

IMPLEMENTASI MIKROKONTROLER ARDUINO UNO PADA RANCANG BANGUN ALAT PEMBERSIH FILTER KABIN AC MOBIL OTOMATIS

Sulaeman¹, Muhidin²

Program Studi Teknik Elektro
Universitas Ibn Khaldun Bogor,
Jl. KH. Sholeh Iskandar Km2, Bogor
Email: sulaeman190321@gmail.com

Abstrak

Membersihkan filter AC mobil secara manual dapat mengotori lingkungan sekitar dan berpotensi membahayakan kesehatan karena debu yang bertebaran. Penelitian ini menghasilkan sebuah alat inovatif untuk membersihkan filter kabin AC mobil secara semi otomatis. Rancangan ini menggunakan sensor debu GP2Y1010AU0F untuk mengetahui tingkat kotor filter yang dilengkapi dengan tempat khusus untuk menampung debu hasil pembersihan. Rancangan dilengkapi buzzer dan LED sebagai pemberitahuan proses pembersihan sudah selesai. Dengan menggunakan mikrokontroller arduino menjadikan proses pembersihan lebih efektif karena terdapat sensor debu yang akan mendeteksi kadar debu pada filter kabin AC mobil kotor. Alat pembersih filter kabin AC mobil otomatis terdapat dua bagian, alat pembersih dan kotak kontrol yang jadi satu kesatuan. Program arduino uno dibuat dengan software arduino IDE untuk mengatur sistem pada alat. Terdapat selisih kecepatan angin pada filter kabin AC mobil kotor dan baru berbahan serat sebesar 0.6 m/s. Selisih kecepatan filter dibersihkan dengan filter kotor sebesar 0.4 m/s. Persentasi kebersihan filter kabin berbahan serat menggunakan alat sebesar 67%. Terdapat selisih kecepatan angin pada filter kabin AC mobil kotor dan baru berbahan karbon sebesar 0.4 m/s. Selisih kecepatan filter dibersihkan dengan filter kotor sebesar 0.2 m/s. Persentasi kebersihan filter kabin berbahan karbon menggunakan alat sebesar 50%. Hasil penelitian ini menunjukkan proses pembersihan dari segi tempat lebih efisien. Dengan sistem kontrol, proses pembersihan filter kabin AC mobil juga menjadi lebih efektif. Tingkat kebersihan mampu mencapai 67% menunjukkan bahwa alat ini dapat melakukan pembersihan dengan baik.

Kata kunci: Filter Kabin AC Mobil, Sistem Kontrol, Mikrokontroler Arduino Uno.

Abstract

Cleaning the car's air conditioner filter manually can pollute the surrounding environment and potentially endanger health due to scattered dust. This research has produced an innovative tool to clean the car air conditioning cabin filter semi-automatically. This design uses a GP2Y1010AU0F dust sensor to determine the dirty level of the filter which is equipped with a special place to accommodate the dust from cleaning. The design is equipped with a buzzer and LEDs to notify the cleaning process that it is complete. Using an Arduino microcontroller makes

the cleaning process more effective because there is a dust sensor that will detect the dust level in the car air conditioning cabin filter. The automatic car AC cabin filter cleaner has two parts, a cleaning tool and a control box that are one unit. The arduino uno program is created with the arduino IDE software to set up the system on the device. There is a difference in wind speed in the cabin filter of dirty and new car air conditioners made of fiber is 0.6 m/s. the difference in filter speed is cleaned with a dirty filter of 0.4 m/s. the percentage of cleanliness of the cabin filter made of fiber using a tool is 67%. There is a difference in wind speed in the cabin filter of dirty and new cars made of carbon by 0.4 m/s. The filter speed difference is cleaned with a dirty filter of 0.2 m/s. The cleanliness percentage of the cabin filter made of carbon using a tool is 50%. This research makes the cleaning process in terms of place more efficient. With a control system, the cleaning process of the car's air conditioning cabin filter becomes more effective. The cleanliness level of being able to reach 67% indicates that this appliance can do cleaning well.

Keywords: *Car Air Conditioning Cabin Filter, Control System, Arduino Uno Microcontrol.*

I. LATAR BELAKANG

Teknologi memunculkan beragam inovasi sesuai dengan fungsi dan kebutuhannya, yang menjadikan bagian penting untuk diperhatikan sehingga dengan perkembangan zaman saat ini dapat memenuhi kebutuhan konsumen, dan efisiensi energi [1]. Dalam industri otomotif yang sangat kompetitif, membangun kepercayaan pelanggan menjadi kunci, sehingga kualitas layanan, termasuk jasa servis, memegang peranan penting [2]. Salah satu layanan yang esensial adalah servis air conditioner (AC) mobil. Layanan ini bertujuan untuk mengurangi potensi kerusakan dan menjaga performa sistem AC mobil agar tetap optimal [3].

Penelitian sebelumnya oleh Syamaidzar Shiddiq Ramadhan telah mengembangkan alat pendekripsi kondisi filter kabin AC mobil berbasis arduino. Sistem ini memanfaatkan sensor kecepatan infrared untuk mengklasifikasikan kondisi filter sebagai bersih, sedang, atau kotor [4]. Metode pembersihan filter AC mobil saat ini umumnya melibatkan penggunaan kompresor angin untuk menyemprotkan kotoran ke udara bebas. Proses ini memerlukan operator dan area yang cukup luas agar tidak mengganggu lingkungan sekitar. Oleh karena itu, inovasi diperlukan untuk mengefisienkan proses pembersihan filter kabin AC mobil.

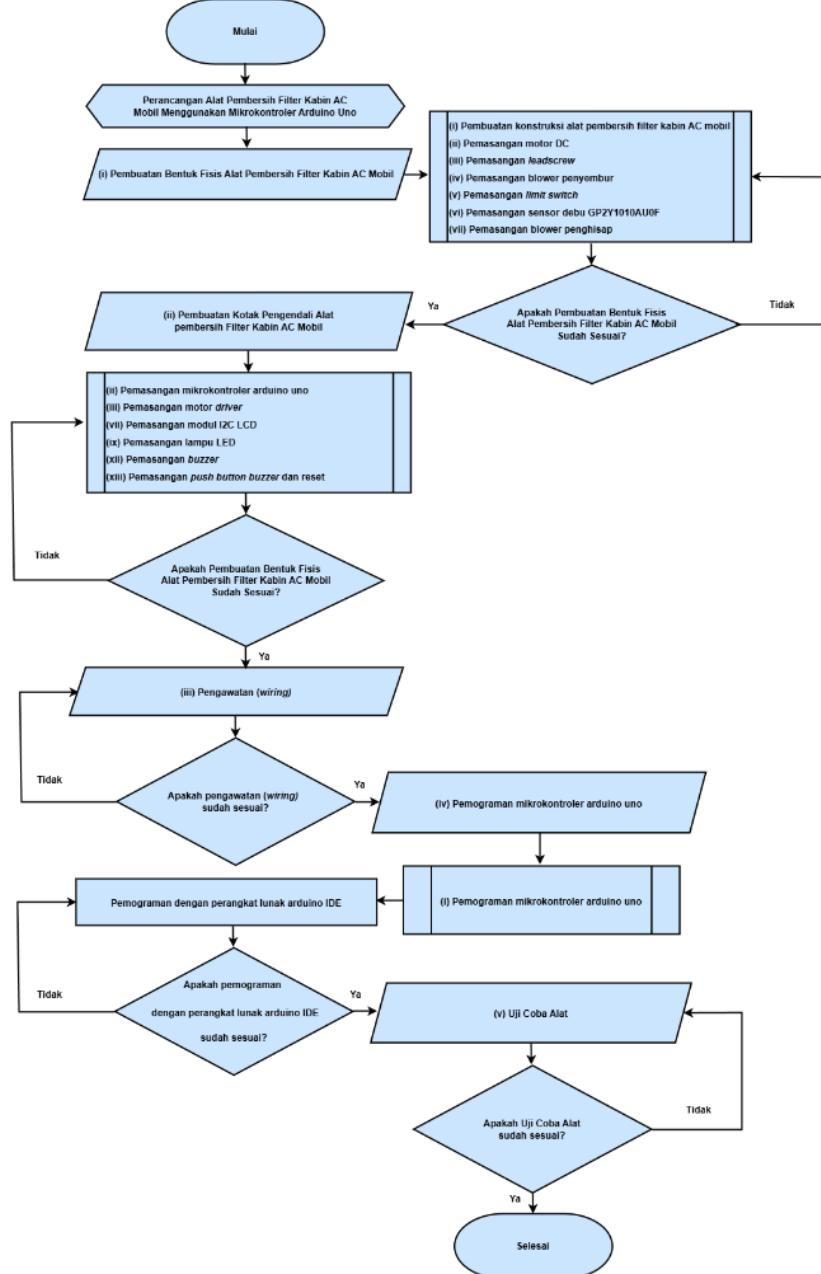
Pembersihan filter kabin AC mobil secara manual memiliki kelemahan, yaitu menghasilkan kotoran yang dapat mengotori lingkungan sekitar dan berpotensi membahayakan kesehatan [5] [6]. Proses ini juga memerlukan alat khusus seperti kompresor angin dan pistol pembersih udara, serta area pembersihan yang sesuai. Untuk mengatasi masalah ini, sistem kontrol menggunakan mikrokontroler arduino uno menjadi solusi yang lebih efektif [7] [8]. Sistem ini memungkinkan pembersihan filter kabin AC mobil tanpa menimbulkan polusi kotoran. Arduino uno berfungsi sebagai otak yang mengontrol berbagai komponen elektronik, memungkinkan sistem untuk memiliki tempat pembersihan yang terintegrasi, mendekripsi tingkat debu maksimal pada filter

kotor, dan melakukan pembersihan secara otomatis [9] [10]. Hal ini akan membuat proses pembersihan filter kabin AC mobil menjadi jauh lebih efisien.

