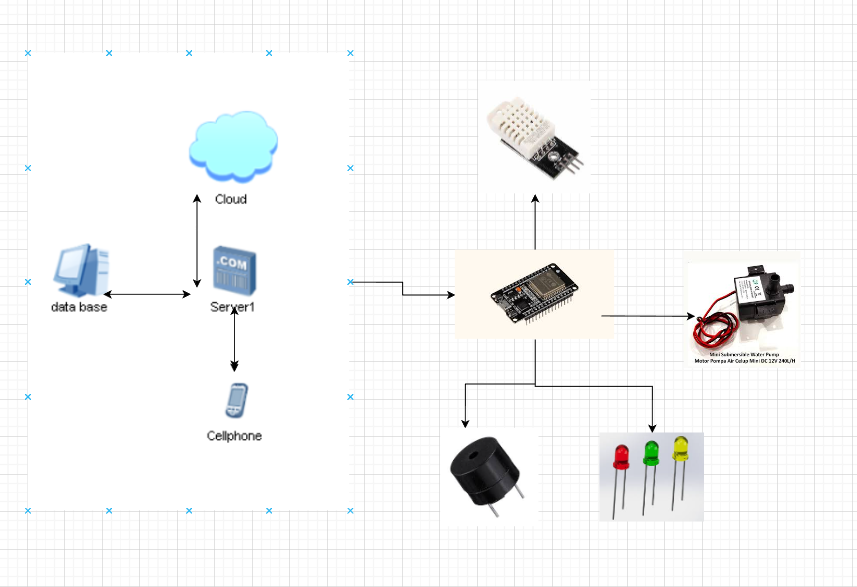
Checa Gattan albikhi

152022200

Jawaban soal uts pemograman iot

1.arsitektur iot



**2.Penjelasan Konsep**:

1. **Pengambilan Data dari Sensor**:
   * Sensor yang digunakan, seperti DHT (untuk suhu dan kelembapan), akan mengukur kondisi lingkungan dan mengirim data dalam bentuk sinyal analog atau digital ke mikrokontroler, seperti Arduino atau ESP32.
2. **Pemrosesan Data di Mikrokontroler**:
   * Mikrokontroler menerima sinyal dari sensor dan mengonversinya ke dalam bentuk data yang dapat dibaca. Data ini kemudian diolah sesuai dengan program yang sudah diatur. Mikrokontroler akan mengaktifkan atau menonaktifkan komponen seperti LED, relay, atau buzzer sesuai dengan kondisi yang diukur.
3. **Pengiriman Data ke User Interface**:
   * Data yang sudah diolah dikirimkan ke user interface aplikasi menggunakan protokol komunikasi, misalnya **MQTT** (Message Queuing Telemetry Transport). MQTT memungkinkan komunikasi data yang ringan dan cepat antara perangkat IoT dan aplikasi. Mikrokontroler dapat mengirimkan data ke broker MQTT yang akan meneruskan data ke user interface di aplikasi.
   * Aplikasi pada perangkat user (web, mobile, atau desktop) akan menerima data tersebut dan menampilkannya dalam bentuk visual yang mudah dibaca pengguna.
4. **Penyimpanan ke Database**:
   * Data yang diterima aplikasi dapat disimpan di database untuk keperluan pencatatan atau analisis lebih lanjut. Database ini bisa berupa **Firebase**, **MySQL**, atau **MongoDB**, tergantung kebutuhan dan kemampuan aplikasi.
5. **Feedback atau Kontrol**:
   * Aplikasi juga dapat mengirim instruksi balik ke mikrokontroler (misalnya, untuk mengaktifkan atau menonaktifkan pompa) berdasarkan kondisi lingkungan atau perintah pengguna, menggunakan MQTT atau protokol lain yang mendukung komunikasi dua arah.

1. #include <DHT.h>

2. #include <WiFi.h>

3. #include <HTTPClient.h>

4. #include "PubSubClient.h"

5.

