APPENDIX

CODING FUZZIFIKASI

```
4. void FuzzifikasiMetana()
5. {
6.   // untuk kondisi baik
7.   if (MQ2Lecture <=1)
8.   {
9.     metana [0] = 1;
10.        }
11.        else if (MQ2Lecture > 1 && MQ2Lecture <= 2)
12.        {
13.          metana [0] = (2 - MQ2Lecture) / (2 - 1);
14.        }
15.        else
16.        {
17.          metana [0] = 0;
18.        }
19.        //untuk kondisi sedang
20.        if (MQ2Lecture <= 1)
21.        {
22.          metana [1] = 0;
23.        }
24.          else if (MQ2Lecture > 1 && MQ2Lecture <2)
25.        {
26.          metana [1] = (MQ2Lecture - 1) / (2 - 1);
27.        }
28.          else if (MQ2Lecture >=2  && MQ2Lecture <=3)
29.        {
30.          metana [1] = 1;
31.        }
32.          else if (MQ2Lecture >3  && MQ2Lecture <=4)
33.        {
34.          metana [1] = (4 - MQ2Lecture) / (4 - 3);
35.        }
36.        else
37.        {
38.          metana [1] = 0;
39.        }
40.          // untuk kondisi berbahaya
41.        if (MQ2Lecture <= 3)
42.        {
43.          metana [2] = 0;
44.        }
45.        else if (MQ2Lecture > 3 && MQ2Lecture <= 4)
46.        {
47.          metana [2] = (MQ2Lecture - 3) / (4 - 3);
48.        }
49.        else
50.        {
51.          metana [2] = 1;
52.        }
53.
54.        Serial.print(" Metana Baik= ");
```

```
55.         Serial.print(metana [0]);
56.         Serial.print(" Metana Sedang= ");
57.         Serial.print(metana [1]);
58.         Serial.print(" Metana Berbahaya= ");
59.         Serial.print(metana [2]);
60.       }
61.       void FuzzifikasiCO()
62.       {
63.         // untuk kondisi baik
64.         if (MQ7Lecture <=25)
65.         {
66.           co [0] = 1;
67.         }
68.         else if (MQ7Lecture > 25 && MQ7Lecture <= 50)
69.         {
70.           co [0] = (50 - MQ7Lecture) / (50 - 25);
71.         }
72.         else
73.         {
74.           co [0] = 0;
75.         }
76.         //untuk kondisi sedang
77.         if (MQ7Lecture <= 25)
78.         {
79.           co [1] = 0;
80.         }
81.           else if (MQ7Lecture > 25 && MQ7Lecture <50)
82.         {
83.           co [1] = (MQ7Lecture - 25) / (50 - 25);
84.         }
85.           else if (MQ7Lecture >=50  && MQ7Lecture <=75)
86.         {
87.           co [1] = 1;
88.         }
89.           else if (MQ7Lecture >75  && MQ7Lecture <=100)
90.         {
91.           co [1] = (100 - MQ7Lecture) / (100 - 75);
92.         }
93.         else
94.         {
95.           co [1] = 0;
96.         }
97.           // untuk kondisi berbahaya
98.         if (MQ7Lecture <= 75)
99.         {
100.          co [2] = 0;
101.        }
102.        else if (MQ7Lecture > 75 && MQ7Lecture <= 100)
103.        {
104.          co [2] = (MQ7Lecture - 75) / (100 - 75);
105.        }
106.        else
107.        {
108.          co [2] = 1;
109.        }
110.        Serial.print(" CO Baik= ");
```

```
111.        Serial.print(co [0]);
112.        Serial.print(" CO Sedang= ");
113.        Serial.print(co [1]);
114.        Serial.print(" CO Berbahaya= ");
115.        Serial.print(co [2]);
116.      }
117.      void FuzzifikasiCO2()
118.      {
119.        // untuk kondisi baik
120.        if (MQ135Lecture <=150)
121.        {
122.          co2 [0] = 1;
123.        }
124.        else if (MQ135Lecture > 150 && MQ135Lecture <= 200)
125.        {
126.          co2 [0] = (200 - MQ135Lecture) / (200 - 150);
127.        }
128.        else
129.        {
130.          co2 [0] = 0;
131.        }
132.        //untuk kondisi sedang
133.        if (MQ135Lecture <= 150)
134.        {
135.          co2 [1] = 0;
136.        }
137.          else if (MQ135Lecture > 150 && MQ135Lecture <200)
138.        {
139.          co2 [1] = (MQ135Lecture - 150) / (200 - 150);
140.        }
141.          else if (MQ135Lecture >=200  && MQ135Lecture <=300)
142.        {
143.          co2 [1] = 1;
144.        }
145.          else if (MQ135Lecture >300  && MQ135Lecture <=350)
146.        {
147.          co2 [1] = (350 - MQ135Lecture) / (350 - 300);
148.        }
149.        else
150.        {
151.          co2 [1] = 0;
152.        }
153.          // untuk kondisi berbahaya
154.        if (MQ135Lecture <= 300)
155.        {
156.          co2 [2] = 0;
157.        }
158.        else if (MQ135Lecture > 300 && MQ135Lecture <= 350)
159.        {
160.          co2 [2] = (MQ135Lecture - 300) / (350 - 300);
161.        }
162.        else
163.        {
164.          co2 [2] = 1;
165.        }
166.
```

