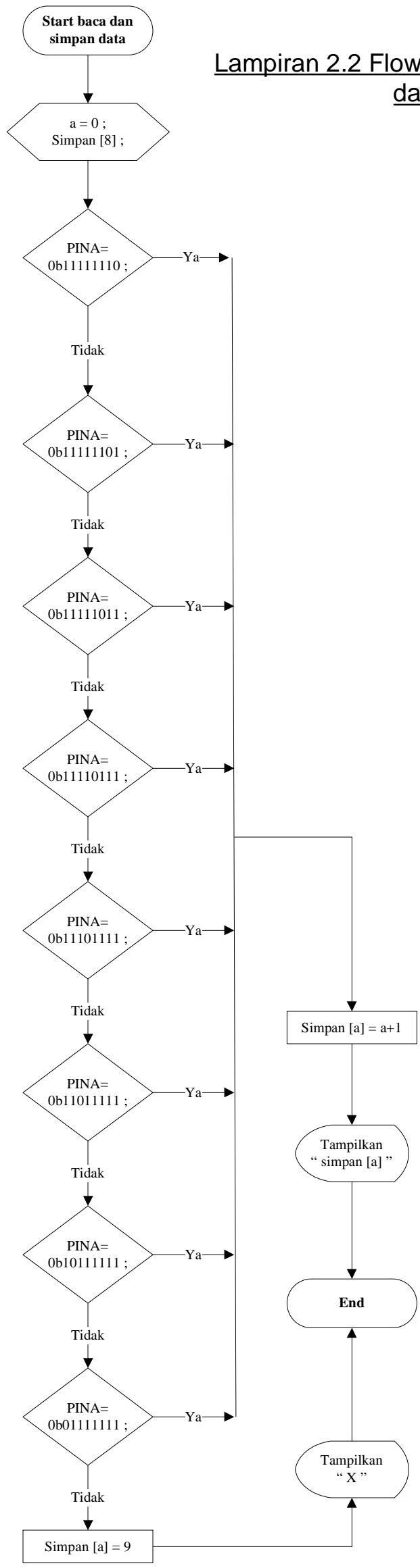
Alat penguji kabel LAN berbasis mikrokontroler dapat mengetahui urutan kabel UTP yang terpasang pada pin konektor RJ45 *male*. Mampu membedakan konfigurasi kabel straight, cross, dan rollover. Menggunakan tampilan LCD 16 kolom 2 baris, dengan latar belakang lampu berwarna biru gelap yang berfungsi sebagai antarmuka bagi pengguna.

Alat penguji kabel LAN menggunakan mikrokontroler sebagai komponen utama yang bertugas sebagai pengirim data dan pembaca data serta mengelola data untuk kemudian ditampilkan pada LCD 16x2. Dibuat untuk mengetahui konfigurasi kabel serta sambungan kabel pada pin-pin konektor RJ45. Alat penguji kabel LAN terdiri dari satu buah modul mikroAVR yang telah terpasang mikrokontroler ATmega8535, dua buah konektor RJ45 *female*, dan modul LCD 16x2 serta sumber tegangan DC +9V atau +5V