II. METODE PENELITIAN

A. Metode pelaksanaan

Langkah penelitian ini meliputi tahapan-tahapan pelaksanaan penelitian dengan dibuat diagram alur penelitian. Seperti ditunjukkan pada Gambar 1.



Gambar 1. Diagram alur penelitian

Berdasarkan Gambar 1 dapat dijelaskan, bahwa diagram alur penelitian berisi sasaran penelitian dan penjabaran setiap sasaran penelitian.

1) Pembuatan bentuk fisis alat pembersih filter kabin AC mobil

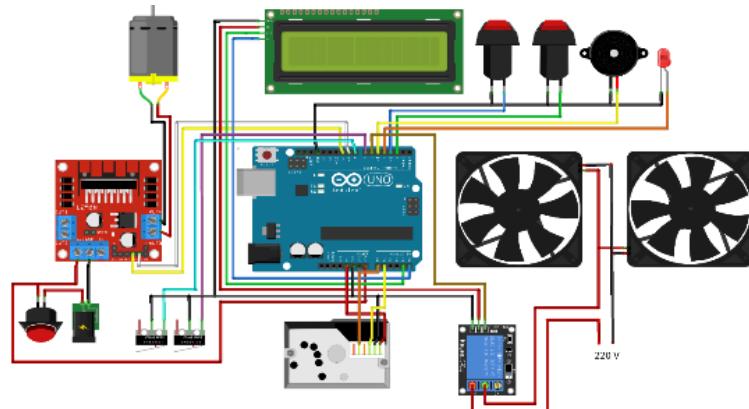
alat pembersih filter AC mobil ini dibuat dari beberapa bagian. Bagian dasarnya menggunakan plat PVC berukuran sedang. Ada juga penyangga dari besi holo yang kuat. Untuk bagian-bagian tertentu, digunakan aluminium tipis dan akrilik tebal yang membentuk wadah khusus untuk filter kabin AC kotor. Di dalam alat ini, ada semacam batang berulir atau *leadscrew* yang digerakkan oleh motor DC. Gerakan batang ini akan menggeser tempat selang penyemprot udara, blower penyemprot, dan blower penghisap. Semua komponen ini diletakkan bersama dalam sebuah kotak sebagai tempatnya. Terakhir, ada kotak kontak listrik untuk menghubungkan blower penyemprot dan penghisap. Jadi, secara keseluruhan, alat ini memiliki bentuk fisik yang dirancang untuk membersihkan filter kabin AC mobil.

2) Pembuatan kotak pengendali alat pembersih filter kabin AC mobil

Kotak pengendali atau panel kontrol alat memiliki tampak depan dengan modul I2C LCD, *buzzer*, *LED*, tombol *buzzer*, tombol reset dan tombol *power*. Pada bagian dalam kotak pengendali atau panel kontrol alat terdapat arduino uno, motor *driver* L298N, *relay*, *power supply switching*.

3) Bentuk pengawatan / *wiring*

Melakukan pengawatan / *wiring* dilakukan agar bentuk fisis alat pembersih filter kabin AC mobil terintegrasi dengan kotak pengendali alat. Bentuk pengawatan / *wiring* ditunjukkan pada Gambar 2.



Gambar 2. Bentuk pengawatan / *wiring*

Berdasarkan Gambar 2 dapat dijelaskan, arduino ini punya banyak kaki atau pin yang terhubung ke berbagai komponen lain. Beberapa kaki digital yaitu D2 sampai D10 Arduino berfungsi untuk menerima perintah dari tombol reset, tombol *buzzer*, dan *limit switch*. Mereka juga mengontrol lampu *LED*, *buzzer* itu sendiri, *relay*, dan motor *driver*. Sementara itu, beberapa kaki analog A0, A1, A4, A5 arduino bertugas membaca sinyal dari sensor debu. Sinyal analog ini kemudian diubah menjadi angka digital dan ditampilkan di layar LCD. Kaki 5V pada arduino memberikan daya

listrik ke sensor debu. Lalu, kaki Vin dan GND adalah jalur utama untuk memberikan daya listrik ke seluruh sistem arduino dan komponen-komponen yang terhubung dengannya. Selain arduino, ada juga tiga sumber listrik atau *power supply*. Yang pertama memberikan tegangan 5V khusus untuk menggerakkan motor *driver* dan motor DC. Yang kedua memberikan tegangan 12V ke arduino supaya bisa berfungsi. Dan yang terakhir, sumber listrik 220V masuk ke *relay* untuk memberikan daya yang besar ke blower penyembur dan blower penghisap agar bisa bekerja. Jadi, arduino uno adalah pusat kontrol yang mengatur semua komponen alat pembersih ini, dibantu oleh berbagai sumber listrik untuk memberikan daya yang sesuai ke masing-masing bagian.

4) Pemograman mikrokontroler arduino uno

Melakukan pemograman mikrokontroler arduino uno dilakukan agar komponen alat yang sudah dilakukan pengawatan dapat terkoneksi dengan mikrokontroler arduino uno untuk dapat diperintahkan bekerja. Terdapat beberapa tahapan seperti pengunduhan *software* arduino IDE, penginstalan *software*, pengunduhan *library*, menginstal model papan *board* pada arduino IDE, mengisntal menu *board* yang diinginkan dan proses *uploading* program.

5) Pengujian alat

Melakukan uji coba alat untuk dapat mengetahui kinerja komponen dapat bekerja serta berfungsi dengan baik dan sebagai pembuktian terhadap objek yang akan dibersihkan bekerja sesuai yang diinginkan. tahapan uji coba alat yang dilakukan seperti pengujian *driver* motor DC, pengujian densitas sensor debu GP2Y1010AU0F, pengujian kinerja *relay*, pengujian kinerja blower penyembur dan blower penghisap, pengujian tombol dan pengujian daya listrik.

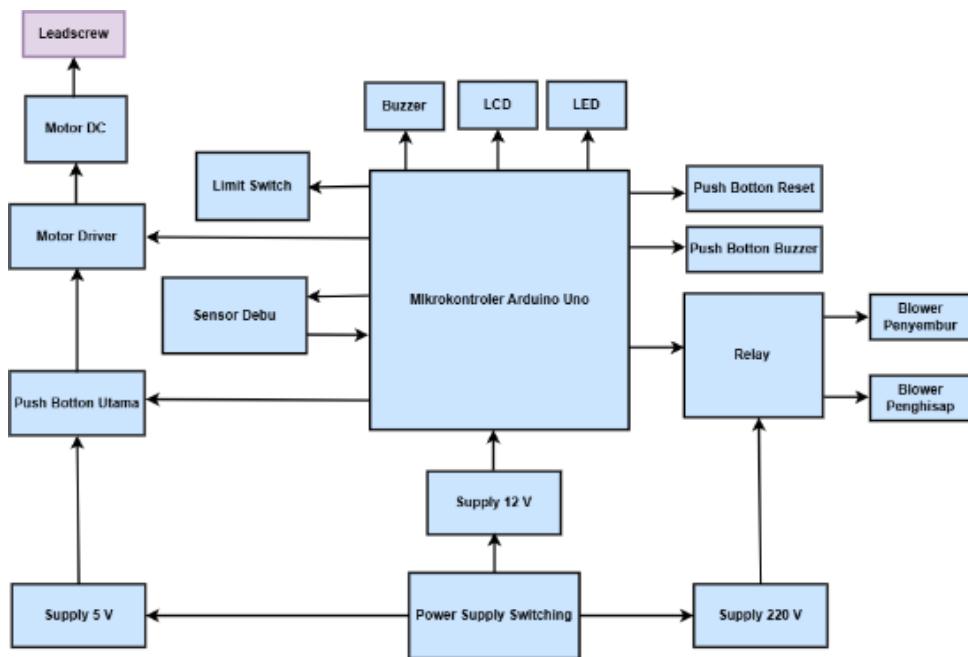
B. Perancangan Kerja Sistem

Perancangan kerja sistem dibuat dengan bentuk blok diagram kerja sistem dan diagram alur kerja sistem, hal ini bertujuan untuk memudahkan dalam proses perancangan sistem kontrol dan kerja sistem dapat bekerja sesuai fungsinya.