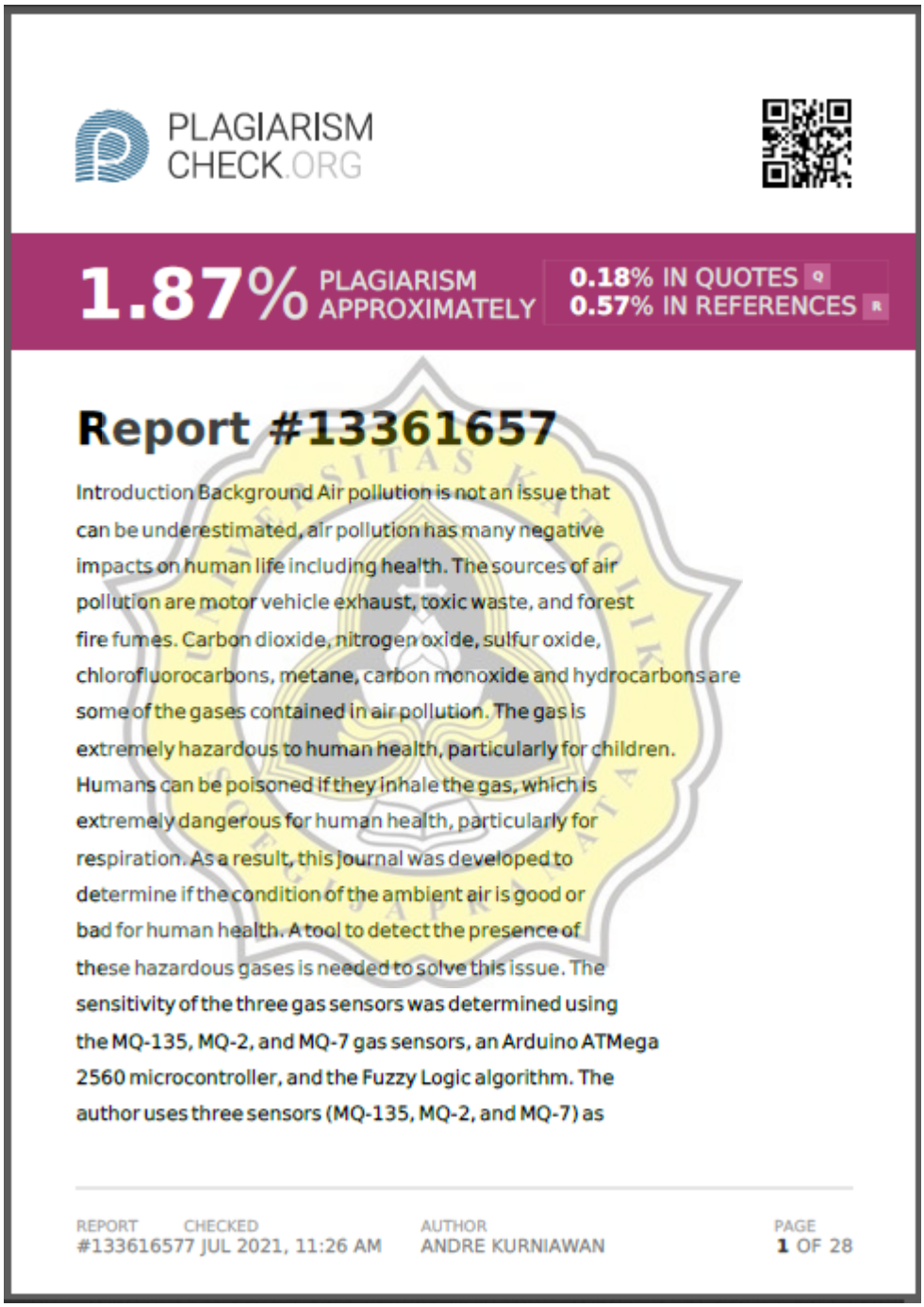
```
167.       Serial.print(" CO2 Baik= ");
168.       Serial.print(co2 [0]);
169.       Serial.print(" CO Sedang= ");
170.       Serial.print(co2 [1]);
171.       Serial.print(" CO2 Berbahaya= ");
172.       Serial.print(co2 [2]);
173.     }
```

