

Advanced Trapwire (Using Laser)

Sultan Fahd MBY- 2106724031, Nisrina Nurkhomsatun P - 2106723956

Departemen Fisika, Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Indonesia
Depok, Jawa Barat 16436
sultan.fahd@ui.ac.id

Trapwire security system adalah sistem keamanan yang menggunakan prinsip elektronika. Sistem ini terdiri dari komponen-komponen analog elektronika, seperti resistor, kapasitor, transistor, potensiometer, timer 555, serta LED dan buzzer sebagai indikator. Rangkaian sistem ini memiliki dua rangkaian yang digabung menjadi satu, yaitu rangkaian LDR dan buzzer sebagai rangkaian yang merespon laser serta rangkaian LED Blip sebagai indikator.

PENDAHULUAN

Ilmu elektronika banyak digunakan di dunia modern. Ilmu ini dapat ditemukan dalam bentuk perangkat elektronika yang dapat membantu manusia dalam kehidupannya. Salah satu bentuk penerapan ilmu elektronika di kehidupan sehari-hari adalah sistem keamanan, baik pada suatu bangunan, maupun ruangan.

Salah satu sistem keamanan yang dapat dibuat adalah Trapwire Security System. Prototipe dari trapwire ini dibuat pada breadboard dan dirancang untuk dapat mendeteksi adanya penyusup. Hal ini bertujuan untuk melakukan perlindungan terhadap pencurian harta benda atau properti yang berharga. Sistem ini biasa ditemukan di tempat yang membutuhkan tingkat keamanan yang tinggi, seperti museum, bank, area militer, dan perusahaan berstandar tinggi,

Sistem keamanan ini memanfaatkan komponen LDR (Light Dependent Resistor) menjadi salah satu sistem keamanan yang cukup efisien. LDR merupakan sensor cahaya yang dapat mengubah intensitas cahaya menjadi hambatan listrik. Semakin banyak cahaya yang diterima oleh permukaan LDR, hambatan listrik menjadi kecil. Hal ini juga berlaku sebaliknya.

Rangkaian trapwire security system ini dibuat dengan menggunakan komponen transistor, buzzer, resistor, kapasitor, LDR, timer 555, serta LED. Pada rangkaian ini, praktikan menggabungkan dua rangkaian, yaitu rangkaian respon yang terdiri dari beberapa komponen, yaitu LDR, timer 555, resistor, dan buzzer, serta rangkaian indikator yang terdiri dari transistor, LED, resistor, dan kapasitor.

TEORI DASAR

Komparator merupakan komponen elektronik yang berfungsi untuk membandingkan dua nilai dan akan menghasilkan suatu output, yang bergantung pada dua nilai tersebut. Komparator dapat dibuat dari konfigurasi open-loop Op-Amp. Jika kedua input pada Op-Amp pada kondisi open loop, maka Op-Amp akan membandingkan kedua saluran tersebut. Hasil perbandingan kedua tegangan pada saluran masukan akan menghasilkan tegangan positif atau tegangan negative.

OP-AMP sebagai komparator

Operasional amplifier (Op-Amp) adalah suatu penguat berpenguatan tinggi yang terintegrasi dalam sebuah chip IC yang memiliki dua input inverting dan non-inverting dengan sebuah terminal output, dimana rangkaian umpan balik dapat ditambahkan untuk mengendalikan karakteristik tanggapan keseluruhan pada operasional amplifier (Op-Amp).

Prinsip kerja sebuah Op-Amp adalah membandingkan nilai kedua input (input inverting dan input non-inverting), apabila kedua input bernilai sama maka output op-amp tidak ada (nol) dan apabila terdapat perbedaan nilai input keduanya maka output op-amp akan memberikan tegangan output. Op-Amp dibuat dari penguat diferensial dengan 2 input.