1) Blok diagram kerja sistem

Pada setiap blok rangkaian atau sub sistem mempunyai fungsi masing-masing dan dapat beroperasi sesuai dengan tujuan yang diinginkan. Berikut ini blok diagram kerja sistem ditunjukkan pada Gambar 3.

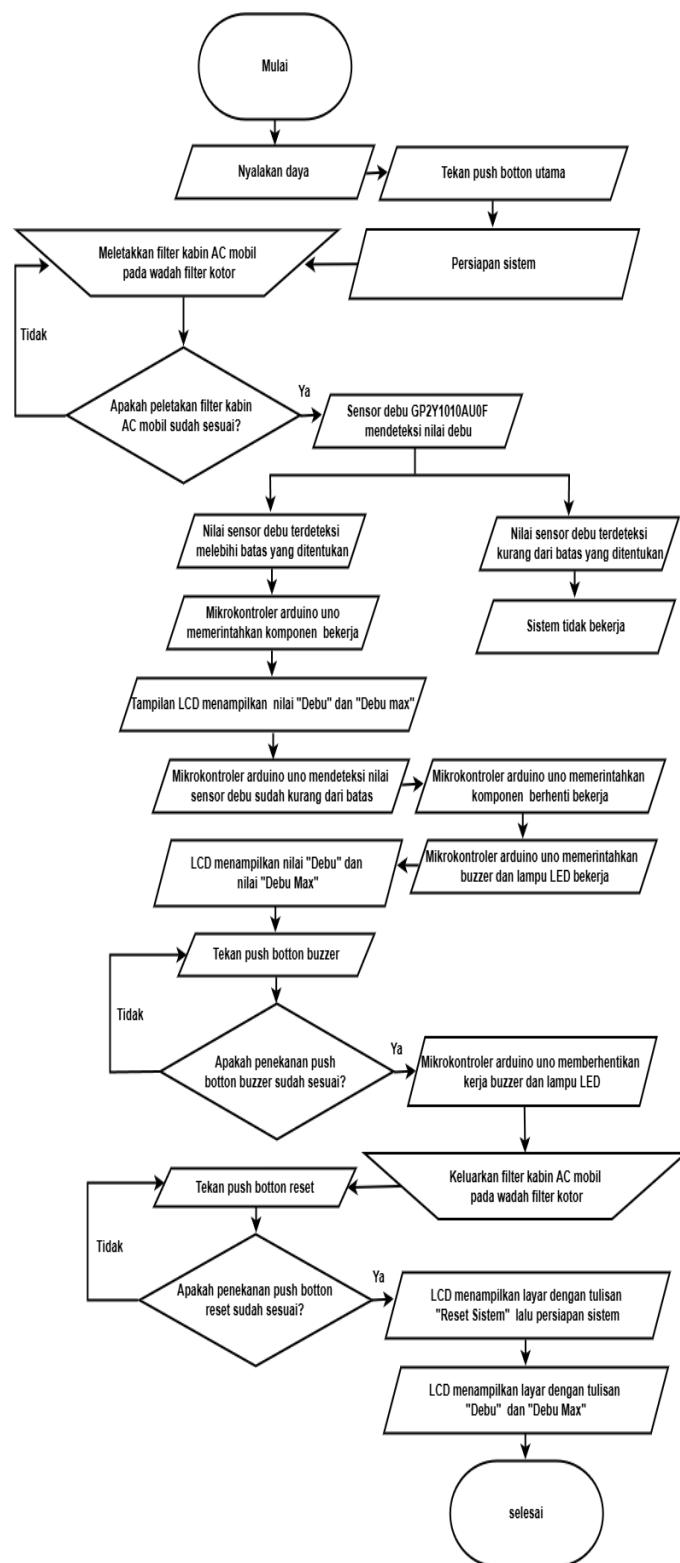
Berdasarkan Gambar 3 dapat dijelaskan, sumber utama listriknya adalah *power supply switching*. *Power supply* 5V memberikan daya ke motor *driver*, sehingga motor DC bisa bergerak. *Power supply* 12V menyalakan mikrokontroler arduino uno. Setelah menyala, arduino akan membaca data dari sensor-sensor. Nilai yang dibaca sensor ini akan diolah oleh arduino dan ditampilkan di layar LCD. Sensor debu juga akan mengirimkan perintah ke arduino sesuai dengan program yang sudah dibuat. Setelah menerima perintah, arduino akan mengaktifkan *relay*. *Relay* ini kemudian menyalakan blower penyembur dan blower penghisap menggunakan daya dari *power supply* 220V. Tombol *buzzer* dan reset yang terhubung ke arduino berfungsi untuk membunyikan *buzzer*, menyalakan LED, dan juga untuk memulai ulang program yang ada di dalam arduino.



Gambar 3. Blok Diagram Kerja Sistem

Jadi, secara keseluruhan, alat ini bekerja dengan menerima daya dari *power supply*, membaca sensor menggunakan arduino, memproses data untuk ditampilkan dan memberikan perintah, lalu mengaktifkan blower untuk membersihkan filter, serta memiliki tombol untuk memberhentikan sinyal suara *buzzer* dan mereset sistem.

- 2) Diagram alur kerja sistem Diagram alur kerja sistem ditunjukkan pada Gambar 4. Berdasarkan Gambar 4 dapat dijelaskan, bahwa alat pembersih berbasis mikrokontroler arduino uno dengan berbantuan sensor debu GP2Y1010AU0F dirancang sebagai sistem pembersih otomatis, terdapat juga *buzzer* dan LED sebagai pemberitahuan alat selesai, serta tombol *buzzer* sebagai pemberhentian suara *buzzer* dan mematikan LED, terdapat juga tombol reset untuk mereset sistem.



Gambar 4. Diagram Alur Kerja Sistem

III. HASIL DAN BAHASAN

A. Alat pembersih filter kabin AC mobil menggunakan mikrokontroler arduino

Alat pembersih filter kabin AC mobil menggunakan mikrokontroler arduino uno memiliki dua bagian, yaitu alat pembersih filter kabin AC mobil dan kotak pengendali atau panel kontrol alat yang menjadi bentuk keseluruhan alat pembersih filter kabin AC mobil menggunakan mikrokontroler arduino uno. Seperti yang ditunjukkan pada Gambar 5.

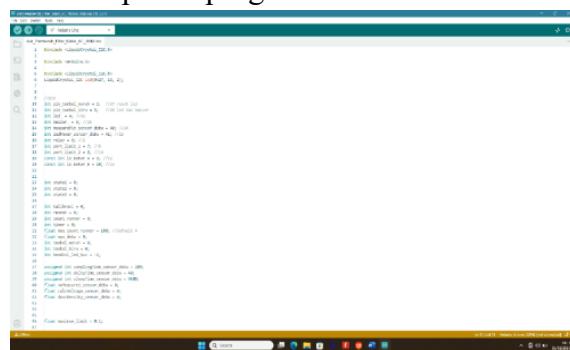


Gambar 5. Alat pembersih filter kabin AC mobil menggunakan mikrokontroler arduino uno

Alat pembersih filter kabin AC mobil bekerja atas perintah mikrokontroler arduino uno yang sudah diprogram, motor *driver* sebagai pengendali gerak motor DC, *power supply switching* sebagai sumber daya dan *push button*, komponen tersebut terdapat pada sistem pengendali atau panel kontrol.

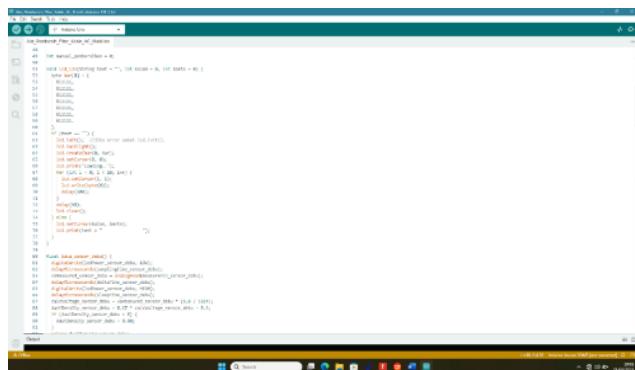
B. Hasil Pemograman Mikrokontroler Arduino Uno

Mikrokontroler arduino uno digunakan untuk mengontrol keseluruhan sistem pada proyek alat ini. Pemograman ini dilakukan menggunakan *software* arduino IDE, program yang di-*coding* pada mikrokontroler arduino uno ini merupakan program sistem kontrol.



Gambar 6. *Library Pin Komponen Dan Penyesuaian Kinerja Sensor Debu*

Gambar 6 dapat dilihat pemrograman ini mengatur arduino uno dan layar LCD I2C. Lalu, ditetapkan pin-pin untuk tombol reset dan *buzzer*, LED, sensor debu, *limit switch*, *relay*, dan *driver* motor. Kinerja sensor debu diatur agar sistem bekerja saat sensor mendeteksi nilai debu $0.5 \mu\text{g}/\text{m}^3$ atau saat *limit switch* tersentuh.

