

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Hasil Pengumpulan Data (Pemaparan)

##### 4.1.1 Observasi

Hasil observasi yang dilakukan di PLTMH andung biru ialah dalam melakukan pengecekan pada tegangan dan debit pada PLTMH, untuk pengecekannya mendatangi ke lokasi PLTMH, dengan pengecekan seperti ini kerap mempersulit petugas dalam pengecekan rutin pada PLTMH, oleh karena itu perlu suatu sistem yang dapat mempermudah petugas dalam melakukan pengecekan arus listrik dan debit air pada PLTMH.

##### 4.1.2 Wawancara

Wawancara dilakukan bersama dengan bapak rasit (Pembangun PLTMH andung biru), Narasumber kedua yaitu bapak suto (pengecek PLTMH) yang bertugas mengecek keseluruhan PLTMH, adapun hasil dari wawancara adalah sebagai berikut:

1. Prosedur dalam pengecekan arus tegangan listrik dan debit air di lakukan dengan mendatangi PLTMH.
2. Pengecek kesulitan dalam pengecekan rutin dikarenakan lokasi yang jauh dan cuaca tidak menentu.
3. Pengecek mengecek arus tegangan listrik yang dihasilkan PLTMH, dan debit air apakah normal atau tidak.

#### 4.2 Hasil Analisis dan Desain

Setelah mengumpulkan data, selanjutnya informasi dari data akan diimplementasikan terhadap sistem. Adapun hasil dari pengembangan sistem yang mengacu pada metode *waterfall* adalah sebagai berikut:

##### 4.2.1 Analisis Sistem (Requirements Definition)

Setelah melakukan penelitian secara langsung bagaimana proses pemantauan arus tegangan listrik dan debit air yang pengecekannya masih manual atau secara langsung ke lokasi PLTMH. Hal ini kurang efisien waktu dan tenaga. Karena untuk mengetahui arus tegangan listrik dan debit air pengawas atau pemantauan harus ke

PLTMH tersebut untuk mengeceknya, perlunya pengecekan atau pemantauan untuk meminimalkan arus tegangan listrik rendah dikarenakan debit air yang rendah, maka perlu dibangun sebuah sistem yang akan membantu dalam pemantauan arus tegangan listrik dan debit air pada PLTMH.

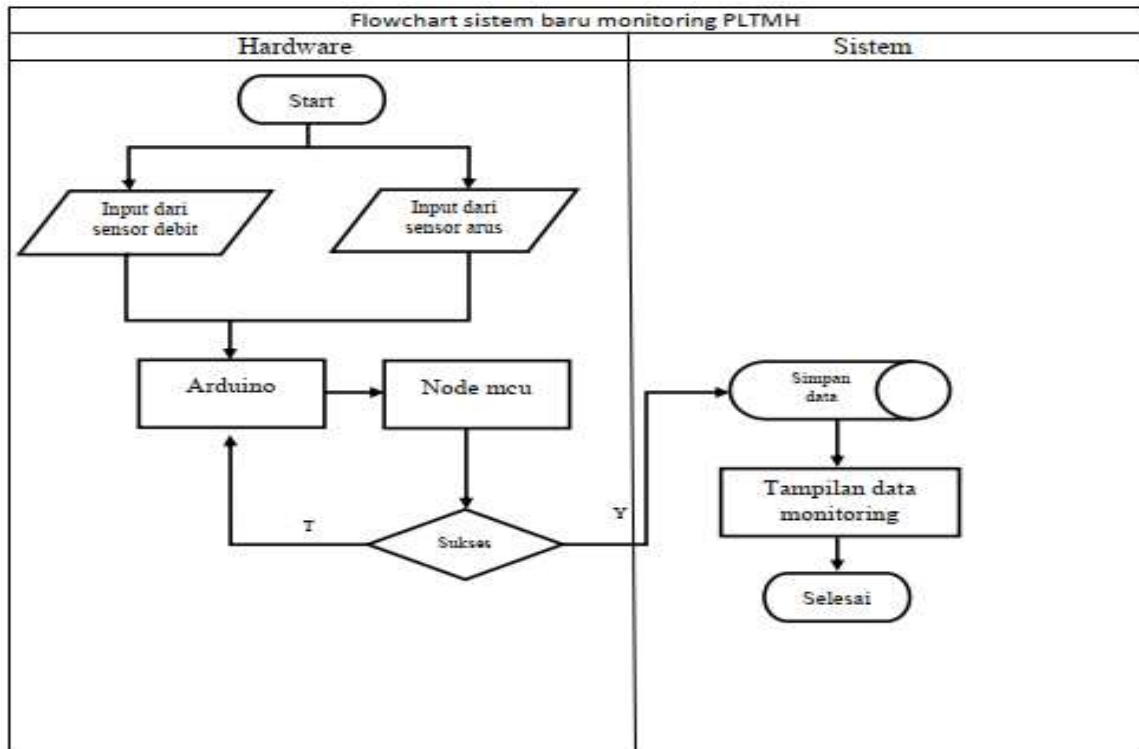
Adapun sistem yang dibutuhkan dalam monitoring arus tegangan listrik dan debit air pada PLTMH adalah sistem yang dapat menampung data monitoring arus tegangan listrik dan debit air yang masih berlangsung dan menyimpan data histori pemantauan sehingga dapat memudahkan pemantauan dengan mengesampingkan jarak dan waktu. Dengan menerapkan sistem pemantauan secara *online* maka pemantauan dapat dengan mudah mendapatkan data arus tegangan listrik dan debit air pada PLTMH, mempermudah proses pemantauan, menghapus jarak dan waktu sebagai salah satu kendala yang dihadapi dalam hal ini proses monitoring arus listrik dan debit air dilakukan secara *online* sehingga diharapkan dapat mencegah terjadinya arus listrik menurun.

