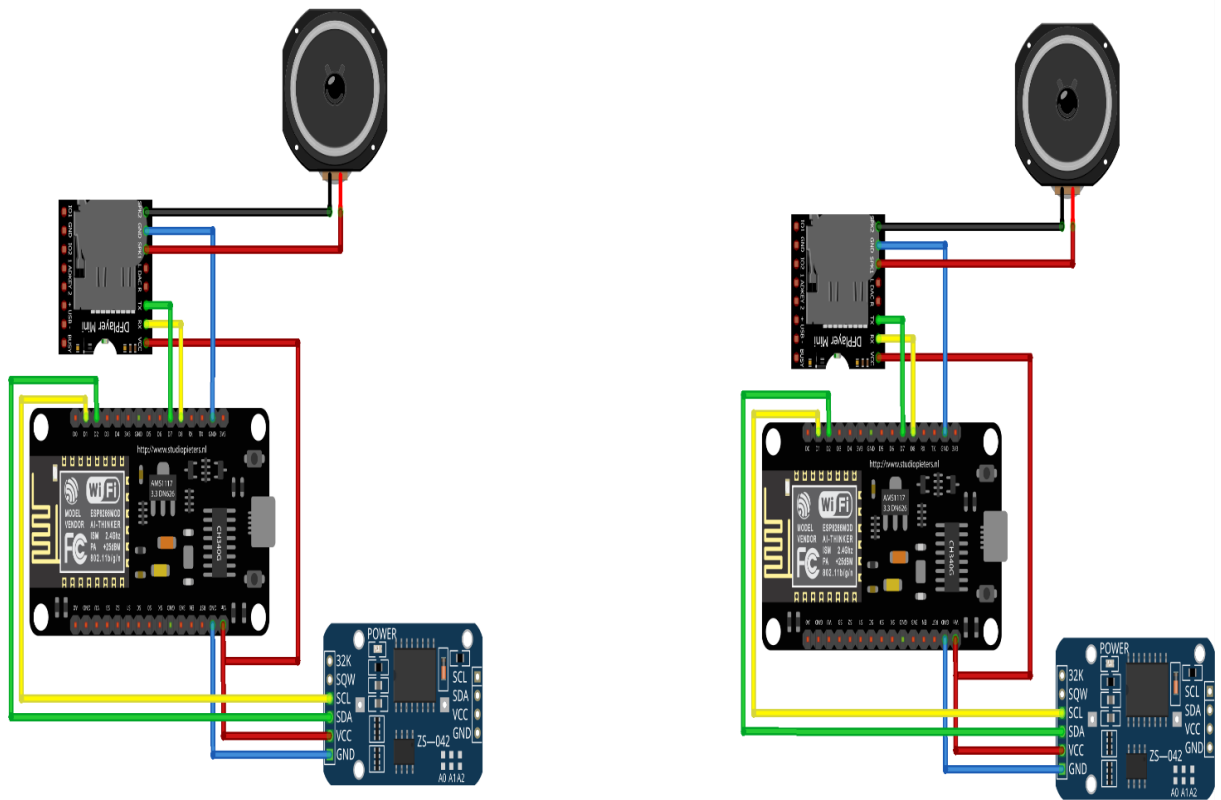
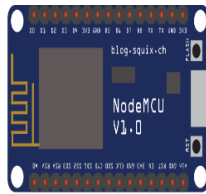