Sebuah rangkaian komparator pada Op-Amp akan membandingkan tegangan yang masuk pada satu saluran input dengan tegangan pada saluran input lain, yang disebut tegangan referensi. Tegangan output berupa tegangan HIGH atau LOW sesuai dengan perbandingan V_{in} dan V_{ref} .

LDR (Light Dependent Resistor)

Resistor foto atau Light-Dependent Resistor (LDR) atau fotosel adalah resistor variabel yang dikontrol cahaya. Hambatan resistor foto berkurang dengan meningkatnya intensitas cahaya; dengan kata lain, menunjukkan konduktivitas foto. Resistor foto dapat diterapkan di sirkuit detektor peka cahaya, dan sirkuit pengaktifan terang dan gelap. Resistor foto bekerja berdasarkan prinsip fotokonduktivitas. Fotokonduktivitas adalah fenomena optik di mana konduktivitas material meningkat ketika cahaya diserap oleh material. Ketika cahaya jatuh atau ketika foton menyentuh perangkat, elektron di pita valensi bahan semikonduktor tereksitasi ke pita konduksi. Foton-foton di cahaya datang harus memiliki energi yang lebih besar daripada celah pita bahan semikonduktor untuk membuat elektron berpindah dari pita valensi ke pita konduksi. Oleh karena itu, ketika cahaya memiliki energi yang cukup mengenai perangkat, semakin banyak elektron yang tereksitasi ke pita konduksi sehingga menghasilkan sejumlah besar pembawa muatan. Hasil dari proses ini adalah terdapat banyak arus yang mulai mengalir melalui perangkat ketika rangkaian ditutup karena resistansi perangkat telah menurun.

Prinsip Kerja LDR

Dalam keadaan gelap, fotoresistor dapat memiliki resistansi setinggi beberapa mega ohm (M ohm), sedangkan dalam keadaan terang, fotoresistor dapat memiliki resistansi serendah beberapa ratus ohm. Jika cahaya datang pada fotoresistor melebihi frekuensi tertentu, foton yang diserap oleh semikonduktor memberikan elektron yang terikat energi yang cukup untuk berpindah ke pita konduksi. Elektron bebas yang dihasilkan menghantarkan listrik, sehingga menurunkan resistansi. Kisaran resistansi dan sensitivitas fotoresistor sangat berbeda di antara perangkat yang berbeda. Selain itu, beberapa resistor foto dapat bereaksi secara substansial berbeda terhadap foton dalam pita panjang gelombang tertentu.

IC 555 : Mode Bistable

Bistable merupakan rangkaian fungsi yang dapat dibentuk oleh IC 555 yang berfungsi sebagai sebuah flip-flop, dapat disebut demikian karena dalam rangkaian ini ada dua kondisi yang diperoleh dari output rangkaian. Mode ini disebut juga sebagai Schmitt trigger. Pada mode ini, IC 555 dapat beroperasi sebagai flip-flop. Jika kaki Discharge atau pin 7 tidak terhubung ke kapasitor. Penggunaannya

meliputi pencacah binerm dan *bouncefree switch latched*.

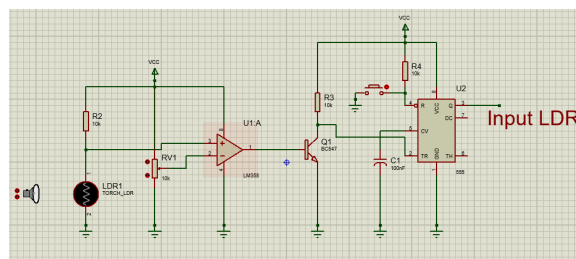
CARA KERJA

Pada proyek akhir praktikum elektronika ini, praktikan akan merancang sebuah keamanan sistem lebih lanjut menggunakan sensor cahaya dan alarm sekuensial dengan bantuan Bahasa VHDL.