6. // Pin sensor

7. const int DHTPIN = 15; // Pin DHT22 untuk suhu dan kelembapan

8. const int DHTTYPE = DHT22; // Tipe sensor DHT yang digunakan

9. const int TURBIDITY\_PIN = 34; // Pin analog untuk simulasi sensor kekeruhan

10. const int PH\_PIN = 35; // Pin analog untuk simulasi sensor pH

11.

12. // Pin kontrol perangkat

13. const int LED\_GREEN = 5; // Pin untuk LED Hijau

14. const int LED\_YELLOW = 17; // Pin untuk LED Kuning

15. const int LED\_RED = 4; // Pin untuk LED Merah

16. const int RELAY\_PUMP = 27; // Pin untuk relay pompa

17. const int BUZZER = 2; // Pin untuk buzzer

18.

19. // WiFi settings

20. const char\* ssid = "frenz"; // Ganti dengan SSID jaringan Anda

21. const char\* password = "frenz1975"; // Ganti dengan password WiFi Anda

22.

23. // MQTT Broker settings

24. const char\* mqtt\_server = "broker.hivemq.com"; // Ganti dengan IP atau alamat broker MQTT

25. const int mqtt\_port = 1883; // Port broker MQTT

26. const char\* pub\_lampu = "status175/lampu";

27. const char\* pub\_relay = "status175/relay";

28. const char\* pub\_buzzer = "status175/buzzer";

29. const char\* sub\_suhu = "sensor175/suhu";

30. const char\* sub\_humi = "sensor175/kelembapan";

31. const char\* sub\_turbidity = "sensor175/kekeruhan";

32. const char\* sub\_ph = "sensor175/ph";

33.

34. // Alamat URL Flask server

35. const char\* flask\_server\_url = "http://192.168.26.248:5000/data"; // Ganti <NGROK\_URL> dengan URL ngrok yang digunakan

36.

37. WiFiClient espClient;

38. PubSubClient client(espClient);

39.

40. DHT dht(DHTPIN, DHTTYPE);

41.

42. // Variabel untuk menyimpan nilai suhu dan kelembapan

43. float suhu = -1;

44. float humidity = -1;

45.

46. void setup() {

47. Serial.begin(115200);

48.

49. // Setup untuk sensor

50. dht.begin();

51. pinMode(LED\_GREEN, OUTPUT);

52. pinMode(LED\_YELLOW, OUTPUT);

53. pinMode(LED\_RED, OUTPUT);

54. pinMode(RELAY\_PUMP, OUTPUT);

55. pinMode(BUZZER, OUTPUT);

56.

57. digitalWrite(LED\_GREEN, LOW);

58. digitalWrite(LED\_YELLOW, LOW);

59. digitalWrite(LED\_RED, LOW);

60. digitalWrite(RELAY\_PUMP, LOW);

61. digitalWrite(BUZZER, LOW);

62.

63. // Connect to WiFi

64. WiFi.begin(ssid, password);

65. while (WiFi.status() != WL\_CONNECTED) {

66. delay(500);

67. Serial.print(".");

68. }

69. Serial.println("Connected to WiFi");

70.

71. // Connect to MQTT Broker

72. client.setServer(mqtt\_server, mqtt\_port);

73. client.setCallback(mqtt\_callback);

74.

75. // Connect to MQTT broker

76. connectToMQTT();

77. }

78.

79. void loop() {

80. if (!client.connected()) {

81. connectToMQTT();

82. }

83. client.loop();

84.

85. delay(2000); // Delay sebelum pembacaan data berikutnya

86. }

87.

88. void connectToMQTT() {

89. while (!client.connected()) {

90. Serial.print("Connecting to MQTT...");

91. String clientId = "ESP32Client-";

92. clientId += String(random(0xffff), HEX);

93.

94. if (client.connect(clientId.c\_str())) {

95. Serial.println("Connected to MQTT broker");

96. // Subscribe to sensor data topics to receive sensor values

97. client.subscribe(sub\_suhu);

98. client.subscribe(sub\_humi);

99. client.subscribe(sub\_turbidity);

100. client.subscribe(sub\_ph);

101. } else {

102. Serial.print("Failed to connect, rc=");

103. Serial.print(client.state());

104. Serial.println(" Try again in 5 seconds");

105. delay(5000);

106. }

107. }

108. }

109.