LAMPIRAN

GAMBAR RANGKAIAN KESELURUHAN



Kode Program Alat

```
1 #include "Server.h"
2
3 AsyncWebServer server(80);
4
5 #define ON_Board_LED 2
6
7 String iplocal;
8 void setup() {
9     pinMode(ON_Board_LED, OUTPUT);
10    Serial.begin(115200);
11    if (error) {
12        Serial.print(F("deserializeJson() failed: "))
13        Serial.println(error.f_str());
14        return;
15    }
16    setup_spiffs();
17    data_default();
18    setupWifi();
19    setupServer();
20    setupUDP();
21 }
22
23 void loop() {
24     delay(1000);
25     Serial.println("IP Local : " + iplocal);
26     Serial.println(WiFi.RSSI());
27 }
28 #include <ArduinoJson.h>
29 #include <Arduino.h>
30 #ifdef ESP32
31 #include <WiFi.h>
32 #include <WiFiClient.h>
33 #include <WiFiAP.h>
34 #include <AsyncTCP.h>
35 #else
36 #include <ESP8266WiFi.h>
37 #include <ESPAsyncTCP.h>
38 #endif
39 #include <ESPAsyncWebServer.h>
40 #include <FS.h>
41 #include <WiFiUdp.h>
42
43 IPAddress DestinationIP_A(192, 168, 10, 5);
44 IPAddress DestinationIP_B(192, 168, 10, 6);
45 struct dataInput {
46     String Hari;
47     String JMasuk;
48     String JIsti;
49     String JIstil;
50     String JPlg;
51     String Aktif;
52 };
53 struct dataBel {
54     String hari;
55     String Aktif;
56     String JPagi;
57     String MPagi;
58     String JIsti;
59     String MIsti;
60     String JIstil;
61     String MIstil;
62     String JPlg;
63     String MPLg;
64 };
65 dataInput InputWeb;
66 dataBel Kirim;
67 dataInput Senin = {"Senin"};
68 dataInput Selasa = {"Selasa"};
69 dataInput Rabu = {"Rabu"};
70 dataInput Kamis = {"Kamis"};
71 dataInput Jumat = {"Jumat"};
72 dataInput Sabtu = {"Sabtu"};
73 dataInput Minggu = {"Minggu"};
74 dataInput BSenin = {"BSenin"};
75 dataInput BSelasa = {"BSelas"};
76 dataInput BRabu = {"BRabu"};
77 dataInput BKamis = {"BKamis"};
78 dataInput BJumat = {"BJumat"};
79 dataInput BSabtu = {"BSabtu"};
80 dataInput BMinngu = {"BMinngu"};
```

```

82 DynamicJsonDocument doc(1024);
83 String json = "{\"hari\":\"Senin\", \"Aktif\":\"1\", \"JPagi\":\"07\", \"MPagi\":\"
84 String output;
85 DeserializationError error = deserializeJson(doc, json);
86 JsonObject obj = doc.as<JsonObject>();
87
88 void ParsingWaktu(String Waktu, String *firstVal, String *secondVal) {
89     byte titik = Waktu.indexOf('.');
90     byte panjang = Waktu.length();
91     *firstVal = Waktu.substring(0, titik);
92     *secondVal = Waktu.substring(titik + 1, panjang);
93 }
94
95 void MasukkanData(dataBel kirim) {
96     obj["hari"] = serialized(kirim.hari);
97     obj["Aktif"] = serialized(kirim.Aktif);
98     obj["JPagi"] = serialized(kirim.JPagi);
99     obj["MPagi"] = serialized(kirim.MPagi);
100    obj["JIstirahat"] = serialized(kirim.JIsti);
101    obj["MIstirahat"] = serialized(kirim.MIsti);
102    obj["JIstirahat1"] = serialized(kirim.JIstil);
103    obj["MIstirahat1"] = serialized(kirim.MIstil);
104    obj["JPulang"] = serialized(kirim.JPlg);
105    obj["MPulang"] = serialized(kirim.MPlg);
106    Serial.println(kirim.hari);
107    output = "";
108    serializeJson(doc, output);
109 }
110 #include <ArduinoJson.h>
111 #include <Arduino.h>
112 #ifdef ESP32
113 #include <WiFi.h>
114 #include <WiFiClient.h>
115 #include <WiFiAP.h>
116 #include <AsyncTCP.h>
117 #else
118 #include <ESP8266WiFi.h>
119 #include <ESPAsyncTCP.h>
120 #endif
121 #include <ESPAsyncWebServer.h>
122 #include <FS.h>
123 #include <WiFiUdp.h>
124
125 IPAddress DestinationIP_A(192, 168, 10, 5);
126 IPAddress DestinationIP_B(192, 168, 10, 6);
127 struct dataInput {
128     String Hari;
129     String JMasuk;
130     String JIsti;
131     String JIstil;
132     String JPlg;
133     String Aktif;
134 };
135 struct dataBel {
136     String hari;
137     String Aktif;
138     String JPagi;
139     String MPagi;
140     String JIsti;
141     String MIsti;
142     String JIstil;
143     String MIstil;
144     String JPlg;
145     String MPlg;
146 };
147 dataInput InputWeb;
148 dataBel Kirim;
149 dataInput Senin = {"Senin"};
150 dataInput Selasa = {"Selasa"};
151 dataInput Rabu = {"Rabu"};
152 dataInput Kamis = {"Kamis"};
153 dataInput Jumat = {"Jumat"};