Pertama-tama praktikan akan membuat rangkaian seperti pada gambar 10.1. Kemudian hubungkan vcc pada Op-Amp 358 dan ICs 555 timer ke vcc ZYBO-7000. Setelah itu, letakan sinar laser agar tepat jatuh pada LDR(sensor cahaya).

Setelah itu praktikan membuat kode untuk alarm sekuensial menggunakan Bahasa VHDL dengan state seperti pada diagram di lapiran. Diusahakan hanya menggunakan sumber tegangan dari ZYBO-7000. Hal tersebut dikarenakan ZYBO-7000 akan mengalami kerusakan jika mendapatkan sumber vcc eksternal.

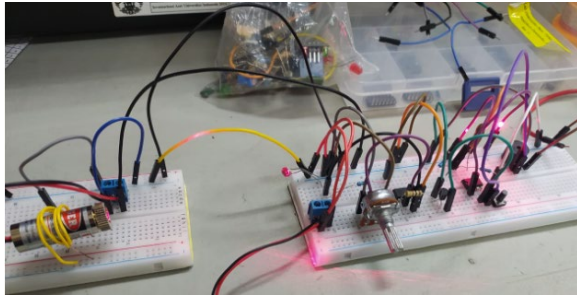
Setelah program telah dibuat, kemudian praktikan mengatur definisi serta constraint I/O serta mensimulasikannya dan mengimplementasikannya pada ZYBO-7000. Lalu menghubungkan PMOD yang telah didefinisikan sebagai I/O pada LED, buzzer, dan LDR. Untuk input LDR ini, praktikan menghubungkannya dengan output ICs 555 timer. Setelah itu, melihat keadaan alarm saat LDR tidak mendapatkan cahaya.



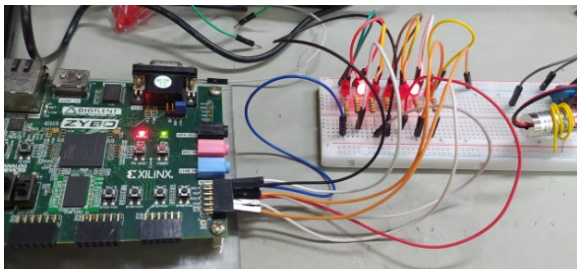
Gambar 10.1 Rangkaian Trapwire

HASIL

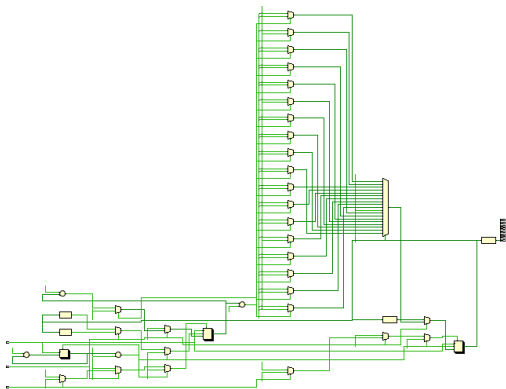
Rangkaian Trapwire



Rangkaian alarm sekuensial



Rangkaian skematik alarm sekuensial



PEMBAHASAN

Pada proyek akhir praktikum elektronika ini, praktikan akan mengaplikasikan dari rangkaian digital dan Bahasa pemrograman VHDL untuk alarm. Pada kesempatan ini, praktikan akan mengontruksi sistem keamanan menggunakan sensor cahaya beserta alarm sekuensialnya.

Rangkaian komparator Op-Amp 358 berfungsi sebagai pembanding dua buah tegangan. Kemudian berdasarkan perbandingan dua sinyal analog tersebut akan dikeluarkan output berupa tegangan sesuai dengan nilai VCC dari zybo. Kemudian pada kaki non-inverting Op-Amp, ditambahkan sebuah LDR dalam rangkaian pembagi tegangan. Hal ini dimanfaatkan jika LDR mendapatkan cahaya laser, maka kaki non-inverting akan lebih kecil dibandingkan kaki inverting. Hal ini dikarenakan nilai hambatan dari LDR dibawah 1K Ω . Begitu pun sebaliknya, jika LDR tidak mendapatkan cahaya laser, maka kaki non-inverting memiliki tegangan yang lebih tinggi dibandingkan kaki inverting. Hal ini karena nilai hambatan LDR menjadi sangat besar yang menyebabkan output pada rangkaian komparator akan bernilai HIGH.