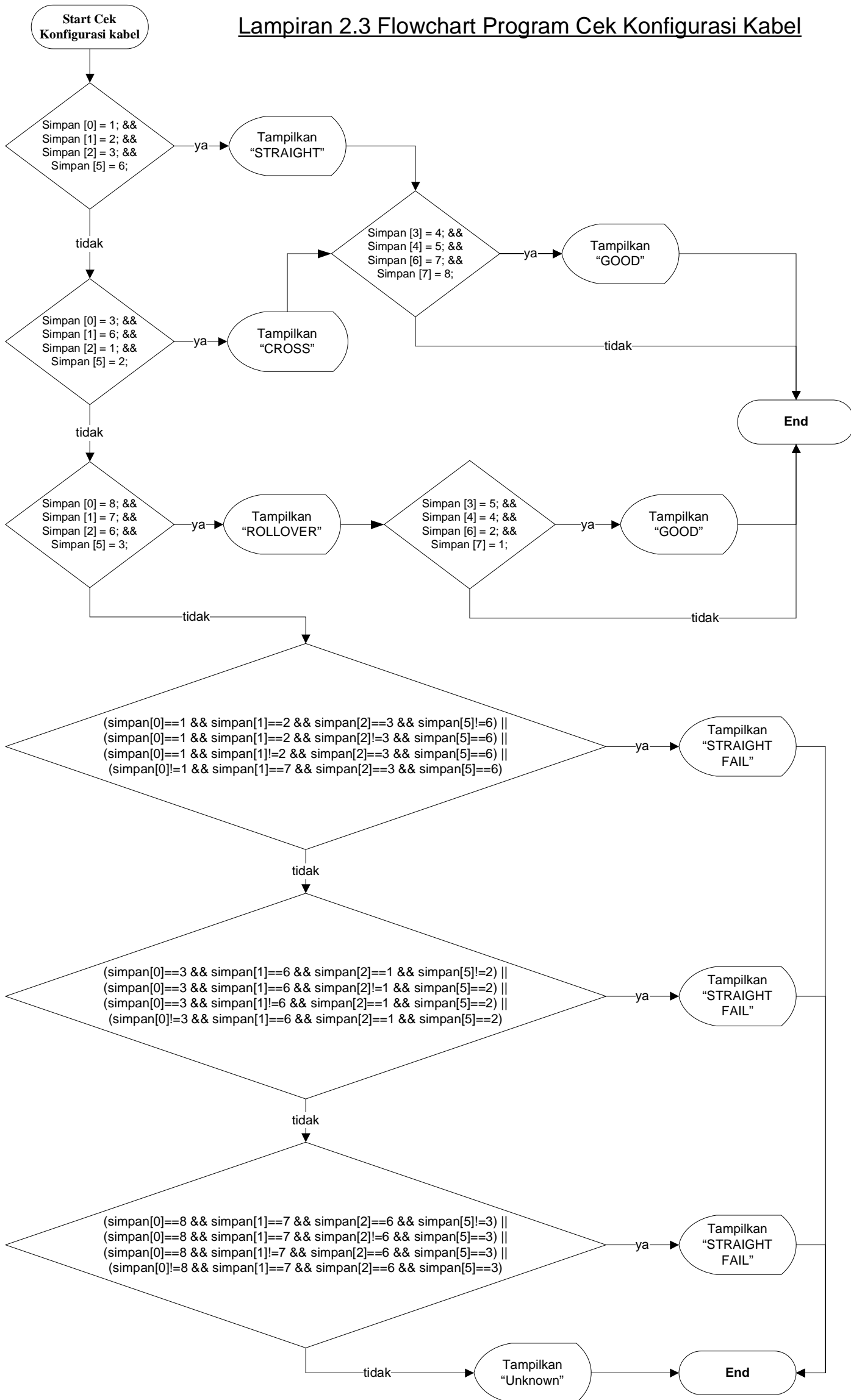
Lampiran 2.1 Flowchart Program Secara Umum, Cek Kabel Terpasang, dan Uji kabel



Lampiran 2.2 Flowchart Program Baca, Simpan dan Tampilkan Data



Lampiran 2.3 Flowchart Program Cek Konfigurasi Kabel



Lampiran 3.1 Kode Program Alat Penguji Kabel LAN Berbasis Mikrokontroler

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79

```

```

/*****
This program was produced by the
CodeWizardAVR V2.05.1b Evaluation
Automatic Program Generator
© Copyright 1998-2011 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com

Project : Skripsi LAN Tester
Version :
Date : 04/06/2011
Author : Fajar Ramadhan
Company : Pendidikan Teknik Elektronika UNJ
Comments:
Chip type : ATmega8535L
Program type : Application
AVR Core Clock frequency: 8,000000 MHz
Memory model : Small
External RAM size : 0
Data Stack size : 128
*****/