```

```

154 dataInput Sabtu = {"Sabtu"};
155 dataInput Minggu = {"Minggu"};
156
157 dataInput BSenin = {"BSenin"};
158 dataInput BSelasa = {"BSelas"};
159 dataInput BRabu = {"BRabu"};
160 dataInput BKamis = {"BKamis"};
161 dataInput BJumat = {"BJumat"};
162 dataInput BSabtu = {"BSabtu"};
163 dataInput BMinggu = {"BMinggu"};
164
165 DynamicJsonDocument doc(1024);
166 String json = "{\"hari\":\"Senin\",\"Aktif\":\"1\",\"JPagi\":\"07\",\"MPagi\":\"00\",\"JIstirahat\":\"09\",\"MIstirahat\"";
167 String output;
168 DeserializationError error = deserializeJson(doc, json);
169 JsonObject obj = doc.as<JsonObject>();
170
171 void ParsingWaktu(String Waktu, String *firstVal, String *secondVal) {
172     byte titik = Waktu.indexOf('.');
173     byte panjang = Waktu.length();
174     *firstVal = Waktu.substring(0, titik);
175     *secondVal = Waktu.substring(titik + 1, panjang);
176 }
177
178 void MasukkanData(dataBel kirim) {
179     obj["hari"] = serialized(kirim.hari);
180     obj["Aktif"] = serialized(kirim.Aktif);
181     obj["JPagi"] = serialized(kirim.JPagi);
182     obj["MPagi"] = serialized(kirim.MPagi);
183     obj["JIstirahat"] = serialized(kirim.JIsti);
184     obj["MIstirahat"] = serialized(kirim.MIsti);
185     obj["JIstirahat1"] = serialized(kirim.JIstil);
186     obj["MIstirahat1"] = serialized(kirim.MIstil);
187     obj["JPulang"] = serialized(kirim.JPlg);
188     obj["MPulang"] = serialized(kirim.MPlg);
189     Serial.println(kirim.hari);
190     output = "";
191     serializeJson(doc, output);
192 }
193 char* ssid = "Server";
194 char* password = "mbalis123";
195 String ip_address = "192.168.10.4";
196
197 String ip_gateway = "192.168.10.1";
198 String ip_subnet = "255.255.255.0";
199 String ip_dns = "8.8.8.8";
200
201 //char* ssid = "cuk";
202 //char* password = "12345678";
203
204 void setupWifi(){
205     // config_statis();
206     WiFi.begin(ssid, password);
207     while (WiFi.status() != WL_CONNECTED) {
208         delay(500);
209         Serial.print(".");
210         blinkledkan(1);
211     }
212     // WiFi.softAP(ssid, password);
213     // IPAddress myIP = WiFi.softAPIP();
214     Serial.println();
215     Serial.print("IP Address: ");
216     iplocal = WiFi.localIP().toString();
217     Serial.println(iplocal);
218     // Serial.println(myIP);
219 }
220
221 void config_statis()
222 {
223     IPAddress local_IP = str2IP(ip_address);
224     IPAddress gateway = str2IP(ip_gateway);
225     IPAddress subnet = str2IP(ip_subnet);