Kemudian rangkaian bi-stable 555 timer digunakan untuk membuat rangkaian dalam 2 kondisi stabil. Saat pin2(trigger) mendapatkan nilai HIGH pada output rangkaian komparator sebelumnya. Maka output pada 555 timer akan bernilai HIGH, Hal ini akan sebagai input sensor ldr yang didefinisikan untuk mengaktifkan alarm sekuensial pada ZYBO-7000. Lalu jika akan menonaktifkan rangkaian sekuensial, praktikan hanya menekan tombol reset pada ICs 555 timer yang telah dikontruksi sebagai rangkaian pull down .

Selanjutnya pada rangkaian alarm sekuensial, praktikan menggunakan Bahasa pemrograman VHDL untuk memudahkan dalam membuat alarm pendukung dari rangkaian sebelumnya. Pada saat output timer 555 bernilai HIGH, maka akan terdefinisi berlogika 1 pada sensor LDR. Hal tersebut akan menyebabkan sekuensial berjalan dari kondisi1 hingga kondisi18. Ini akan terus terulang hingga praktikan menekan tombol pull down pada pin reset timer 555. Kemudian jika input saklar VCC bernilai low atau berlogika "0", maka sekuensial akan menjalankan kodisilost1 dan kondisilost2 secara terus menerus hingga input VCC kembali HIGH.

KESIMPULAN

Laser security system atau Trapwire memberikan kita keamanan untuk melawan tindak kejahatan dilingkungan kita sehari-hari. Berbagai sistem keamanan elektronik dapat digunakan di rumah atau tempat penting lainnya yang bertujuan untuk keamanan dan keselamatan. Hal ini merupakan peluang besar dalam tahap memanfaatkan sumber tenaga listrik dan upaya untuk penghematan tenaga manusia. Dengan adanya sistem keamanan ini, maka perampokan, pencurian, serta kejahatan dapat dihindarkan secara besar-besaran.

Komponen LDR ini sangat sensitif dengan cahaya. Setiap kali laser cahaya terganggu dengan cara apapun, hal tersebut akan mentrigger alarm untuk berbunyi. Oleh karena itu, sistem keamanan ini sangat cocok untuk pengawasan dalam suatu industri

REFERENSI

Buku Penuntun Praktikum Elektronika 2: Elektronika

Digital. Laboratorium Elektronika. Departemen Fisika.