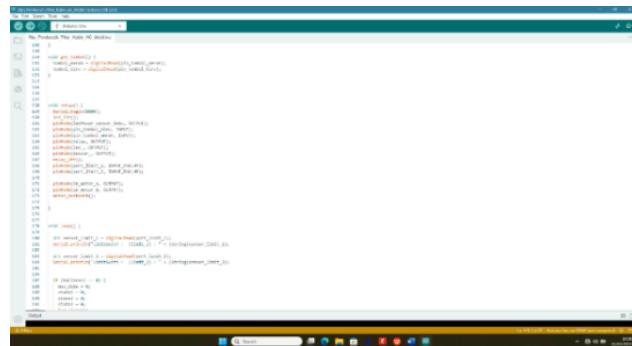


```

40 for(;;delay(1000))
41 {
42     lcd_clear();
43     lcd_string("Debu: ");
44     lcd_string(debu);
45     lcd_string(" %");
46     lcd.setCursor(0,1);
47     lcd_string("Temp: ");
48     lcd_string(temp);
49     lcd_string(" C");
50     lcd.setCursor(0,2);
51     lcd_string("Hum: ");
52     lcd_string(hum);
53     lcd_string(" %");
54 }
55
56 void lcd_clear()
57 {
58     lcd.setCursor(0,0);
59     lcd.print("          ");
60 }
61
62 void lcd_string(char c)
63 {
64     lcd.setCursor(lcd_x, lcd_y);
65     lcd.print(c);
66 }
67
68 void lcd.setCursor(int x, int y)
69 {
70     lcd_x = x;
71     lcd_y = y;
72 }
73
74 void lcd_string(char *str)
75 {
76     lcd.setCursor(0,0);
77     lcd.print(str);
78 }
79
80 void lcd_string(int num)
81 {
82     lcd.setCursor(0,0);
83     lcd.print(num);
84 }
85
86 void lcd_string(float num)
87 {
88     lcd.setCursor(0,0);
89     lcd.print(num);
90 }
91
92 void lcd_string_hex(int num)
93 {
94     lcd.setCursor(0,0);
95     lcd.print(num,HEX);
96 }
97
98 void lcd_string_hex(float num)
99 {
100    lcd.setCursor(0,0);
101    lcd.print(num,HEX);
102 }
103
104 void lcd_string_hex2(int num)
105 {
106    lcd.setCursor(0,0);
107    lcd.print(num,HEX,2);
108 }
109
110 void lcd_string_hex2(float num)
111 {
112    lcd.setCursor(0,0);
113    lcd.print(num,HEX,2);
114 }
115
116 void lcd_string_hex3(int num)
117 {
118    lcd.setCursor(0,0);
119    lcd.print(num,HEX,3);
120 }
121
122 void lcd_string_hex3(float num)
123 {
124    lcd.setCursor(0,0);
125    lcd.print(num,HEX,3);
126 }
127
128 void lcd_string_hex4(int num)
129 {
130    lcd.setCursor(0,0);
131    lcd.print(num,HEX,4);
132 }
133
134 void lcd_string_hex4(float num)
135 {
136    lcd.setCursor(0,0);
137    lcd.print(num,HEX,4);
138 }
139
140 void lcd_string_hex5(int num)
141 {
142    lcd.setCursor(0,0);
143    lcd.print(num,HEX,5);
144 }
145
146 void lcd_string_hex5(float num)
147 {
148    lcd.setCursor(0,0);
149    lcd.print(num,HEX,5);
150 }
151
152 void lcd_string_hex6(int num)
153 {
154    lcd.setCursor(0,0);
155    lcd.print(num,HEX,6);
156 }
157
158 void lcd_string_hex6(float num)
159 {
160    lcd.setCursor(0,0);
161    lcd.print(num,HEX,6);
162 }
163
164 void lcd_string_hex7(int num)
165 {
166    lcd.setCursor(0,0);
167    lcd.print(num,HEX,7);
168 }
169
170 void lcd_string_hex7(float num)
171 {
172    lcd.setCursor(0,0);
173    lcd.print(num,HEX,7);
174 }
175
176 void lcd_string_hex8(int num)
177 {
178    lcd.setCursor(0,0);
179    lcd.print(num,HEX,8);
180 }
181
182 void lcd_string_hex8(float num)
183 {
184    lcd.setCursor(0,0);
185    lcd.print(num,HEX,8);
186 }
187
188 void lcd_string_hex9(int num)
189 {
190    lcd.setCursor(0,0);
191    lcd.print(num,HEX,9);
192 }
193
194 void lcd_string_hex9(float num)
195 {
196    lcd.setCursor(0,0);
197    lcd.print(num,HEX,9);
198 }
199
200 void lcd_string_hex10(int num)
201 {
202    lcd.setCursor(0,0);
203    lcd.print(num,HEX,10);
204 }
205
206 void lcd_string_hex10(float num)
207 {
208    lcd.setCursor(0,0);
209    lcd.print(num,HEX,10);
210 }
211
212 void lcd_string_hex11(int num)
213 {
214    lcd.setCursor(0,0);
215    lcd.print(num,HEX,11);
216 }
217
218 void lcd_string_hex11(float num)
219 {
220    lcd.setCursor(0,0);
221    lcd.print(num,HEX,11);
222 }
223
224 void lcd_string_hex12(int num)
225 {
226    lcd.setCursor(0,0);
227    lcd.print(num,HEX,12);
228 }
229
230 void lcd_string_hex12(float num)
231 {
232    lcd.setCursor(0,0);
233    lcd.print(num,HEX,12);
234 }
235
236 void lcd_string_hex13(int num)
237 {
238    lcd.setCursor(0,0);
239    lcd.print(num,HEX,13);
240 }
241
242 void lcd_string_hex13(float num)
243 {
244    lcd.setCursor(0,0);
245    lcd.print(num,HEX,13);
246 }
247
248 void lcd_string_hex14(int num)
249 {
250    lcd.setCursor(0,0);
251    lcd.print(num,HEX,14);
252 }
253
254 void lcd_string_hex14(float num)
255 {
256    lcd.setCursor(0,0);
257    lcd.print(num,HEX,14);
258 }
259
260 void lcd_string_hex15(int num)
261 {
262    lcd.setCursor(0,0);
263    lcd.print(num,HEX,15);
264 }
265
266 void lcd_string_hex15(float num)
267 {
268    lcd.setCursor(0,0);
269    lcd.print(num,HEX,15);
270 }
271
272 void lcd_string_hex16(int num)
273 {
274    lcd.setCursor(0,0);
275    lcd.print(num,HEX,16);
276 }
277
278 void lcd_string_hex16(float num)
279 {
280    lcd.setCursor(0,0);
281    lcd.print(num,HEX,16);
282 }
283
284 void lcd_string_hex17(int num)
285 {
286    lcd.setCursor(0,0);
287    lcd.print(num,HEX,17);
288 }
289
290 void lcd_string_hex17(float num)
291 {
292    lcd.setCursor(0,0);
293    lcd.print(num,HEX,17);
294 }
295
296 void lcd_string_hex18(int num)
297 {
298    lcd.setCursor(0,0);
299    lcd.print(num,HEX,18);
300 }
301
302 void lcd_string_hex18(float num)
303 {
304    lcd.setCursor(0,0);
305    lcd.print(num,HEX,18);
306 }
307
308 void lcd_string_hex19(int num)
309 {
310    lcd.setCursor(0,0);
311    lcd.print(num,HEX,19);
312 }
313
314 void lcd_string_hex19(float num)
315 {
316    lcd.setCursor(0,0);
317    lcd.print(num,HEX,19);
318 }
319
320 void lcd_string_hex20(int num)
321 {
322    lcd.setCursor(0,0);
323    lcd.print(num,HEX,20);
324 }
325
326 void lcd_string_hex20(float num)
327 {
328    lcd.setCursor(0,0);
329    lcd.print(num,HEX,20);
330 }
331
332 void lcd_string_hex21(int num)
333 {
334    lcd.setCursor(0,0);
335    lcd.print(num,HEX,21);
336 }
337
338 void lcd_string_hex21(float num)
339 {
340    lcd.setCursor(0,0);
341    lcd.print(num,HEX,21);
342 }
343
344 void lcd_string_hex22(int num)
345 {
346    lcd.setCursor(0,0);
347    lcd.print(num,HEX,22);
348 }
349
350 void lcd_string_hex22(float num)
351 {
352    lcd.setCursor(0,0);
353    lcd.print(num,HEX,22);
354 }
355
356 void lcd_string_hex23(int num)
357 {
358    lcd.setCursor(0,0);
359    lcd.print(num,HEX,23);
360 }
361
362 void lcd_string_hex23(float num)
363 {
364    lcd.setCursor(0,0);
365    lcd.print(num,HEX,23);
366 }
367
368 void lcd_string_hex24(int num)
369 {
370    lcd.setCursor(0,0);
371    lcd.print(num,HEX,24);
372 }
373
374 void lcd_string_hex24(float num)
375 {
376    lcd.setCursor(0,0);
377    lcd.print(num,HEX,24);
378 }
379
380 void lcd_string_hex25(int num)
381 {
382    lcd.setCursor(0,0);
383    lcd.print(num,HEX,25);
384 }
385
386 void lcd_string_hex25(float num)
387 {
388    lcd.setCursor(0,0);
389    lcd.print(num,HEX,25);
390 }
391
392 void lcd_string_hex26(int num)
393 {
394    lcd.setCursor(0,0);
395    lcd.print(num,HEX,26);
396 }
397
398 void lcd_string_hex26(float num)
399 {
400    lcd.setCursor(0,0);
401    lcd.print(num,HEX,26);
402 }
403
404 void lcd_string_hex27(int num)
405 {
406    lcd.setCursor(0,0);
407    lcd.print(num,HEX,27);
408 }
409
410 void lcd_string_hex27(float num)
411 {
412    lcd.setCursor(0,0);
413    lcd.print(num,HEX,27);
414 }
415
416 void lcd_string_hex28(int num)
417 {
418    lcd.setCursor(0,0);
419    lcd.print(num,HEX,28);
420 }
421
422 void lcd_string_hex28(float num)
423 {
424    lcd.setCursor(0,0);
425    lcd.print(num,HEX,28);
426 }
427
428 void lcd_string_hex29(int num)
429 {
430    lcd.setCursor(0,0);
431    lcd.print(num,HEX,29);
432 }
433
434 void lcd_string_hex29(float num)
435 {
436    lcd.setCursor(0,0);
437    lcd.print(num,HEX,29);
438 }
439
440 void lcd_string_hex30(int num)
441 {
442    lcd.setCursor(0,0);
443    lcd.print(num,HEX,30);
444 }
445
446 void lcd_string_hex30(float num)
447 {
448    lcd.setCursor(0,0);
449    lcd.print(num,HEX,30);
450 }
451
452 void lcd_string_hex31(int num)
453 {
454    lcd.setCursor(0,0);
455    lcd.print(num,HEX,31);
456 }
457
458 void lcd_string_hex31(float num)
459 {
460    lcd.setCursor(0,0);
461    lcd.print(num,HEX,31);
462 }
463
464 void lcd_string_hex32(int num)
465 {
466    lcd.setCursor(0,0);
467    lcd.print(num,HEX,32);
468 }
469
470 void lcd_string_hex32(float num)
471 {
472    lcd.setCursor(0,0);
473    lcd.print(num,HEX,32);
474 }
475
476 void lcd_string_hex33(int num)
477 {
478    lcd.setCursor(0,0);
479    lcd.print(num,HEX,33);
480 }
481
482 void lcd_string_hex33(float num)
483 {
484    lcd.setCursor(0,0);
485    lcd.print(num,HEX,33);
486 }
487
488 void lcd_string_hex34(int num)
489 {
490    lcd.setCursor(0,0);