#include <mega8535.h>
#include <delay.h>
#include <string.h>
#include <stdio.h>

// Alphanumeric LCD Module functions
#include <alcd.h>

// Declare your global variables here
char buffer_lcd[]="LAN TESTER READY ";
char lcd_number = 16;
char buf[2],buff[2];
char simpan[8],simpan2[8],reset[8];
char a,b,c,d;

void tampilan()
{
  unsigned int i,j,k;
  unsigned int data_len = strlen(buffer_lcd);
  lcd_clear();
  lcd_gotoxy(0,0);
  lcd_putsf("ELEKTRONIKA>>UNJ");

  for (i=lcd_number; i>=0; --i){
    if (i > lcd_number)break;
    lcd_gotoxy(i,1);
    for (j=0; j<(lcd_number-i); j++){
      lcd_putchar(buffer_lcd[j]);
    }
    delay_ms(200);
  };
  k=0;
  for (i=0; i<=data_len; i++){
    k++;
    lcd_gotoxy(0,1);
    for (j=0; j<lcd_number; j++){
      if (buffer_lcd[j+k] == NULL){
        return;
      }
      lcd_putchar(buffer_lcd[j+k]);
    }
    delay_ms(200);
  };
}

void warning() // indikator jika kabel belum terpasang
{
  b=0;c=4;PORTC=0x00;
  if (PINA==0xFF)
  {
    lcd_clear();
    for (a=0;a<=7;a++)
    {simpan2[a]=reset[a];}
    loop:
    lcd_gotoxy(0,0);
    lcd_putsf(" NOT CONNECT! ");
  }
}

```

```

80     delay_ms(1000);
81 loop1:
82     lcd_gotoxy(c,1);
83     lcd_putsf(">");
84     delay_ms(1000);
85     c++;b++;
86     if (b==8)
87     {
88         delay_ms(500);
89         lcd_gotoxy(0,0);
90         lcd_putsf("    Please    ");
91         lcd_gotoxy(0,1);
92         lcd_putsf(" Insert Cable!! ");
93         b=0;c=4;
94     loop2:
95         if (PINA==0xFF)
96         {
97             b++;
98             delay_ms(1000);
99             if (b==10)
100                {b=0;lcd_clear();goto loop1;}
101             else
102                 goto loop2;
103         }
104     }
105     if (PINA==0xFF)
106     {goto loop1;}
107     else
108         {b=0;c=4;lcd_clear();}
109 }
110 }
111
112 void kirim_data ()
113 {
114     PORTC=~d;
115     d<<=1;
116     delay_ms(500);
117 }
118
119 void tampil_data() //menampilkan hasil baca data yang telah disimpan
120 {
121     lcd_gotoxy(a,0);
122     sprintf(buf,"%x",a+1);
123     lcd_puts(buf);
124     if(simpan[a]==9)
125     {lcd_gotoxy(a,1);
126     lcd_puts("X");}
127     else
128     {lcd_gotoxy(a,1);
129     sprintf(buff,"%x",simpan[a]);
130     lcd_puts(buff);}
131 }
132
133 void baca_data () // baca data kabel lalu simpan
134 {
135     if (PINA==~0b00000001)
136     {simpan[a]=1;}
137     else if (PINA==~0b00000010)
138     {simpan[a]=2;}
139     else if (PINA==~0b00000100)
140     {simpan[a]=3;}
141     else if (PINA==~0b00001000)
142     {simpan[a]=4;}
143     else if (PINA==~0b00010000)
144     {simpan[a]=5;}
145     else if (PINA==~0b00100000)
146     {simpan[a]=6;}
147     else if (PINA==~0b01000000)
148     {simpan[a]=7;}
149     else if (PINA==~0b10000000)
150     {simpan[a]=8;}
151     else
152     {simpan[a]=9;}
153 }
154
155 void cek_kabel() //konfigurasi kabel
156 {
157     if (simpan[0]==1 && simpan[1]==2 && simpan[2]==3 && simpan[5]==6)
158     {

```