#### 4.2.2 Desain Sistem (Design System)

Desain sistem dilakukan setelah tahap analisis selesai dilakukan. Tahapan ini bertujuan memberikan gambaran kepada *user* untuk menspesifikasi kebutuhan sistem dan *software* serta mendefinisikan arsitektur sistem secara keseluruhan. Desain sistem yang diimplementasikan dalam pembuatan aplikasi *marketplace* ini berupa perancangan bagan alir sistem (*flowchart*), dan desain *View*.

##### 1. Bagan Alir Sistem (*Flowchart*)

*Flowchart* yang digunakan pada penelitian ini adalah *flowchart* program, dideskripsikan pada **Gambar 4.1**



**Gambar 4. 1** Flowchart Sistem Monitoring

## 2. Struktur Database

Dalam mengimplementasikan database dibutuhkan struktur database yang tepat agar tidak ada overload, redundansi dan sinkonsistensi data. Struktur database dari sistem monitoring PLTMH akan dijelaskan dalam bentuk tabel-tabel berikut:

**Tabel 4. 1** Tabel *User* (Pemonitoring)

No.	Nama	Jenis	Key
1.	Id	bigint (20)	<i>Primary Key</i>
2.	Name	Varchar(50)	
3.	Email	Varchar(50)	
4.	email_verified_at	timestamp	
5.	Password	Varchar(20)	

6.	remember_token	timestamp	
7.	created_at	timestamp	
8.	updated_at	timestamp	

**Tabel 4. 2** Tabel Monitoring

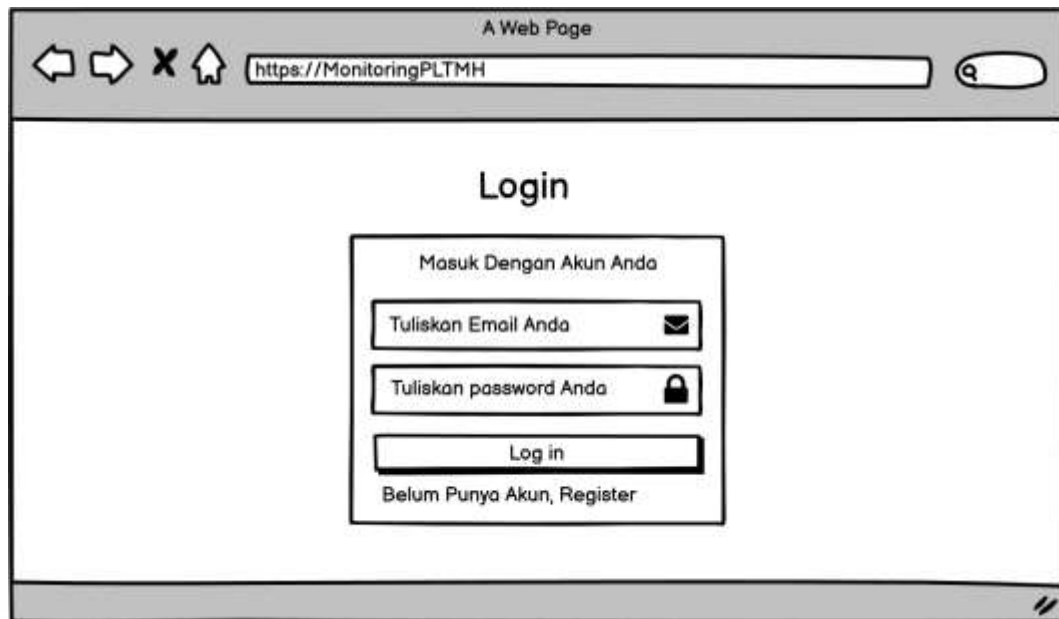
No.	Nama	Jenis	Key
1.	Id	bigint (20)	<i>Primary Key</i>
2.	Tegangan	varchar(255)	
3.	Debid	varchar(255)	
4.	created_at	timestamp	
5.	updated_at	timestamp	