491    lcd.print(num,HEX,34);
492 }
493
494 void lcd_string_hex34(float num)
495 {
496    lcd.setCursor(0,0);
497    lcd.print(num,HEX,34);
498 }
499
500 void lcd_string_hex35(int num)
501 {
502    lcd.setCursor(0,0);
503    lcd.print(num,HEX,35);
504 }
505
506 void lcd_string_hex35(float num)
507 {
508    lcd.setCursor(0,0);
509    lcd.print(num,HEX,35);
510 }
511
512 void lcd_string_hex36(int num)
513 {
514    lcd.setCursor(0,0);
515    lcd.print(num,HEX,36);
516 }
517
518 void lcd_string_hex36(float num)
519 {
520    lcd.setCursor(0,0);
521    lcd.print(num,HEX,36);
522 }
523
524 void lcd_string_hex37(int num)
525 {
526    lcd.setCursor(0,0);
527    lcd.print(num,HEX,37);
528 }
529
530 void lcd_string_hex37(float num)
531 {
532    lcd.setCursor(0,0);
533    lcd.print(num,HEX,37);
534 }
535
536 void lcd_string_hex38(int num)
537 {
538    lcd.setCursor(0,0);
539    lcd.print(num,HEX,38);
540 }
541
542 void lcd_string_hex38(float num)
543 {
544    lcd.setCursor(0,0);
545    lcd.print(num,HEX,38);
546 }
547
548 void lcd_string_hex39(int num)
549 {
550    lcd.setCursor(0,0);
551    lcd.print(num,HEX,39);
552 }
553
554 void lcd_string_hex39(float num)
555 {
556    lcd.setCursor(0,0);
557    lcd.print(num,HEX,39);
558 }
559
560 void lcd_string_hex40(int num)
561 {
562    lcd.setCursor(0,0);
563    lcd.print(num,HEX,40);
564 }
565
566 void lcd_string_hex40(float num)
567 {
568    lcd.setCursor(0,0);
569    lcd.print(num,HEX,40);
570 }
571
572 void lcd_string_hex41(int num)
573 {
574    lcd.setCursor(0,0);
575    lcd.print(num,HEX,41);
576 }
577
578 void lcd_string_hex41(float num)
579 {
580    lcd.setCursor(0,0);
581    lcd.print(num,HEX,41);
582 }
583
584 void lcd_string_hex42(int num)
585 {
586    lcd.setCursor(0,0);
587    lcd.print(num,HEX,42);
588 }
589
590 void lcd_string_hex42(float num)
591 {
592    lcd.setCursor(0,0);
593    lcd.print(num,HEX,42);
594 }
595
596 void lcd_string_hex43(int num)
597 {
598    lcd.setCursor(0,0);
599    lcd.print(num,HEX,43);
600 }
601
602 void lcd_string_hex43(float num)
603 {
604    lcd.setCursor(0,0);
605    lcd.print(num,HEX,43);
606 }
607
608 void lcd_string_hex44(int num)
609 {
610    lcd.setCursor(0,0);
611    lcd.print(num,HEX,44);
612 }
613
614 void lcd_string_hex44(float num)
615 {
616    lcd.setCursor(0,0);
617    lcd.print(num,HEX,44);
618 }
619
620 void lcd_string_hex45(int num)
621 {
622    lcd.setCursor(0,0);
623    lcd.print(num,HEX,45);
624 }
625
626 void lcd_string_hex45(float num)
627 {
628    lcd.setCursor(0,0);
629    lcd.print(num,HEX,45);
630 }
631
632 void lcd_string_hex46(int num)
633 {
634    lcd.setCursor(0,0);
635    lcd.print(num,HEX,46);
636 }
637
638 void lcd_string_hex46(float num)
639 {
640    lcd.setCursor(0,0);
641    lcd.print(num,HEX,46);
642 }
643
644 void lcd_string_hex47(int num)
645 {
646    lcd.setCursor(0,0);
647    lcd.print(num,HEX,47);
648 }
649
650 void lcd_string_hex47(float num)
651 {
652    lcd.setCursor(0,0);
653    lcd.print(num,HEX,47);
654 }
655
656 void lcd_string_hex48(int num)
657 {
658    lcd.setCursor(0,0);
659    lcd.print(num,HEX,48);
660 }
661
662 void lcd_string_hex48(float num)
663 {
664    lcd.setCursor(0,0);
665    lcd.print(num,HEX,48);
666 }
667
668 void lcd_string_hex49(int num)
669 {
670    lcd.setCursor(0,0);
671    lcd.print(num,HEX,49);
672 }
673
674 void lcd_string_hex49(float num)
675 {
676    lcd.setCursor(0,0);
677    lcd.print(num,HEX,49);
678 }
679
680 void lcd_string_hex50(int num)
681 {
682    lcd.setCursor(0,0);
683    lcd.print(num,HEX,50);
684 }
685
686 void lcd_string_hex50(float num)
687 {
688    lcd.setCursor(0,0);
689    lcd.print(num,HEX,50);
690 }
691
692 void lcd_string_hex51(int num)
693 {
694    lcd.setCursor(0,0);
695    lcd.print(num,HEX,51);
696 }
697
698 void lcd_string_hex51(float num)
699 {
700    lcd.setCursor(0,0);
701    lcd.print(num,HEX,51);
702 }
703
704 void lcd_string_hex52(int num)
705 {
706    lcd.setCursor(0,0);
707    lcd.print(num,HEX,52);
708 }
709
710 void lcd_string_hex52(float num)
711 {
712    lcd.setCursor(0,0);
713    lcd.print(num,HEX,52);
714 }
715
716 void lcd_string_hex53(int num)
717 {
718    lcd.setCursor(0,0);
719    lcd.print(num,HEX,53);
720 }
721
722 void lcd_string_hex53(float num)
723 {
724    lcd.setCursor(0,0);
725    lcd.print(num,HEX,53);
726 }
727
728 void lcd_string_hex54(int num)
729 {
730    lcd.setCursor(0,0);
731    lcd.print(num,HEX,54);
732 }
733
734 void lcd_string_hex54(float num)
735 {
736    lcd.setCursor(0,0);
737    lcd.print(num,HEX,54);
738 }
739
740 void lcd_string_hex55(int num)
741 {
742    lcd.setCursor(0,0);
743    lcd.print(num,HEX,55);
744 }
745
746 void lcd_string_hex55(float num)
747 {
748    lcd.setCursor(0,0);
749    lcd.print(num,HEX,55);
750 }
751
752 void lcd_string_hex56(int num)
753 {
754    lcd.setCursor(0,0);
755    lcd.print(num,HEX,56);
756 }
757
758 void lcd_string_hex56(float num)
759 {
760    lcd.setCursor(0,0);
761    lcd.print(num,HEX,56);
762 }
763
764 void lcd_string_hex57(int num)
765 {
766    lcd.setCursor(0,0);
767    lcd.print(num,HEX,57);
768 }
769
770 void lcd_string_hex57(float num)
771 {
772    lcd.setCursor(0,0);
773    lcd.print(num,HEX,57);
774 }
775
776 void lcd_string_hex58(int num)
777 {
778    lcd.setCursor(0,0);
779    lcd.print(num,HEX,58);
780 }
781
782 void lcd_string_hex58(float num)
783 {
784    lcd.setCursor(0,0);
785    lcd.print(num,HEX,58);
786 }
787
788 void lcd_string_hex59(int num)
789 {
790    lcd.setCursor(0,0);
791    lcd.print(num,HEX,59);
792 }
793
794 void lcd_string_hex59(float num)
795 {
796    lcd.setCursor(0,0);
797    lcd.print(num,HEX,59);
798 }
799
800 void lcd_string_hex60(int num)
801 {
802    lcd.setCursor(0,0);
803    lcd.print(num,HEX,60);
804 }
805
806 void lcd_string_hex60(float num)
807 {
808    lcd.setCursor(0,0);
809    lcd.print(num,HEX,60);
810 }
811
812 void lcd_string_hex61(int num)
813 {
814    lcd.setCursor(0,0);
815    lcd.print(num,HEX,61);
816 }
817
818 void lcd_string_hex61(float num)
819 {
820    lcd.setCursor(0,0);
821    lcd.print(num,HEX,61);
822 }
823
824 void lcd_string_hex62(int num)
825 {
826    lcd.setCursor(0,0);
827    lcd.print(num,HEX,62);
828 }
829
830 void lcd_string_hex62(float num)
831 {
832    lcd.setCursor(0,0);
833    lcd.print(num,HEX,62);
834 }
835
836 void lcd_string_hex63(int num)
837 {
838    lcd.setCursor(0,0);
839    lcd.print(num,HEX,63);
840 }
841
842 void lcd_string_hex63(float num)
843 {
844    lcd.setCursor(0,0);
845    lcd.print(num,HEX,63);
846 }
847
848 void lcd_string_hex64(int num)
849 {
850    lcd.setCursor(0,0);
851    lcd.print(num,HEX,64);
852 }
853
854 void lcd_string_hex64(float num)
855 {
856    lcd.setCursor(0,0);
857    lcd.print(num,HEX,64);
858 }
859
860 void lcd_string_hex65(int num)
861 {
862    lcd.setCursor(0,0);
863    lcd.print(num,HEX,65);
864 }
865
866 void lcd_string_hex65(float num)
867 {
868    lcd.setCursor(0,0);
869    lcd.print(num,HEX,65);
870 }
871
872 void lcd_string_hex66(int num)
873 {
874    lcd.setCursor(0,0);
875    lcd.print(num,HEX,66);
876 }
877
878 void lcd_string_hex66(float num)
879 {
880    lcd.setCursor(0,0);
881    lcd.print(num,HEX,66);
882 }
883
884 void lcd_string_hex67(int num)
885 {
886    lcd.setCursor(0,0);
887    lcd.print(num,HEX,67);
888 }
889
890 void lcd_string_hex67(float num)
891 {
892    lcd.setCursor(0,0);
893    lcd.print(num,HEX,67);
894 }
895
896 void lcd_string_hex68(int num)
897 {
898    lcd.setCursor(0,0);
899    lcd.print(num,HEX,68);
900 }
901
902 void lcd_string_hex68(float num)
903 {
904    lcd.setCursor(0,0);
905    lcd.print(num,HEX,68);
906 }
907
908 void lcd_string_hex69(int num)
909 {
910    lcd.setCursor(0,0);
911    lcd.print(num,HEX,69);
912 }
913
914 void lcd_string_hex69(float num)
915 {
916    lcd.setCursor(0,0);
917    lcd.print(num,HEX,69);
918 }
919
920 void lcd_string_hex70(int num)
921 {
922    lcd.setCursor(0,0);
923    lcd.print(num,HEX,70);
924 }
925
926 void lcd_string_hex70(float num)
927 {
928    lcd.setCursor(0,0);
929    lcd.print(num,HEX,70);
930 }
931
932 void lcd_string_hex71(int num)
933 {
934    lcd.setCursor(0,0);
935    lcd.print(num,HEX,71);
936 }
937
938 void lcd_string_hex71(float num)
939 {
940    lcd.setCursor(0,0);