```

159     if (simpan[3]==4 && simpan[4]==5 && simpan[6]==7 && simpan[7]==8)
160     {
161         lcd_gotoxy(9,0);
162         lcd_putsf("STRGHT ");
163         lcd_gotoxy(9,1);
164         lcd_putsf("GOOD  ");
165     }
166     else
167     {
168         lcd_gotoxy(9,0);
169         lcd_putsf("STRGHT ");
170         lcd_gotoxy(9,1);
171         lcd_putsf("      ");
172     }
173 }
174 else if ((simpan[0]==1 && simpan[1]==2 && simpan[2]==3 && simpan[5]!=6) ||
175          (simpan[0]==1 && simpan[1]==2 && simpan[2]!=3 && simpan[5]==6) ||
176          (simpan[0]==1 && simpan[1]!=2 && simpan[2]==3 && simpan[5]==6) ||
177          (simpan[0]!=1 && simpan[1]==2 && simpan[2]==3 && simpan[5]==6))
178 {
179     lcd_gotoxy(9,0);
180     lcd_putsf("STRGHT ");
181     lcd_gotoxy(9,1);
182     lcd_putsf("FAIL!  ");
183 }
184 else if (simpan[0]==3 && simpan[1]==6 && simpan[2]==1 && simpan[5]==2)
185 {
186     lcd_gotoxy(9,0);
187     lcd_putsf("CROSS  ");
188     lcd_gotoxy(9,1);
189     lcd_putsf("      ");
190     if (simpan[3]==4 && simpan[4]==5 && simpan[6]==7 && simpan[7]==8)
191     {
192         lcd_gotoxy(9,1);
193         lcd_putsf("GOOD  ");
194     }
195 }
196 else if ((simpan[0]==3 && simpan[1]==6 && simpan[2]==1 && simpan[5]!=2) ||
197          (simpan[0]==3 && simpan[1]==6 && simpan[2]!=1 && simpan[5]==2) ||
198          (simpan[0]==3 && simpan[1]!=6 && simpan[2]==1 && simpan[5]==2) ||
199          (simpan[0]!=3 && simpan[1]==6 && simpan[2]==1 && simpan[5]==2))
200 {
201     lcd_gotoxy(9,0);
202     lcd_putsf("CROSS  ");
203     lcd_gotoxy(9,1);
204     lcd_putsf("FAIL!  ");
205 }
206 else if (simpan[0]==8 && simpan[1]==7 && simpan[2]==6 && simpan[5]==3)
207 {
208     if (simpan[3]==5 && simpan[4]==4 && simpan[6]==2 && simpan[7]==1)
209     {
210         lcd_gotoxy(9,0);
211         lcd_putsf("ROLLOVR");
212         lcd_gotoxy(9,1);
213         lcd_putsf("GOOD  ");
214     }
215     else
216     {
217         lcd_gotoxy(9,0);
218         lcd_putsf("ROLL  ");
219         lcd_gotoxy(9,1);
220         lcd_putsf("OVER  ");
221     }
222 }
223 else if ((simpan[0]==8 && simpan[1]==7 && simpan[2]==6 && simpan[5]!=3) ||
224          (simpan[0]==8 && simpan[1]==7 && simpan[2]!=6 && simpan[5]==3) ||
225          (simpan[0]==8 && simpan[1]!=7 && simpan[2]==6 && simpan[5]==3) ||
226          (simpan[0]!=8 && simpan[1]==7 && simpan[2]==6 && simpan[5]==3) )
227 {
228     lcd_gotoxy(9,0);
229     lcd_putsf("ROLLOVR");
230     lcd_gotoxy(9,1);
231     lcd_putsf("FAIL!  ");
232 }
233 else if (simpan[0]==9 && simpan[1]==9 && simpan[2]==9 && simpan[3]==9
234          && simpan[4]==9 && simpan[5]==9 && simpan[6]==9 && simpan[7]==9)
235 {lcd_clear();warning();}
236 else
237 {

```