CODING RULES

```
174.     void Rules(){
175.      int i, j, k;
176.      int no=1;
177.      for ( i=0; i<=2; i=i+1)
178.      {
179.        for ( j=0; j<=2; j=j+1)
180.        {
181.          for ( k=0; k<=2; k=k+1)
182.          {
183.          temp = min(min(metana[i], co[j]),co2[k]);
184.          rule [i][j][k] = temp;
185.          Serial.print("Rule ke-");
186.          Serial.print(no++);
187.          Serial.print(" : ");
188.          Serial.println( rule [i][j][k]);
189.          }
190.        }
191.      }
192.      rule000 = rule [0][0][0]; // (baik,baik, baik = hijau)
193.      rule001 = rule [0][0][1]; // (baik,baik,sedang = hijau)
194.      rule002 = rule [0][0][2]; // (baik,baik,berbahaya = kuning)
195.      rule012 = rule [0][1][2]; //(baik,sedang,berbahaya = merah)
196.      rule011 = rule [0][1][1]; // (baik,sedang,sedang = kuning)
197.      rule010 = rule [0][1][0]; // (baik,sedang,baik = hijau)
198.      rule020 = rule [0][2][0]; // (baik,berbahaya,baik = kuning)
199.      rule021 = rule [0][2][1]; //(baik,berbahaya,sedang= merah)
200.      rule022 = rule [0][2][2]; // (baik, berbahaya,berbahaya = merah)
201.      rule100 = rule [1][0][0]; // (sedang,baik,baik = hijau)
202.      rule101 = rule [1][0][1]; // (sedang,baik,sedang= kuning )
203.      rule102 = rule [1][0][2]; // (sedang,baik, berbahaya = merah)
204.      rule112 = rule [1][1][2]; // (sedang,sedang,berbahaya = merah)
205.      rule111 = rule [1][1][1]; // (sedang,sedang,sedang = kuning)
206.      rule110 = rule [1][1][0]; //(sedang,sedang,baik = kuning)
207.      rule120 = rule [1][2][0]; // (sedang,berbahaya,baik = merah)
208.      rule121 = rule [1][2][1]; // (sedang,berbahaya,sedang = merah)
209.      rule122 = rule [1][2][2]; // (sedang,berbahaya,berbahaya = merah)
210.      rule200 = rule [2][0][0]; //(berbahaya,baik,baik= kuning)
211.      rule201 = rule [2][0][1]; // (berbahaya, baik,sedang = merah)
212.      rule202 = rule [2][0][2]; // (berbahaya,baik,berbahaya = merah)
213.      rule212 = rule [2][1][2]; // (berbahaya,sedang,berbahaya= merah)
214.      rule211 = rule [2][1][1]; // (berbahaya,sedang,sedang = merah)
215.      rule210 = rule [2][1][0]; //(berbahaya,sedang,baik= merah)
216.      rule220 = rule [2][2][0]; // (berbahaya, berbahaya,baik = merah)
217.      rule221 = rule [2][2][1]; // (berbahaya,berbahaya,sedang = merah)
218.      rule222  =  rule  [2][2][2];  //  (berbahaya,berbahaya,berbahaya=
   merah)
219.     }
```

CODING DEFUZZIFIKASI

```
220.    output1 = (rule000 * Hijau) + (rule001 * Hijau)+ (rule010 * Hijau)+
   (rule100 * Hijau)+ (rule002 * Kuning)+ (rule011 * Kuning) + (rule020 *
   Kuning)+ (rule101 * Kuning)+   (rule111 * Kuning)+ (rule110 * Kuning)+
   (rule200 * Kuning)+ (rule012 * Merah)+ (rule021 * Merah)+ (rule022 *
   Merah)+ (rule102 * Merah)+ (rule112 * Merah)+ (rule120 * Merah)+(rule121
   * Merah)+ (rule122 * Merah)+ (rule201 * Merah)+ (rule202 * Merah)+
   (rule212 * Merah)+ (rule211 * Merah)+ (rule210 * Merah)+ (rule220 *
   Merah)+ (rule221 * Merah)+(rule222 * Merah);
221.     output2 = (rule000 + rule001 + rule010 + rule100 + rule002 +
   rule011 + rule020 + rule101 + rule111 + rule110 + rule200 + rule012 +
   rule021 + rule022 + rule102 + rule112 + rule120 + rule121 + rule122 +
   rule201 + rule202 +  rule212 + rule211 + rule210 + rule220 + rule221 +
   rule222);
222.    hasil_output = output1 / output2;
```



PLAGIARISM CHECK.ORG

1.87% PLAGIARISM APPROXIMATELY

0.18% IN QUOTES Q
0.57% IN REFERENCES R

Report #13361657

Introduction Background Air pollution is not an issue that can be underestimated, air pollution has many negative impacts on human life including health. The sources of air pollution are motor vehicle exhaust, toxic waste, and forest fire fumes. Carbon dioxide, nitrogen oxide, sulfur oxide, chlorofluorocarbons, metane, carbon monoxide and hydrocarbons are some of the gases contained in air pollution. The gas is extremely hazardous to human health, particularly for children. Humans can be poisoned if they inhale the gas, which is extremely dangerous for human health, particularly for respiration. As a result, this journal was developed to determine if the condition of the ambient air is good or bad for human health. A tool to detect the presence of these hazardous gases is needed to solve this issue. The sensitivity of the three gas sensors was determined using the MQ-135, MQ-2, and MQ-7 gas sensors, an Arduino ATMega 2560 microcontroller, and the Fuzzy Logic algorithm. The author uses three sensors (MQ-135, MQ-2, and MQ-7) as

REPORT #133616577 CHECKED JUL 2021, 11:26 AM
AUTHOR ANDRE KURNIAWAN
PAGE **1** OF 28