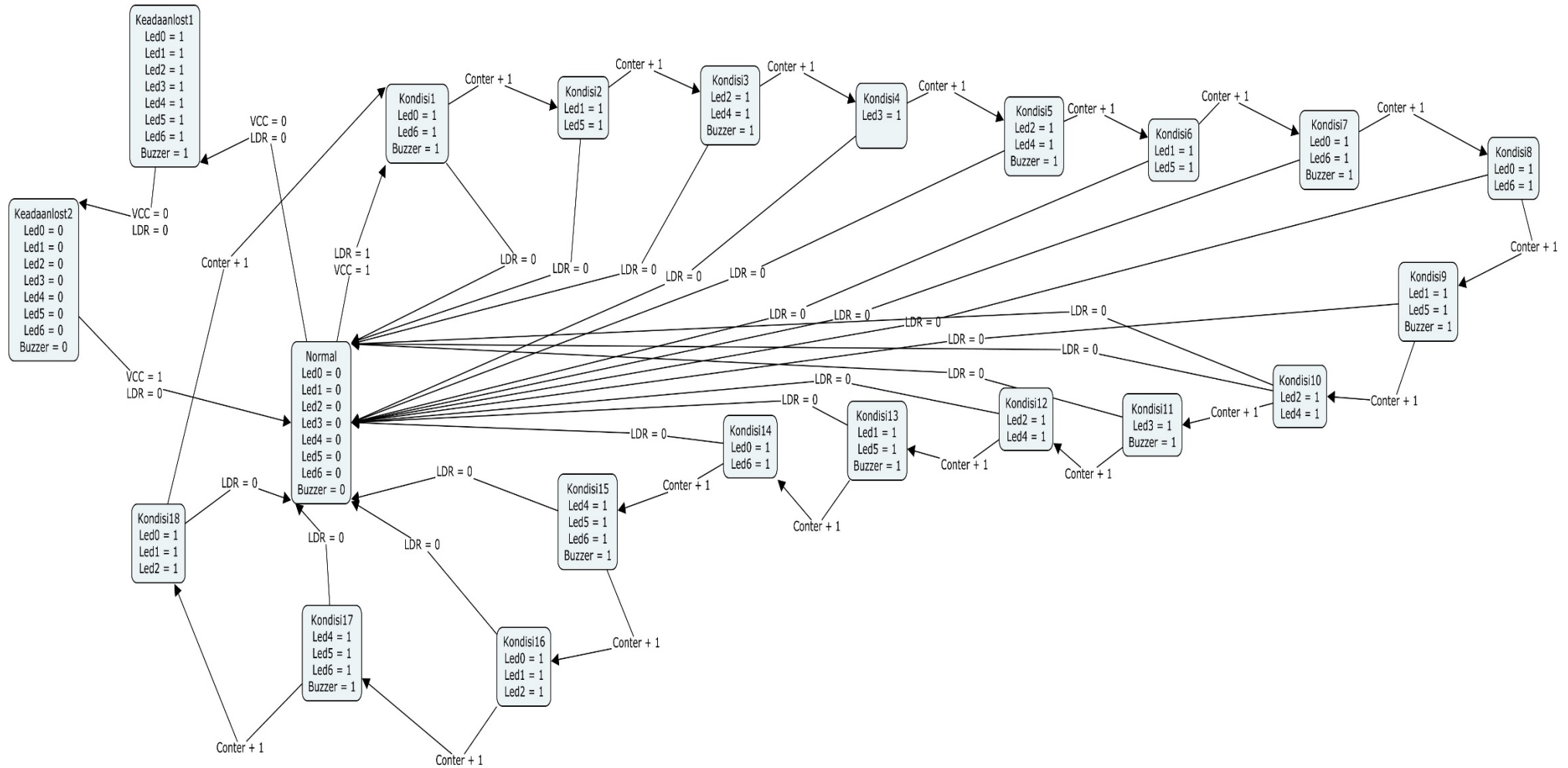
Fakultas Matematika dan Ilmu Pengetahuan Alam.

Universitas Indonesia, 2022.