```

238     lcd_gotoxy(9,0);
239     lcd_putsf("UNKNOWN");
240     lcd_gotoxy(9,1);
241     lcd_putsf("CONFIG!");
242     }
243 }
244
245 void main(void)
246 {
247     // Declare your local variables here
248
249     // Input/Output Ports initialization
250     // Port A initialization
251     PORTA=0xFF;
252     DDRA=0x00;
253
254     // Port B initialization
255     PORTB=0xFF;
256     DDRB=0x00;
257
258     // Port C initialization
259     PORTC=0x00;
260     DDRC=0xFF;
261
262     // Port D initialization
263     PORTD=0x00;
264     DDRD=0x00;
265
266     // Timer/Counter 0 initialization
267     // Clock source: System Clock
268     // Clock value: Timer 0 Stopped
269     // Mode: Normal top=0xFF
270     // OC0 output: Disconnected
271     TCCR0=0x00;
272     TCNT0=0x00;
273     OCR0=0x00;
274
275     // Timer/Counter 1 initialization
276     // Clock source: System Clock
277     // Clock value: Timer1 Stopped
278     // Mode: Normal top=0xFFFF
279     // OC1A output: Discon.
280     // OC1B output: Discon.
281     // Noise Canceler: Off
282     // Input Capture on Falling Edge
283     // Timer1 Overflow Interrupt: Off
284     // Input Capture Interrupt: Off
285     // Compare A Match Interrupt: Off
286     // Compare B Match Interrupt: Off
287     TCCR1A=0x00;
288     TCCR1B=0x00;
289     TCNT1H=0x00;
290     TCNT1L=0x00;
291     ICR1H=0x00;
292     ICR1L=0x00;
293     OCR1AH=0x00;
294     OCR1AL=0x00;
295     OCR1BH=0x00;
296     OCR1BL=0x00;
297
298     // Timer/Counter 2 initialization
299     // Clock source: System Clock
300     // Clock value: Timer2 Stopped
301     // Mode: Normal top=0xFF
302     // OC2 output: Disconnected
303     ASSR=0x00;
304     TCCR2=0x00;
305     TCNT2=0x00;
306     OCR2=0x00;
307
308     // External Interrupt(s) initialization
309     // INT0: Off
310     // INT1: Off
311     // INT2: Off
312     MCUCR=0x00;
313     MCUCSR=0x00;
314
315     // Timer(s)/Counter(s) Interrupt(s) initialization
316     TIMSK=0x00;
317

```



```

318 // USART initialization
319 // USART disabled
320 UCSRB=0x00;
321
322 // Analog Comparator initialization
323 // Analog Comparator: Off
324 // Analog Comparator Input Capture by Timer/Counter 1: Off
325 ACSR=0x80;
326 SFIOR=0x00;
327
328 // ADC initialization
329 // ADC disabled
330 ADCSRA=0x00;
331
332 // SPI initialization
333 // SPI disabled
334 SPCR=0x00;
335
336 // TWI initialization
337 // TWI disabled
338 TWCR=0x00;
339
340 // Alphanumeric LCD initialization
341 // Connections are specified in the
342 // Project/Configure/C Compiler/Libraries/Alphanumeric LCD menu:
343 // RS - PORTD Bit 0
344 // RD - PORTD Bit 1
345 // EN - PORTD Bit 2
346 // D4 - PORTD Bit 4
347 // D5 - PORTD Bit 5
348 // D6 - PORTD Bit 6
349 // D7 - PORTD Bit 7
350 // Characters/line: 16
351 lcd_init(16);
352 tampilan();
353 lcd_clear();
354
355 while (1)
356 {
357 // Place your code here
358 b=0;c=0;d=1;
359 ulang:
360 PORTC=0x00;
361 delay_ms(500);
362 if (PINA==0xFF)
363 {warning();}
364 else
365 {
366 for(a=0;a<=7;a++)
367 {
368 if(c<=7)
369 {lcd_gotoxy(a,0);
370 sprintf(buf,"%x",a+1);
371 lcd_puts(buf);c++;}
372 kirim_data();
373 baca_data();
374 if(simpan[a]!=simpan2[a])
375 {simpan2[a]=simpan[a];
376 tampil_data();
377 delay_ms(500);
378 }
379 else
380 {b++;}
381 }
382 if(c==8)
383 {cek_kabel();c++;d=1;goto ulang;}
384 else if(b!=8)
385 {cek_kabel();b=0;d=1;goto ulang;}
386 else
387 {b=0;d=1;goto ulang;}
388 }
389 }
390 }

```

Lampiran 3.2 Kode Program Uji Modul

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79