941    lcd.print(num,HEX,71);
942 }
943
944 void lcd_string_hex72(int num)
945 {
946    lcd.setCursor(0,0);
947    lcd.print(num,HEX,72);
948 }
949
950 void lcd_string_hex72(float num)
951 {
952    lcd.setCursor(0,0);
953    lcd.print(num,HEX,72);
954 }
955
956 void lcd_string_hex73(int num)
957 {
958    lcd.setCursor(0,0);
959    lcd.print(num,HEX,73);
960 }
961
962 void lcd_string_hex73(float num)
963 {
964    lcd.setCursor(0,0);
965    lcd.print(num,HEX,73);
966 }
967
968 void lcd_string_hex74(int num)
969 {
970    lcd.setCursor(0,0);
971    lcd.print(num,HEX,74);
972 }
973
974 void lcd_string_hex74(float num)
975 {
976    lcd.setCursor(0,0);
977    lcd.print(num,HEX,74);
978 }
979
980 void lcd_string_hex75(int num)
981 {
982    lcd.setCursor(0,0);
983    lcd.print(num,HEX,75);
984 }
985
986 void lcd_string_hex75(float num)
987 {
988    lcd.setCursor(0,0);
989    lcd.print(num,HEX,75);
990 }
991
992 void lcd_string_hex76(int num)
993 {
994    lcd.setCursor(0,0);
995    lcd.print(num,HEX,76);
996 }
997
998 void lcd_string_hex76(float num)
999 {
1000    lcd.setCursor(0,0);
1001    lcd.print(num,HEX,76);
1002 }
1003
1004 void lcd_string_hex77(int num)
1005 {
1006    lcd.setCursor(0,0);
1007    lcd.print(num,HEX,77);
1008 }
1009
1010 void lcd_string_hex77(float num)
1011 {
1012    lcd.setCursor(0,0);
1013    lcd.print(num,HEX,77);
1014 }
1015
1016 void lcd_string_hex78(int num)
1017 {
1018    lcd.setCursor(0,0);
1019    lcd.print(num,HEX,78);
1020 }
1021
1022 void lcd_string_hex78(float num)
1023 {
1024    lcd.setCursor(0,0);
1025    lcd.print(num,HEX,78);
1026 }
1027
1028 void lcd_string_hex79(int num)
1029 {
1030    lcd.setCursor(0,0);
1031    lcd.print(num,HEX,79);
1032 }
1033
1034 void lcd_string_hex79(float num)
1035 {
1036    lcd.setCursor(0,0);
1037    lcd.print(num,HEX,79);
1038 }
1039
1040 void lcd_string_hex80(int num)
1041 {
1042    lcd.setCursor(0,0);
1043    lcd.print(num,HEX,80);
1044 }
1045
1046 void lcd_string_hex80(float num)
1047 {
1048    lcd.setCursor(0,0);
1049    lcd.print(num,HEX,80);
1050 }
1051
1052 void lcd_string_hex81(int num)
1053 {
1054    lcd.setCursor(0,0);
1055    lcd.print(num,HEX,81);
1056 }
1057
1058 void lcd_string_hex81(float num)
1059 {
1060    lcd.setCursor(0,0);
1061    lcd.print(num,HEX,81);
1062 }
1063
1064 void lcd_string_hex82(int num)
1065 {
1066    lcd.setCursor(0,0);
1067    lcd.print(num,HEX,82);
1068 }
1069
1070 void lcd_string_hex82(float num)
1071 {
1072    lcd.setCursor(0,0);
1073    lcd.print(num,HEX,82);
1074 }
1075
1076 void lcd_string_hex83(int num)
1077 {
1078    lcd.setCursor(0,0);
1079    lcd.print(num,HEX,83);
1080 }
1081
1082 void lcd_string_hex83(float num)
1083 {
1084    lcd.setCursor(0,0);
1085    lcd.print(num,HEX,83);
1086 }
1087
1088 void lcd_string_hex84(int num)
1089 {
1090    lcd.setCursor(0,0);
1091    lcd.print(num,HEX,84);
1092 }
1093
1094 void lcd_string_hex84(float num)
1095 {
1096    lcd.setCursor(0,0);
1097    lcd.print(num,HEX,84);
1098 }
1099
1100 void lcd_string_hex85(int num)
1101 {
1102    lcd.setCursor(0,0);
1103    lcd.print(num,HEX,85);
1104 }
1105
1106 void lcd_string_hex85(float num)
1107 {
1108    lcd.setCursor(0,0);
1109    lcd.print(num,HEX,85);
1110 }
1111
1112 void lcd_string_hex86(int num)
1113 {
1114    lcd.setCursor(0,0);
1115    lcd.print(num,HEX,86);
1116 }
1117
1118 void lcd_string_hex86(float num)
1119 {
1120    lcd.setCursor(0,0);
1121    lcd.print(num,HEX,86);
1122 }
1123
1124 void lcd_string_hex87(int num)
1125 {
1126    lcd.setCursor(0,0);
1127    lcd.print(num,HEX,87);
1128 }
1129
1130 void lcd_string_hex87(float num)
1131 {
1132    lcd.setCursor(0,0);
1133    lcd.print(num,HEX,87);
1134 }
1135
1136 void lcd_string_hex88(int num)
1137 {
1138    lcd.setCursor(0,0);
1139    lcd.print(num,HEX,88);
1140 }
1141
1142 void lcd_string_hex88(float num)
1143 {
1144    lcd.setCursor(0,0);
1145    lcd.print(num,HEX,88);
1146 }
1147
1148 void lcd_string_hex89(int num)
1149 {
1150    lcd.setCursor(0,0);
1151    lcd.print(num,HEX,89);
1152 }
1153
1154 void lcd_string_hex89(float num)
1155 {
1156    lcd.setCursor(0,0);
1157    lcd.print(num,HEX,89);
1158 }
1159
1160 void lcd_string_hex90(int num)
1161 {
1162    lcd.setCursor(0,0);
1163    lcd.print(num,HEX,90);
1164 }
1165
1166 void lcd_string_hex90(float num)
1167 {
1168    lcd.setCursor(0,0);
1169    lcd.print(num,HEX,90);
1170 }
1171
1172 void lcd_string_hex91(int num)
1173 {
1174    lcd.setCursor(0,0);
1175    lcd.print(num,HEX,91);
1176 }
1177
1178 void lcd_string_hex91(float num)
1179 {
1180    lcd.setCursor(0,0);
1181    lcd.print(num,HEX,91);
1182 }
1183
1184 void lcd_string_hex92(int num)
1185 {
1186    lcd.setCursor(0,0);
1187    lcd.print(num,HEX,92);
1188 }
1189
1190 void lcd_string_hex92(float num)
1191 {
1192    lcd.setCursor(0,0);
1193    lcd.print(num,HEX,92);
1194 }
1195
1196 void lcd_string_hex93(int num)
1197 {
1198    lcd.setCursor(0,0);
1199    lcd.print(num,HEX,93);
1200 }
1201
1202 void lcd_string_hex93(float num)
1203 {
1204    lcd.setCursor(0,0);
1205    lcd.print(num,HEX,93);
1206 }
1207
1208 void lcd_string_hex94(int num)
1209 {
1210    lcd.setCursor(0,0);
1211    lcd.print(num,HEX,94);
1212 }
1213
1214 void lcd_string_hex94(float num)
1215 {
1216    lcd.setCursor(0,0);
1217    lcd.print(num,HEX,94);
1218 }
1219
1220 void lcd_string_hex95(int num)
1221 {
1222    lcd.setCursor(0,0);
1223    lcd.print(num,HEX,95);
1224 }
1225
1226 void lcd_string_hex95(float num)
1227 {
1228    lcd.setCursor(0,0);
1229    lcd.print(num,HEX,95);
1230 }
1231
1232 void lcd_string_hex96(int num)
1233 {
1234    lcd.setCursor(0,0);
1235    lcd.print(num,HEX,96);
1236 }
1237
1238 void lcd_string_hex96(float num)
1239 {
1240    lcd.setCursor(0,0);
1241    lcd.print(num,HEX,96);
1242 }
1243
1244 void lcd_string_hex97(int num)
1245 {
1246    lcd.setCursor(0,0);
1247    lcd.print(num,HEX,97);
1248 }
1249
1250 void lcd_string_hex97(float num)
1251 {
1252    lcd.setCursor(0,0);
1253    lcd.print(num,HEX,97);
1254 }
1255
1256 void lcd_string_hex98(int num)
1257 {
1258    lcd.setCursor(0,0);
1259    lcd.print(num,HEX,98);
1260 }
1261
1262 void lcd_string_hex98(float num)
1263 {
1264    lcd.setCursor(0,0);
1265    lcd.print(num,HEX,98);
1266 }
1267
1268 void lcd_string_hex99(int num)
1269 {
1270    lcd.setCursor(0,0);
1271    lcd.print(num,HEX,99);
1272 }
1273
1274 void lcd_string_hex99(float num)
1275 {
1276    lcd.setCursor(0,0);
1277    lcd.print(num,HEX,99);
1278 }
1279
1280 void lcd_string_hex100(int num)
1281 {
1282    lcd.setCursor(0,0);
1283    lcd.print(num,HEX,100);
1284 }
1285
1286 void lcd_string_hex100(float num)
1287 {
1288    lcd.setCursor(0,0);
1289    lcd.print(num,HEX,100);
1290 }
1291
1292 void lcd_string_hex101(int num)
1293 {
1294    lcd.setCursor(0,0);
1295    lcd.print(num,HEX,101);
1296 }
1297
1298 void lcd_string_hex101(float num)
1299 {
1300    lcd.setCursor(0,0);
1301    lcd.print(num,HEX,101);
1302 }
1303
1304 void lcd_string_hex102(int num)
1305 {
1306    lcd.setCursor(0,0);
1307    lcd.print(num,HEX,102);
1308 }
1309
1310 void lcd_string_hex102(float num)
1311 {
1312    lcd.setCursor(0,0);
1313    lcd.print(num,HEX,102);
1314 }
1315
1316 void lcd_string_hex103(int num)
1317 {
1318    lcd.setCursor(0,0);
1319    lcd.print(num,HEX,103);
1320 }
1321
1322 void lcd_string_hex103(float num)
1323 {
1324    lcd.setCursor(0,0);
1325    lcd.print(num,HEX,103);
1326 }
1327
1328 void lcd_string_hex104(int num)
1329 {
1330    lcd.setCursor(0,0);
1331    lcd.print(num,HEX,104);
1332 }
1333
1334 void lcd_string_hex104(float num)
1335 {
1336    lcd.setCursor(0,0);
1337    lcd.print(num,HEX,104);
1338 }
1339
1340 void lcd_string_hex105(int num)
1341 {
1342    lcd.setCursor(0,0);
1343    lcd.print(num,HEX,105);
1344 }
1345
1346 void lcd_string_hex105(float num)
1347 {
1348    lcd.setCursor(0,0);
1349    lcd.print(num,HEX,105);
1350 }
1351
1352 void lcd_string_hex106(int num)
1353 {
1354    lcd.setCursor(0,0);
1355    lcd.print(num,HEX,106);
1356 }
1357
1358 void lcd_string_hex106(float num)
1359 {
1360    lcd.setCursor(0,0);
1361    lcd.print(num,HEX,106);
1362 }
1363
1364 void lcd_string_hex107(int num)
1365 {
1366    lcd.setCursor(0,0);
1367    lcd.print(num,HEX,107);
1368 }
136
```

