

# LAMPIRAN

## 1. Kode Program NodeMCU ESP8266

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266HTTPClient.h>
#include <DHT.h>

#define DHT_PIN D2
#define DHT_TYPE DHT11
DHT dht(DHT_PIN, DHT_TYPE);

unsigned long timeEnc, timeMk;
unsigned char *key = (unsigned char *)"Bambang Fitriadi Wiansyah";
unsigned char *plaintext;
String Data;

const char *ssid = "R";
const char *pass = "00000000";
const char *host = "http://rivestcode.000webhostapp.com/model/insert.php";

void setup() {
  Serial.begin(115200);
  Serial.print(F("Config DHT11 => "));
  dht.begin();
  Serial.println(F("Done"));

  Serial.println(F("Set WiFi mode to WIFI_STA"));
  WiFi.mode(WIFI_STA);
  Serial.print(F("Connecting to "));
  Serial.print(ssid);
```

```
WiFi.begin(ssid, pass);
while (WiFi.status() != WL_CONNECTED) {
  Serial.print(".");
  delay(3000);
}
Serial.println();
Serial.print("Connected!");
Serial.print("IP address: ");
Serial.println(WiFi.localIP());
}

void loop() {
  timeMk = millis();

  Serial.print(F("Getting Data Sensor => "));
  int tem = dht.readTemperature();
  int hum = dht.readHumidity();

  if (isnan(tem) || isnan(hum)) {
    Serial.println(F("Error"));
    return;
  }
  Serial.println(F("Success"));

  Data = "suhu=" + String(tem) + ",kelembaban=" + String(hum);
  Serial.println("Data Sensor : " + Data);

  encryption();
  sendData();
}
```

```
timeMk = millis() - timeMk;
Serial.print("Time Loop : ");
Serial.print(timeMk);
Serial.println(" ms");
Serial.print("Time Enc : ");
Serial.print(timeEnc);
Serial.println(" ms\n");
delay(5000);
}
```

```
void sendData() {
    String postData = "data=" + Data;
    Serial.println("Send to server : " + postData);
    HTTPClient http;
    WiFiClient client;
    http.begin(client, host);
    http.addHeader("Content-Type", "application/x-www-form-urlencoded");
    auto httpCode = http.POST(postData);
    String payload = http.getString();
    Serial.print("Status Data Upload : ");
    Serial.println(payload);
    http.end();
}
```

```
void encryption() {
    timeEnc = millis();
    int i, j = 0, temp, n, t, keyStream;
    int strlength = Data.length() + 1;
```

```

char charBuffer[strlength];
Data.toCharArray(charBuffer, strlength);
plaintext = (unsigned char *)charBuffer;
Serial.print("Key for Encrypt : ");
Serial.println((char *)key);
int sbox[256], skey[256];
for (i = 0; i < 256; i++) {
    sbox[i] = i;
    skey[i] = key[(i % strlen((char *)key))];
}
Serial.println("\nsBox :");
for (i = 0; i < 256; i++) {
    Serial.print(sbox[i]);
    Serial.print(" ");
}
Serial.println("\nKey :");
for (i = 0; i < 256; i++) {
    Serial.print(skey[i]);
    Serial.print(" ");
}
Serial.println("\nKSA :");
for (i = 0; i < 256; i++) {
    j = (j + sbox[i] + skey[i]) % 256;
    temp = sbox[i];
    sbox[i] = sbox[j];
    sbox[j] = temp;
}
for (i = 0; i < 256; i++) {
    Serial.print(sbox[i]);

```

```
Serial.print(" ");
}
Data = "";
Serial.print("\n\nData Encrypted : ");
for (i = j = n = 0; n < (int)strlen((char *)plaintext); n++) {
    i = (i + 1) % 256;
    j = (j + sbox[i]) % 256;
    temp = sbox[i];
    sbox[i] = sbox[j];
    sbox[j] = temp; //swap
    t = (sbox[i] + sbox[j]) % 256;
    keyStream = sbox[t]; //keyStream
    Data += (plaintext[n] ^ keyStream);
    Serial.print((plaintext[n] ^ keyStream));
    if (n < (int)strlen((char *)plaintext) - 1) {
        Data += ",";
        Serial.print(" ");
    }
}
timeEnc = millis() - timeEnc;
Serial.println();
}
```

## 2. Kode Program Tampilan Web

```
<script setup lang="ts">
import { ref } from "vue";
import { database } from "@/firebase";
import { ref as rtDb, onValue } from "firebase/database";
import { rc4 } from "@/library/rc4";

const enc = ref();
const text = ref();
const key = ref('Bambang Fitriadi Wiansyah');
const temperature = ref();
const humidity = ref();
const timestamp = ref();

const path = rtDb(database, "/");
onValue(path, (Snapshot) => {
  const snapshot = Snapshot.val()
  const { encrypt, decrypt, suhu, kelembaban } = rc4(snapshot.data, key.value);
  enc.value = encrypt;
  text.value = decrypt;
  temperature.value = suhu;
  humidity.value = kelembaban;
  timestamp.value = snapshot.timestamp
})
</script>