```

```

/*****
This program was produced by the
CodeWizardAVR V2.05.1b Evaluation
Automatic Program Generator
© Copyright 1998-2011 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com

Project :
Version :
Date : 04/06/2011
Author : Freeware, for evaluation and non-commercial use only
Company :
Comments:

Chip type : ATmega8535L
Program type : Application
AVR Core Clock frequency: 8,000000 MHz
Memory model : Small
External RAM size : 0
Data Stack size : 128
*****/

#include <mega8535.h>
#include <delay.h>
#include <string.h>
#include <stdio.h>

// Alphanumeric LCD Module functions
#include <alcd.h>

// Declare your global variables here
char buffer_lcd[]="PROGRAM UJI MODUL ";
char lcd_number = 16;
//char buf[3],buff[8];
//unsigned char simpan[10];
//int a,b,d=1;
// Function prototype
//void tampilan(void);
//void data (void);
// Display routine :D
void tampilan()
{
  unsigned int i,j,k;
  unsigned int data_len = strlen(buffer_lcd);
  //_lcd_ready();
  lcd_clear();
  lcd_gotoxy(0,0);
  lcd_putsf("#FAJAR RAMADHAN#");

  for (i=lcd_number; i>=0; --i){
    if (i > lcd_number)break;
    lcd_gotoxy(i,1);
    for (j=0; j<(lcd_number-i); j++){
      lcd_putchar(buffer_lcd[j]);
    }
    delay_ms(150);
  };
  k=0;
  for (i=0; i<=data_len; i++){
    k++;
    lcd_gotoxy(0,1);
    for (j=0; j<lcd_number; j++){
      if (buffer_lcd[j+k] == NULL){
        return;
      }
      lcd_putchar(buffer_lcd[j+k]);
    }
    delay_ms(150);
  };
}

void main(void)
{
  // Declare your local variables here

  // Input/Output Ports initialization
  // Port A initialization