PROGRAM STUDI TEKNIK ELEKTRO FTSUika-BOGOR

Gambar 9 menghubungkan tombol *push button* pada komponen sistem LCD I2C, *relay*, *buzzer* dan *LED*, *limit switch* dan motor *driver*.



Gambar 10. Menghubungkan LCD I2c dan *push button* *buzzer*

Gambar 10 menghubungkan LCD I2C pada perintah tombol *push button* *buzzer* dengan cara menyesuaikan perintah kerja sensor debu, *limit switch* dan motor *driver*.



Gambar 11. Menghubungkan tombol *push button* reset

Gambar 11 menghubungkan tombol *push button* reset dengan cara mengembalikan nilai sensor debu dan tampilan LCD I2C.



Gambar 12. Mengembalikan kinerja sensor debu

Gambar 12 Mengembalikan kinerja sensor debu dilakukan agar sistem dapat melakukan pendekripsi nilai sensor debu sehingga dapat melakukan pembersihan kembali.

C. Pengujian alat pada filter kabin AC mobil

Pengujian ini bertujuan untuk melihat bagaimana alat pembersih filter AC mobil yang dikendalikan arduino uno bekerja saat membersihkan filter AC mobil yang kotor. Digunakan anemometer sebagai instrumen alat ukur kecepatan angin agar mengetahui kecepatan pada filter kabin AC mobil. Ada dua jenis filter kotor yang diuji yaitu filter berbahan serat dan filter berbahan karbon.