W. Kleitz. *Digital Electronics: A Practical Approach with VHDL*, 9th edition. Pearson, 2011.

Tocci, R. J., Moss, G. L., & Widmer, N. S. (2017). *Digital Systems*. Pears

DIAGRAM TRAPWIRE STATE MACHINE



LAMPIRAN

PEMROGRAMAN VHDL	HASIL
<pre> 1 library IEEE; 2 use IEEE.STD_LOGIC_1164.ALL; 3 use IEEE.STD_LOGIC_UNSIGNED.ALL; 4 5 6 entity Trapsire is 7 Port (clk : in STD_LOGIC; 8 ldr : in STD_LOGIC; 9 vcc : in STD_LOGIC; 10 red0 : out STD_LOGIC; 11 red1 : out STD_LOGIC; 12 red2 : out STD_LOGIC; 13 red3 : out STD_LOGIC; 14 red4 : out STD_LOGIC; 15 red5 : out STD_LOGIC; 16 red6 : out STD_LOGIC; 17 alarm : out STD_LOGIC); 18 end Trapsire; 19 20 architecture Behavioral of Trapsire is 21 TYPE state_type IS (normal, kondisi1, kondisi2, kondisi3, kondisi4, kondisi5, kondisi6, kondisi7, kondisi8, kondisi9, kondisi10, 22 kondisi11, kondisi12, kondisi13, kondisi14, kondisi15, kondisi16, kondisi17, kondisi18, kondisi19, kondisi20); 23 SIGNAL state : state_type := normal; 24 SIGNAL count : STD_LOGIC_VECTOR (3 downto 0); 25 SIGNAL lighta : STD_LOGIC_VECTOR (7 downto 0); 26 SIGNAL clk_count : natural range 0 to 50000000 := 0; 27 28 BEGIN 29 PROCESS (clk, vcc, ldr) 30 BEGIN 31 IF rising_edge(clk) THEN 32 clk_count <= clk_count + 1 ; 33 IF clk_count >= 25000000 THEN 34 clk_count <= 0; 35 IF vcc = '0' THEN 36 CASE state is 37 WHEN kondisi10 => 38 IF count < "0000" THEN 39 state <= kondisi10; 40 count <= count + 1; 41 ELSE 42 state <= kondisi11; 43 count <= "0000"; 44 END IF; 45 WHEN kondisi11 => 46 IF count < "0000" THEN 47 state <= kondisi11; 48 count <= count + 1; 49 ELSE 50 state <= kondisi12; 51 count <= "0000"; 52 END IF; 53 WHEN others => state <= kondisi10; 54 END CASE; 55 ELSEIF ldr = '0' THEN 56 state <= normal; 57 count <= "0000"; 58 ELSEIF ldr = '1' THEN 59 CASE state is 60 WHEN kondisi1 => </pre>	<p>SKEMATIK</p>

```

63         IF count <= "0000" THEN
64             state <= kondisi1;
65             count <= count + 1;
66         ELSE
67             state <= kondisi2;
68             count <= "0000";
69         END IF;
70
71     WHEN kondisi2 =>
72         IF count <= "0000" THEN
73             state <= kondisi2;
74             count <= count + 1;
75         ELSE
76             state <= kondisi3;
77             count <= "0000";
78         END IF;
79
80     WHEN kondisi3 =>
81         IF count <= "0000" THEN
82             state <= kondisi3;
83             count <= count + 1;
84         ELSE
85             state <= kondisi4;
86             count <= "0000";
87         END IF;
88
89     WHEN kondisi4 =>
90         IF count <= "0000" THEN
91             state <= kondisi4;
92             count <= count + 1;
93         ELSE
94             state <= kondisi5;
95             count <= "0000";
96
97
98     WHEN kondisi5 =>
99         IF count <= "0000" THEN
100             state <= kondisi5;
101             count <= count + 1;
102         ELSE
103             state <= kondisi6;
104             count <= "0000";
105         END IF;
106
107     WHEN kondisi6 =>
108         IF count <= "0000" THEN
109             state <= kondisi6;
110             count <= count + 1;
111         ELSE
112             state <= kondisi7;
113             count <= "0000";
114         END IF;
115
116     WHEN kondisi7 =>
117         IF count <= "0000" THEN
118             state <= kondisi7;
119             count <= count + 1;
120         ELSE
121             state <= kondisi8;
122             count <= "0000";
123         END IF;
124
125     WHEN kondisi8 =>
126         IF count <= "0000" THEN
127             state <= kondisi8;
128             count <= count + 1;
129         ELSE
130             state <= kondisi9;
131             count <= "0000";
132
133     WHEN kondisi9 =>
134         IF count <= "0000" THEN
135             state <= kondisi9;
136             count <= count + 1;
137         ELSE
138             state <= kondisi10;
139             count <= "0000";
140         END IF;
141
142     WHEN kondisi10 =>
143         IF count <= "0000" THEN
144             state <= kondisi10;
145             count <= count + 1;
146         ELSE
147             state <= kondisi11;
148             count <= "0000";
149         END IF;
150
151     WHEN kondisi11 =>
152         IF count <= "0000" THEN
153             state <= kondisi11;
154             count <= count + 1;
155         ELSE
156             state <= kondisi12;
157             count <= "0000";
158         END IF;
159
160     WHEN kondisi12 =>
161         IF count <= "0000" THEN
162             state <= kondisi12;
163             count <= count + 1;
164         ELSE
165             state <= kondisi13;
166             count <= "0000";
167         END IF;