```

```

80 PORTA=0xFF;
81 DDRA=0x00;
82
83 // Port B initialization
84 PORTB=0xFF;
85 DDRB=0x00;
86
87 // Port C initialization
88 PORTC=0x00;
89 DDRC=0xFF;
90
91 // Port D initialization
92 PORTD=0x00;
93 DDRD=0x00;
94
95 // Timer/Counter 0 initialization
96 // Clock source: System Clock
97 // Clock value: Timer 0 Stopped
98 // Mode: Normal top=0xFF
99 // OC0 output: Disconnected
100 TCCR0=0x00;
101 TCNT0=0x00;
102 OCR0=0x00;
103
104 // Timer/Counter 1 initialization
105 // Clock source: System Clock
106 // Clock value: Timer1 Stopped
107 // Mode: Normal top=0xFFFF
108 // OC1A output: Discon.
109 // OC1B output: Discon.
110 // Noise Canceler: Off
111 // Input Capture on Falling Edge
112 // Timer1 Overflow Interrupt: Off
113 // Input Capture Interrupt: Off
114 // Compare A Match Interrupt: Off
115 // Compare B Match Interrupt: Off
116 TCCR1A=0x00;
117 TCCR1B=0x00;
118 TCNT1H=0x00;
119 TCNT1L=0x00;
120 ICR1H=0x00;
121 ICR1L=0x00;
122 OCR1AH=0x00;
123 OCR1AL=0x00;
124 OCR1BH=0x00;
125 OCR1BL=0x00;
126
127 // Timer/Counter 2 initialization
128 // Clock source: System Clock
129 // Clock value: Timer2 Stopped
130 // Mode: Normal top=0xFF
131 // OC2 output: Disconnected
132 ASSR=0x00;
133 TCCR2=0x00;
134 TCNT2=0x00;
135 OCR2=0x00;
136
137 // External Interrupt(s) initialization
138 // INT0: Off
139 // INT1: Off
140 // INT2: Off
141 MCUCR=0x00;
142 MCUCSR=0x00;
143
144 // Timer(s)/Counter(s) Interrupt(s) initialization
145 TIMSK=0x00;
146
147 // USART initialization
148 // USART disabled
149 UCSRB=0x00;
150
151 // Analog Comparator initialization
152 // Analog Comparator: Off
153 // Analog Comparator Input Capture by Timer/Counter 1: Off
154 ACSR=0x80;
155 SFIOR=0x00;
156
157 // ADC initialization
158 // ADC disabled
159 ADCSRA=0x00;

```

```

160
161 // SPI initialization
162 // SPI disabled
163 SPCR=0x00;
164
165 // TWI initialization
166 // TWI disabled
167 TWCR=0x00;
168
169 // Alphanumeric LCD initialization
170 // Connections are specified in the
171 // Project/Configure/C Compiler/Libraries/Alphanumeric LCD menu:
172 // RS - PORTD Bit 0
173 // RD - PORTD Bit 1
174 // EN - PORTD Bit 2
175 // D4 - PORTD Bit 4
176 // D5 - PORTD Bit 5
177 // D6 - PORTD Bit 6
178 // D7 - PORTD Bit 7
179 // Characters/line: 16
180 lcd_init(16);
181 //_lcd_ready();
182
183 tampilan();
184 lcd_clear();
185
186 while (1)
187 {
188     // Place your code here
189
190     lcd_gotoxy(0,0);
191     lcd_putsf(" UJI PORT I/O ");
192     PORTC=0x00;
193     delay_ms(2000);
194     if (PINA==0x00)
195     {lcd_gotoxy(0,1);
196     lcd_putsf(" PORT I/O = OK! ");
197     }
198     else
199     {
200         lcd_gotoxy(0,1);
201         lcd_putsf("PERIKSA PORT I/O");
202     }
203 }
204 }

```

Lampiran 3.2 Kode Program Voltmeter dengan ADC

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79

```

```

/*****
This program was produced by the
CodeWizardAVR V2.05.0 Professional
Automatic Program Generator
© Copyright 1998-2010 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com

Project :
Version :
Date    : 22/06/2011
Author  :
Company :
Comments:

Chip type      : ATmega8535L
Program type   : Application
AVR Core Clock frequency: 8,000000 MHz
Memory model   : Small
External RAM size : 0
Data Stack size : 128
*****/

#include <mega8535.h>
#include <math.h>
#include <delay.h>
#include <string.h>
#include <stdio.h>

// Alphanumeric LCD Module functions
#include <alcd.h>

#define ADC_VREF_TYPE 0x40

// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
    // Delay needed for the stabilization of the ADC input voltage
    delay_us(10);
    // Start the AD conversion
    ADCSRA|=0x40;
    // Wait for the AD conversion to complete
    while ((ADCSRA & 0x10)==0);
    ADCSRA|=0x10;
    return ADCW;
}

// Declare your global variables here
char buffer[16];
float tegangan; //Untuk data perhitungan tegangan (float)
unsigned int temporary; //Untuk temporary data ADC (komparasi)

void main(void)
{
    // Declare your local variables here

    // Input/Output Ports initialization
    PORTA=0xFF;
    DDRA=0x00;