<template>
  <main class="flex flex-col justify-center flex-grow bg-stone-700">
    <div class="container flex flex-col gap-4 p-4 mx-auto">
```

```

<div class="grid gap-4 md:grid-cols-2 lg:grid-cols-4">
  <div :class="(enc) ? 'bg-teal-700' : 'bg-teal-800'"
class="component">
    <div class="bg-teal-500 icon"></div>
    <p class="">Data Logged</p>
    <h4 class=""></h4>
  </div>

  <div :class="(text) ? 'bg-green-700':'bg-gray-600'" class="component">
    <div class="bg-green-500 icon"></div>
    <p class="">Status</p>
    <h4 class=""></h4>
  </div>

  <div :class="(temperature) ? 'bg-red-700':'bg-red-
800'" class="component">
    <div class="bg-red-500 icon"></div>
    <p class="">Temperature</p>
    <h4 class="">{{ (temperature) ?
`${temperature}°C` : "" }}</h4>
  </div>

  <div :class="(humidity) ? 'bg-blue-700':'bg-blue-
800'" class="component">
    <div class="bg-blue-500 icon"></div>
    <p>Humidity</p>
    <h4 class="">{{ (humidity) ?
`${humidity}%` : "" }}</h4>
  </div>
</div>

```



```

<div class="grid gap-4 md:grid-cols-2">
  <div :class="(enc) ? 'bg-gray-600' : 'bg-gray-800'"
class="component">
    <div class="bg-gray-500 icon"></div>
    <p class="">Data Encrypted</p>
    <h4 class="font-mono ">{{ enc ?? ""
}}</h4>
  </div>

  <div :class="(text) ? 'bg-gray-600': 'bg-gray-800'"
class="component">
    <div class="bg-gray-500 icon"></div>
    <p class="">Data Decrypted</p>
    <h4 class="font-mono ">{{ text ?? ""
}}</h4>
  </div>
</div>

<div class="grid gap-4">
  <div :class="(enc) ? 'bg-green-700' : 'bg-green-800'"
class="component">
    <div class="bg-green-500 icon"></div>
    <p class="">Last Updated</p>
    <h4 class="">{{ timestamp ?? "" }}</h4>
  </div>
</div>
</div>
</main>
</template>
<style scoped>
.component {

```

```
    @apply p-4 transition-colors duration-300 rounded-lg flex flex-col items-
center justify-center relative overflow-hidden;
}
.icon {
    @apply absolute w-10 lg:w-12 aspect-square rounded-br-3xl top-0 left-0;
}
main > div > div {
    @apply lg:h-36;
}
p {
    @apply text-xl font-light underline underline-offset-4;
}
div > h4 {
    @apply text-2xl lg:text-4xl font-semibold;
}
</style>
```

### 3. Kode Program Dekripsi Data Pada Web

```
export function rc4(data:String, key:String) {
  const sbox: Array<number> = [];
    const skey: Array<number> = [];
  const ciphertext: Array<number> = data.split(",").map(Number);
  const plaintext: Array<String> = [];
    const cipher: Array<String> = [];

  let index, tmp, pointer, indexKeyStream, indexPlaintext;
    for(index = 0; index < 256; index++) {
      skey[index] = key.charCodeAt(index%key.length)
    }

  for (index = 0; index < 256; index++) {
    sbox[index] = index;
  }

  for(index = tmp = pointer = 0; index < 256; index++) {
    pointer = (pointer + sbox[index] + skey[index]) % 256;
    tmp = sbox[index];
    sbox[index] = sbox[pointer];
    sbox[pointer] = tmp;
  }

  for(index = pointer = indexPlaintext = 0; indexPlaintext <
  ciphertext.length; indexPlaintext++) {
    index = (index + 1) % 256;
    pointer = (pointer + sbox[index]) % 256;
    tmp = sbox[index];
    sbox[index] = sbox[pointer];
```

```

        sbox[pointer] = tmp;
        indexKeyStream = (sbox[index] + sbox[pointer]) % 256;
        plaintext.push(String.fromCharCode(ciphertext[indexPlaintext] ^
sbox[indexKeyStream]));
    }

    for(index = 0; index < ciphertext.length; index++) {
        cipher.push(String.fromCharCode(ciphertext[index]));
    }

    const encrypt = cipher.join("");
    const decrypt = plaintext.join("");

    const value = decrypt.split(",");
    const suhu = (value[0].indexOf("suhu=") !== -1) ?
value[0].split("suhu=").pop() : null;
    const kelembaban = (value[1].indexOf("kelembaban=") !== -1) ?
value[1].split("kelembaban=").pop() : null;

    return { encrypt, decrypt, suhu, kelembaban }
    // return { encrypt, decrypt }
}

```