Pada sistem ini tidak memiliki *Data Flow Diagram* (DFD) dan Entity Relationship Diagram (ERD), dikarenakan antara table *User* dan table monitoring tidak memiliki relasi.

### 3. Desain View

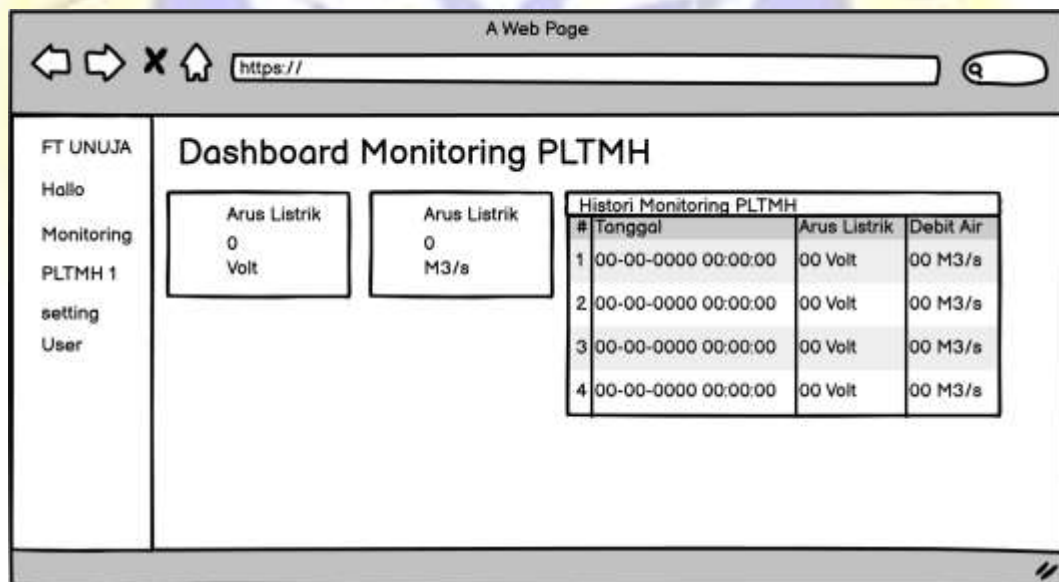
Desain view dibutuhkan sebagai gambaran interface yang akan diimplementasikan dalam sistem. Dibawah ini adalah desain view yang digunakan.

- Desain Halaman *Login*



**Gambar 4. 2** Desain Halaman *Login*

- Desain Halaman *Dashboard*



**Gambar 4. 3** Desain Halaman *Dashboard*

### 4.3 Implementasi

Setelah tahap desain dan perancangan alur program dilakukan, tahap berikutnya yaitu implementasi dari desain dan perancangan yang telah dibuat sebelumnya. Pada penelitian ini aplikasi yang digunakan dalam tahap implementasi khususnya penulisan