    PORTB=0x00;
    DDRB=0x00;

    PORTC=0x00;
    DDRC=0xFF;

    PORTD=0x00;
    DDRD=0x00;

    // Timer/Counter 0 initialization
    // Clock source: System Clock
    // Clock value: Timer 0 Stopped
    // Mode: Normal top=0xFF
    // OCO output: Disconnected
    TCCR0=0x00;
    TCNT0=0x00;
    OCR0=0x00;

```

```

80 // Timer/Counter 1 initialization
81 // Clock source: System Clock
82 // Clock value: Timer1 Stopped
83 // Mode: Normal top=0xFFFF
84 // OC1A output: Discon.
85 // OC1B output: Discon.
86 // Noise Canceler: Off
87 // Input Capture on Falling Edge
88 // Timer1 Overflow Interrupt: Off
89 // Input Capture Interrupt: Off
90 // Compare A Match Interrupt: Off
91 // Compare B Match Interrupt: Off
92 TCCR1A=0x00;
93 TCCR1B=0x00;
94 TCNT1H=0x00;
95 TCNT1L=0x00;
96 ICR1H=0x00;
97 ICR1L=0x00;
98 OCR1AH=0x00;
99 OCR1AL=0x00;
100 OCR1BH=0x00;
101 OCR1BL=0x00;
102
103 // Timer/Counter 2 initialization
104 // Clock source: System Clock
105 // Clock value: Timer2 Stopped
106 // Mode: Normal top=0xFF
107 // OC2 output: Disconnected
108 ASSR=0x00;
109 TCCR2=0x00;
110 TCNT2=0x00;
111 OCR2=0x00;
112
113 // External Interrupt(s) initialization
114 // INT0: Off
115 // INT1: Off
116 // INT2: Off
117 MCUCR=0x00;
118 MCUCSR=0x00;
119
120 // Timer(s)/Counter(s) Interrupt(s) initialization
121 TIMSK=0x00;
122
123 // USART initialization
124 // USART disabled
125 UCSRB=0x00;
126
127 // Analog Comparator initialization
128 // Analog Comparator: Off
129 // Analog Comparator Input Capture by Timer/Counter 1: Off
130 ACSR=0x80;
131 SFIOR=0x00;
132
133 // ADC initialization
134 // ADC Clock frequency: 125,000 kHz
135 // ADC Voltage Reference: AVCC pin
136 // ADC High Speed Mode: Off
137 // ADC Auto Trigger Source: ADC Stopped
138 ADMUX=ADC_VREF_TYPE & 0xff;
139 ADCSRA=0x86;
140 SFIOR&=0xEF;
141
142 // SPI initialization
143 // SPI disabled
144 SPCR=0x00;
145
146 // TWI initialization
147 // TWI disabled
148 TWCR=0x00;
149
150 // Alphanumeric LCD initialization
151 // Connections specified in the
152 // Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
153 // RS - PORTD Bit 0
154 // RD - PORTD Bit 1
155 // EN - PORTD Bit 2
156 // D4 - PORTD Bit 4
157 // D5 - PORTD Bit 5
158 // D6 - PORTD Bit 6
159 // D7 - PORTD Bit 7

```

```
160 // Characters/line: 16
161 lcd_init(16);
162 while (1)
163 {
164     // Place your code here
165
166     read_adc(0);
167     if (temporary != read_adc(0)){
168         tegangan = 5-((read_adc(0)*4.8828125)/1000);
169         lcd_gotoxy(0,0);
170         lcd_putsf(" - 737rr - ");
171         lcd_gotoxy(0,1);
172         sprintf(buffer,"VOLT:%f V",tegangannya);
173         lcd_puts(buffer);
174         temporary = read_adc(0);
175         delay_ms(1000);
176     }
177 }
178 }
179 }
```