```

```

226 IPAddress primaryDNS = str2IP(ip_dns);
227
228 WiFi.config(local_IP, gateway, subnet, primaryDNS, primaryDNS);
229 }
230
231 IPAddress str2IP(String str) {
232   IPAddress ret( getIpBlock(0, str), getIpBlock(1, str), getIpBlock(2, str), getIpBlock(3, str) );
233   return ret;
234 }
235
236 int getIpBlock(int index, String str) {
237   char separator = '.';
238   int found = 0;
239   int strIndex[] = {0, -1};
240   int maxIndex = str.length() - 1;
241   for (int i = 0; i <= maxIndex && found <= index; i++) {
242     if (str.charAt(i) == separator || i == maxIndex) {
243       found++;
244       strIndex[0] = strIndex[1] + 1;
245       strIndex[1] = (i == maxIndex) ? i + 1 : i;
246     }
247   }
248   return found > index ? str.substring(strIndex[0], strIndex[1]).toInt() : 0;
249 }
250 unsigned int localPort = 8080;
251
252 WiFiUDP udp;
253 char packetBuffer[200];
254 String inString;
255 char replyBuffer[200];
256
257 void setupUDP() {
258   digitalWrite(ON_Board_LED, HIGH); //--> Turn off Led On Board
259   //-----Wait for connection
260   udp.begin(localPort);
261   Serial.print("Local Port UDP: ");
262   udp.localPort();
263 }
264
265 void KirimJson(int Kelas, String Data) {
266   Data.toCharArray(replyBuffer, 200);
267   if (Kelas == 1) udp.beginPacket(DestinationIP_A, 1000);
268   else udp.beginPacket(DestinationIP_B, 1000);
269   udp.write(replyBuffer);
270   udp.endPacket();
271 }
272
273 void blinkledkan(byte i) {
274   for (byte k = 0; k < i; k++) {
275     delay(100);
276     digitalWrite(ON_Board_LED, LOW);
277     delay(100);
278     digitalWrite(ON_Board_LED, HIGH);
279   }
280 }
281 void data_default() {
282   if (readFile(SPIFFS, "/Senin.pagi.txt") == "") {
283     Serial.println("data default");
284     dataInput data = {
285       "Senin",
286       "07.00",
287       "09.00",
288       "09.15",
289       "11.00",
290       "1"
291     };
292     Senin = data;
293     Senin.Hari = "Senin";
294     Selasa = data;
295     Selasa.Hari = "Selasa";
296     Rabu = data;
297     Rabu.Hari = "Rabu";