```

```

169      WHEN kondisil3 =>
170          IF count <= "0000" THEN
171              state <= kondisil3;
172              count <= count + 1;
173          ELSE
174              state <= kondisil4;
175              count <= "0000";
176          END IF;
177
178      WHEN kondisil4 =>
179          IF count <= "0000" THEN
180              state <= kondisil4;
181              count <= count + 1;
182          ELSE
183              state <= kondisil5;
184              count <= "0000";
185          END IF;
186
187      WHEN kondisil5 =>
188          IF count <= "0000" THEN
189              state <= kondisil5;
190              count <= count + 1;
191          ELSE
192              state <= kondisil6;
193              count <= "0000";
194          END IF;
195
196      WHEN kondisil6 =>
197          IF count <= "0000" THEN
198              state <= kondisil6;
199              count <= count + 1;
200          ELSE
201              state <= kondisil7;
202              count <= "0000";
203          END IF;
204
205      WHEN kondisil7 =>
206          IF count <= "0000" THEN
207              state <= kondisil7;
208              count <= count + 1;
209          ELSE
210              state <= kondisil8;
211              count <= "0000";
212          END IF;
213
214      WHEN kondisil8 =>
215          IF count <= "0000" THEN
216              state <= kondisil8;
217              count <= count + 1;
218          ELSE
219              state <= kondisil;
220              count <= "0000";
221          END IF;
222
223      WHEN others => state <= kondisil;
224  END CASE;
225  END IF;
226  END IF;
227  END IF;
228  END IF;
229  END PROCESS;

```



```

231 process (state, lights)
232 begin
233     case state is
234         WHEN normal => lights <= "00000000";
235         WHEN kondisi1 => lights <= "11000001";
236         WHEN kondisi2 => lights <= "00100010";
237         WHEN kondisi3 => lights <= "10010100";
238         WHEN kondisi4 => lights <= "00001000";
239         WHEN kondisi5 => lights <= "10010100";
240         WHEN kondisi6 => lights <= "00100010";
241         WHEN kondisi7 => lights <= "11000001";
242         WHEN kondisi8 => lights <= "01000001";
243         WHEN kondisi9 => lights <= "10100010";
244         WHEN kondisi10 => lights <= "00010100";
245         WHEN kondisi11 => lights <= "10001000";
246         WHEN kondisi12 => lights <= "00010100";
247         WHEN kondisi13 => lights <= "10100010";
248         WHEN kondisi14 => lights <= "01000001";
249         WHEN kondisi15 => lights <= "11110000";
250         WHEN kondisi16 => lights <= "00000111";
251         WHEN kondisi17 => lights <= "11110000";
252         WHEN kondisi18 => lights <= "00000111";
253         WHEN kondisi1ost1 => lights <= "11111111";
254         WHEN kondisi1ost2 => lights <= "00000000";
255         WHEN others => lights <= "00000000";
256     END CASE;
257 END PROCESS;
258
259 red0 <= lights(0);
260 red1 <= lights(1);
261 red2 <= lights(2);
262 red3 <= lights(3);
263 red4 <= lights(4);
264 red5 <= lights(5);
265 red6 <= lights(6);
266 alarm <= lights(7);
267
268 end Behavioral;

```