110. void mqtt\_callback(char\* topic, byte\* payload, unsigned int length) {

111. String message = "";

112. for (int i = 0; i < length; i++) {

113. message += (char)payload[i];

114. }

115.

116. if (String(topic) == sub\_suhu) {

117. Serial.print("Received Suhu: ");

118. Serial.println(message);

119.

120. suhu = message.toFloat();

121. String lampuStatus;

122.

123. if (suhu < 30) {

124. digitalWrite(LED\_GREEN, HIGH);

125. digitalWrite(LED\_YELLOW, LOW);

126. digitalWrite(LED\_RED, LOW);

127. lampuStatus = "{\"green\": \"ON\"}";

128. } else if (suhu >= 30 && suhu <= 35) {

129. digitalWrite(LED\_GREEN, LOW);

130. digitalWrite(LED\_YELLOW, HIGH);

131. digitalWrite(LED\_RED, LOW);

132. lampuStatus = "{\"yellow\": \"ON\"}";

133. } else {

134. digitalWrite(LED\_GREEN, LOW);

135. digitalWrite(LED\_YELLOW, LOW);

136. digitalWrite(LED\_RED, HIGH);

137. digitalWrite(BUZZER, HIGH);

138. lampuStatus = "{\"red\": \"ON\"}";

139. }

140. client.publish(pub\_lampu, lampuStatus.c\_str());

141.

142. // Kirim suhu dan kelembapan ke Flask jika kelembapan telah diterima

143. if (humidity != -1) {

144. sendToFlask(suhu, humidity);

145. }

146. }

147.

148. if (String(topic) == sub\_humi) {

149. Serial.print("Received Kelembapan: ");

150. Serial.println(message);

151.

152. humidity = message.toFloat();

153.

154. if (humidity < 50) {

155. if (digitalRead(RELAY\_PUMP) == LOW) {

156. digitalWrite(RELAY\_PUMP, HIGH);

157. client.publish(pub\_relay, "Pompa ON");

158. }

159. if (digitalRead(BUZZER) == LOW) {

160. digitalWrite(BUZZER, HIGH);

161. client.publish(pub\_buzzer, "Buzzer ON");

162. }

163. } else {

164. if (digitalRead(RELAY\_PUMP) == HIGH) {

165. digitalWrite(RELAY\_PUMP, LOW);

166. client.publish(pub\_relay, "Pompa OFF");

167. }

168. if (digitalRead(BUZZER) == HIGH) {

169. digitalWrite(BUZZER, LOW);

170. client.publish(pub\_buzzer, "Buzzer OFF");

171. }

172. }

173.

174. // Kirim suhu dan kelembapan ke Flask jika suhu telah diterima

175. if (suhu != -1) {

176. sendToFlask(suhu, humidity);

177. }

178. }

179. }

180.

181. void sendToFlask(float suhu, float humidity) {

182. if (WiFi.status() == WL\_CONNECTED) {

183. HTTPClient http;

184. http.begin(flask\_server\_url);

185.

186. // JSON data sesuai dengan yang diberikan

187. String jsonData = "{\"suhumax\":50,\"suhumin\":35,\"suhurata2\":42.5,\"nilaisuhuhumid\":[{\"id\":1,\"suhu\":"

188. + String(suhu) + ",\"humid\":" + String(humidity)

189. + ",\"kecerahan\":200,\"timestamp\":\"2024-11-12T08:00:00\"}],\"month\_year\":\"11-2024\"}";

190.

191. http.addHeader("Content-Type", "application/json");

192.

193. int httpResponseCode = http.POST(jsonData);

194. if (httpResponseCode > 0) {

195. String response = http.getString();

196. Serial.println("Response: " + response);

197. } else {

198. Serial.print("Error on sending POST: ");

199. Serial.println(httpResponseCode);

200. }

201.

202. http.end();

203. } else {

204. Serial.println("Error in WiFi connection");

205. }

206. }

207.

**2.Membuat backend yang menghasilkan format json**