```

```

268 | else udp.beginPacket(DestinationIP_B, 1000);
269 | udp.write(replyBuffer);
270 | udp.endPacket();
271 | }
272 |
273 | void blinkledkan(byte i){
274 |   for (byte k = 0; k < i; k++) {
275 |     delay(100);
276 |     digitalWrite(ON_Board_LED, LOW);
277 |     delay(100);
278 |     digitalWrite(ON_Board_LED, HIGH);
279 |   }
280 | }
281 | void data_default() {
282 |   if (readFile(SPIFFS, "/Senin.pagi.txt") == "") {
283 |     Serial.println("data default");
284 |     dataInput data = {
285 |       "Senin",
286 |       "07.00",
287 |       "09.00",
288 |       "09.15",
289 |       "11.00",
290 |       "1"
291 |     };
292 |     Senin = data;
293 |     Senin.Hari = "Senin";
294 |     Selasa = data;
295 |     Selasa.Hari = "Selasa";
296 |     Rabu = data;
297 |     Rabu.Hari = "Rabu";
298 |     Kamis = data;
299 |     Kamis.Hari = "Kamis";
300 |     Jumat = data;
301 |     Jumat.Hari = "Jumat";
302 |     Sabtu = data;
303 |     Sabtu.Hari = "Sabtu";
304 |
305 |     BSenin = data;
306 |     BSenin.Hari = "Senin";
307 |     BSelasa = data;
308 |     BSelasa.Hari = "Selasa";
309 |     BRabu = data;
310 |     BRabu.Hari = "Rabu";
311 |     BKamis = data;
312 |     BKamis.Hari = "Kamis";
313 |     BJumat = data;
314 |     BJumat.Hari = "Jumat";
315 |     BSabtu = data;
316 |     BSabtu.Hari = "Sabtu";
317 |     simpandata('A', Senin);
318 |     simpandata('A', Selasa);
319 |     simpandata('A', Rabu);
320 |     simpandata('A', Kamis);
321 |     simpandata('A', Jumat);
322 |     simpandata('A', Sabtu);
323 |     simpandata('B', Senin);
324 |     simpandata('B', Selasa);
325 |     simpandata('B', Rabu);
326 |     simpandata('B', Kamis);
327 |     simpandata('B', Jumat);
328 |     simpandata('B', Sabtu);
329 |     writeFile(SPIFFS, "/IP_A.txt", "192.168.10.11");
330 |     writeFile(SPIFFS, "/IP_B.txt", "192.168.10.12");
331 |   }
332 | }
333 | void simpandata(char kelas, dataInput data) {
334 |   if (kelas == 'A') {
335 |     Serial.println("Kelas A");
336 |     String hari = data.Hari;
337 |     String JMasuk = data.Hari + ".pagi";
338 |     String JIsti = data.Hari + ".isti";

```



```

339     String JIstil = data.Hari + ".istil";
340     String JPlg = data.Hari + ".plg";
341     String Aktif = data.Hari + ".Aktif";
342     simpan(JMasuk, (String)data.JMasuk);
343     simpan(JIsti, (String)data.JIsti);
344     simpan(JIstil, (String)data.JIstil);
345     simpan(JPlg, (String)data.JPlg);
346     simpan(Aktif, (String)data.Aktif);
347 }
348 else {
349     Serial.println("Kelas B");
350     String hari = data.Hari;
351     String JMasuk = "B" + data.Hari + ".pagi";
352     String JIsti = "B" + data.Hari + ".isti";
353     String JIstil = "B" + data.Hari + ".istil";
354     String JPlg = "B" + data.Hari + ".plg";
355     String Aktif = "B" + data.Hari + ".Aktif";
356     simpan(JMasuk, data.JMasuk);
357     simpan(JIsti, data.JIsti);
358     simpan(JIstil, data.JIstil);
359     simpan(JPlg, data.JPlg);
360     simpan(Aktif, data.Aktif);
361 }
362 }
363
364 void simpan(String text, String data) {
365     String nametext = "/" + text + ".txt";
366     writeFile(SPIFFS, nametext.c_str(), data.c_str());
367     Serial.println("Menyimpan : " + nametext + " : " + data);
368 }
369
370 void setup_spiffs() {
371     if (!SPIFFS.begin()) {
372         Serial.println("An Error has occurred while mounting SPIFFS");
373         return;
374     }
375     else {
376         Serial.println("SPIFFS running ok");
377     }
378 }
379
380 String readFile(fs::FS &fs, const char * path) {
381     File file = fs.open(path, "r");
382     if (!file || file.isDirectory())
383     {
384         return String();
385     }
386     String fileContent;
387     while (file.available())
388     {
389         fileContent += String((char)file.read());
390     }
391     return fileContent;
392 }
393
394 void writeFile(fs::FS &fs, const char * path, const char * message)
395 {
396     File file = fs.open(path, "w");
397     if (!file)
398     {
399         return;
400     }
401     file.print(message);
402 }

```

Kode Program Alat client

https://drive.google.com/file/d/18CXw1ExoAlj9fe9lbdLa8T48oEnErqAH/view?usp=share_link

Kode program aplikasi

https://drive.google.com/file/d/1L31rRVTMxfzZ21xesqJ-UcWC9iIVm_sC/view?usp=share_link