1) Pengujian filter kabin AC mobil berbahan serat

Kecepatan angin dari ventilasi AC diukur menggunakan anemometer digital. Untuk filter kabin AC mobil kotor berbahan serat, kecepatan anginnya adalah 0.8 m/s. Untuk mengetahui kecepatan angin maksimal pada filter berbahan serat, pengukuran juga dilakukan pada filter kabin AC mobil yang baru berbahan serat kecepatan anginnya adalah 1.4 m/s. perbandingan kecepatan angin pada filter kabin AC mobil kotor dan filter kabin AC mobil baru berbahan serat memiliki selisih 0.6 m/s. Seperti yang ditunjukkan pada Tabel 1.

Tabel 1. Selisih kecepatan angin pada filter kabin AC mobil kotor dan baru berbahan serat

Kondisi Filter	Kecepatan Ventilasi AC (m/s)	Selisih Kecepatan (m/s)
Kotor	0.8	
Dibersihkan	1.4	0.6

Setelah filter kabin AC mobil kotor berbahan serat dibersihkan menggunakan alat pembersih filter kabin AC mobil menggunakan mikrokontroler arduino uno, kecepatan angin pada ventilasi AC kembali diukur. Hasilnya menunjukkan kecepatan angin menjadi 1.2 m/s. selisih kecepatan filter dibersihkan dengan filter kotor sebesar 0.4 m/s. Seperti yang ditunjukkan pada Tabel 2.

Tabel 2. Selisih kecepatan angin filter kabin AC mobil kotor dan dibersihkan berbahan serat

Kondisi Filter	Kecepatan Ventilasi AC (m/s)	Selisih Kecepatan (m/s)
Kotor	0.8	
Dibersihkan	1.2	0.4

Percobaan ini, filter kabin AC mobil berbahan serat dengan pembersihan menggunakan alat pembersih filter kabin AC mobil menggunakan mikrokontroler arduino uno terdapat persentase kebersihan filter sebesar 67%. Menandakan alat bisa bekerja sesuai dengan baik.

2) Pengujian filter kabin AC mobil berbahan karbon

Kecepatan angin dari ventilasi AC diukur menggunakan anemometer digital. Untuk filter kabin AC mobil kotor berbahan karbon, kecepatan anginnya adalah 1.4 m/s. Untuk mengetahui kecepatan angin maksimal pada filter berbahan serat, pengukuran juga dilakukan pada filter

kabin AC mobil yang baru berbahan serat kecepatan anginnya adalah 1.8 m/s . perbandingan kecepatan angin pada filter kabin AC mobil kotor dan filter kabin AC mobil baru berbahan karbon memiliki selisih 0.4 m/s. seperti yang ditunjukkan pada Tabel 3.

Tabel 3. Selisih kecepatan angin pada filter kabin AC mobil kotor dan baru berbahan karbon

Kondisi Filter	Kecepatan Ventilasi AC (m/s)	Selisih Kecepatan (m/s)
Kotor	1.4	
Dibersihkan	1.8	0.4

Setelah filter kabin AC mobil kotor berbahan karbon dibersihkan menggunakan alat pembersih filter kabin AC mobil menggunakan mikrokontroler arduino uno, kecepatan angin pada ventilasi AC kembali diukur. Hasilnya menunjukkan kecepatan angin menjadi 1.6 m/s. selisih kecepatan filter dibersihkan dengan filter kotor sebesar 0.2 m/s. Seperti yang ditunjukkan pada Tabel 4.

Tabel 4. Selisih kecepatan angin filter kabin AC mobil kotor dan dibersihkan berbahan karbon

Kondisi Filter	Kecepatan Ventilasi AC (m/s)	Selisih Kecepatan (m/s)
Kotor	1.4	
Dibersihkan	1.6	0.2

Percobaan ini, filter kabin AC berbahan karbon dengan pembersihan menggunakan alat pembersih filter kabin AC mobil menggunakan mikrokontroler arduino uno terdapat persentase kebersihan filter sebesar 50%.

Tabel 5. Hasil percobaan pembersihan filter kabin AC mobil berbahan serat dan berbahan karbon

Hasil	Bahan Serat	Bahan Karbon
Selisih Filter Kotor Dan Dibersihkan (m/s)	0.4	0.2
Selisih Filter Kotor Dan Baru (m/s)	0.6	0.4
Persentase Kebersihan Filter (%)	67	50

Berdasarkan table 5 diatas, dapat diketahui bahwa hasil percobaan pembersihan filter AC mobil kotor berbahan serat dan filter kabin AC mobil kotor berbahan karbon dengan pembersihan menggunakan alat pembersih filter kabin AC mobil menggunakan mikrokontroler arduino menunjukkan hasil yang berbeda. Perbedaan ini dipengaruhi oleh jenis filter, kotoran yang

menempel, kondisi komponen AC seperti blower maupun evaporator, dan desain ventilasi. Hasil percobaan menunjukkan tingkat kebersihan filter serat mencapai 67%, sedangkan filter karbon hanya 50%. Ini berarti alat pembersih lebih efektif membersihkan filter kabin AC mobil yang berbahan serat.

IV. KESIMPULAN

Berdasarkan pengujian, dapat disimpulkan bahwa alat pembersih filter kabin AC mobil berbasis arduino uno berhasil dibuat dan berfungsi dengan baik. Alat ini mampu menghisap kotoran ke wadah penampung, memberikan notifikasi selesai melalui *buzzer* dan LED, menjangkau seluruh bagian filter kotor dengan motor DC, dan bekerja otomatis dengan sensor debu. Tingkat keberhasilan pembersihan terukur melalui peningkatan kecepatan angin pada ventilasi AC. Dalam percobaan ini, tingkat kebersihan mampu mencapai 67% pada filter kabin AC mobil berbahan serat, ini berarti alat pembersih filter kabin AC mobil menggunakan mikrokontroler arduino uno lebih efektif membersihkan filter kabin AC mobil yang berbahan serat dan menunjukkan bahwa alat ini dapat melakukan pembersihan dengan baik.

V. REFERENSI

- [1] A. Z. Yusuf, and Y. Yasdin, "Analisis Desain Media Pembelajaran Alat Test Dan Pembersih Injektor Sepeda Motor," *Jurnal Pendidikan Vokasi Otomotif*, vol. 6, pp. 61-70, 2023.
- [2] M. I. Mukhlis, "Pengaruh Kualitas Pelayanan, Harga, Dan Citra Perusahaan Terhadap Keputusan Penggunaan Jasa Service Pada Pt. Astra International Tbk-Bmw Cabang Surabaya," *Soetomo Management Review*, vol. 1, pp. 471-486, 2023.
- [3] J. Jamaludin, "Rancang Bangun Simulator Sistem Air Condition Pada Mobil Type Sanden," *Jurnal Teknik Mesin*, vol. 3, 2020.
- [4] R. S. Shiddiq, "Rancang Bangun Alat Pendekripsi Filter Air Conditioner (AC) Mobil Berbasis Arduino," Doctoral dissertation, Politeknik Keselamatan Transportasi Jalan, 2021.
- [5] B. Afif, "Dampak Pencemaran Udara Pada Lingkungan," *Berita Dirgantara*, vol. 2, 2010.
- [6] F. Muliawati, and A. Seftiana, "Prototipe Sistem Otomatis Pengukuran Densitas Debu, Kelembaban Udara, Dan Suhu Ruang Berbasis Mikrokontroler Atmega 32 Untuk Sterilisasi Udara Pada Ruang Perakitan Lensa Kamera," *Jurnal Teknik Elektro Fakultas Teknik Universitas Ibn Khaldun Bogor*, vol. 3, pp. 18-25, 2016.
- [7] M. A. Saputra, "Rancang Bangun Modul Ac Pengatur Suhu Otomatis Berbasis Arduino Mega 2560," *Jurnal Teknik Elektro Fakultas Teknik Universitas Ibn Khaldun Bogor*, vol. 8, pp. 13-1, 2021.
- [8] G. A. Saputra, and R. Endra, "Analisis Cara Kerja Sensor Ph-E4502c Menggunakan Mikrokontroler Arduino Uno Untuk Merancang Alat Pengendalian Ph Air Pada Tambak," no. December, vol. 1, pp. 1-45, 2020.

- [9] H. F. Nur, , I. A. P. Purwanti, and D. T. Sugiarto, “Perancangan Produk Helm Pintar Penyandang Tunanetra Sistem Navigasi Berbasis Suara Dan Aplikasi Mobile Asisten,” Journal of Industrial Engineering and Technology, vol. 2, pp. 140-152. 2022.
- [10] A. Abdulrahman, and O. Mustopa, “Prototipe Sistem Pencampur Ragi Otomatis Berbasis Arduino Uno,” Jurnal Teknik Elektro Fakultas Teknik Universitas Ibn Khaldun Bogor, vol. 7, no. 1, 2020.