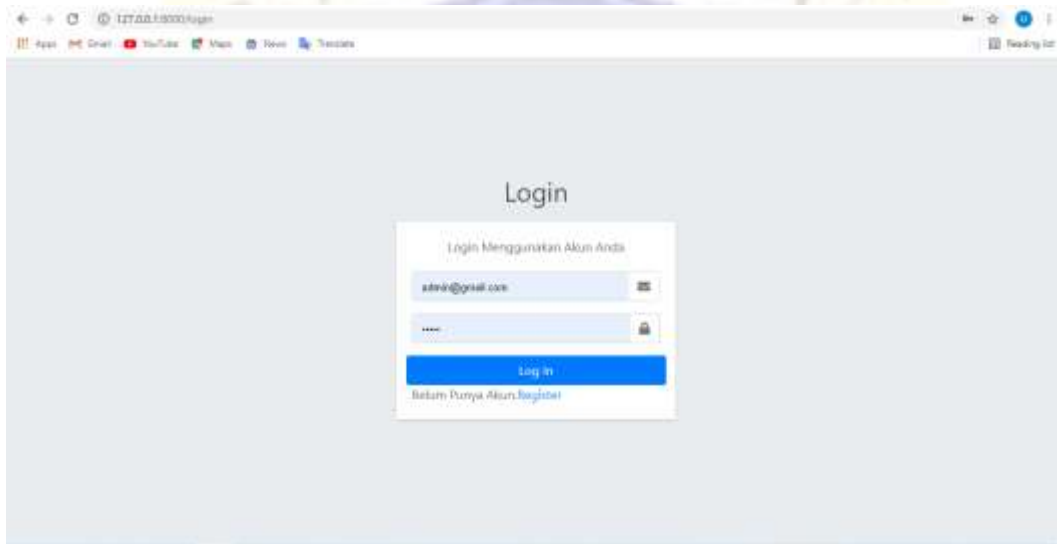
sinkode adalah *Visual Studio Code* sebagai *text editor* dan *Google Chrome* untuk menampilkan hasil dari penulisan sinkode (*View*).

#### 4.3.1 Implementasi (Coding dan View)

Berikut hasil dari implementasi Sistem monitoring arus tegangan listrik dan debit air pada PLTMH:

##### 1. Halaman *Login User* (Pengawas keseharian PLTMH)

Halaman login adalah halaman yang utama sebelum pemantauan masuk ke halaman utama aplikasi. Halaman ini juga sebagai implementasi dari proses *authentication* dari aplikasi.

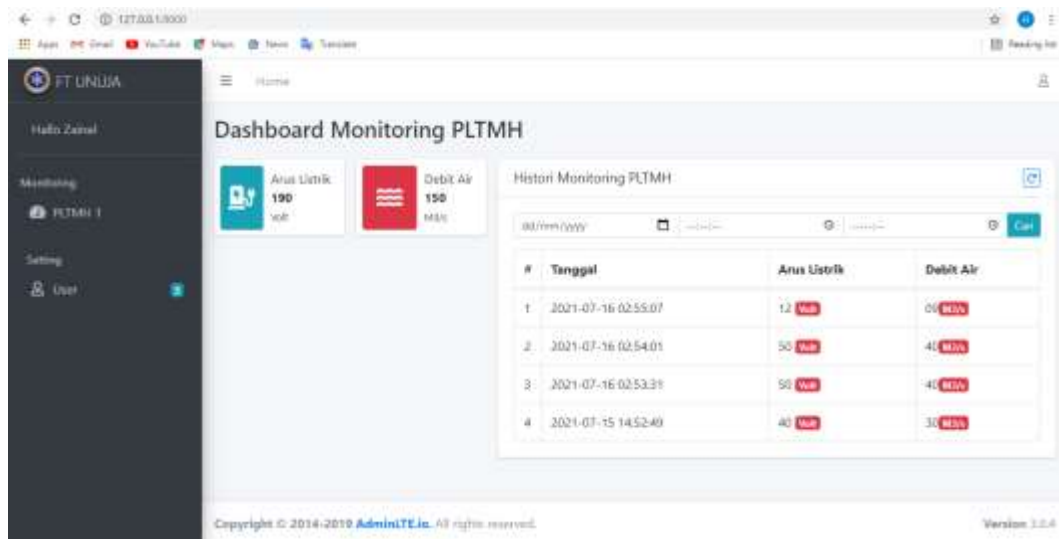


**Gambar 4. 4** Halaman *Login*

##### 2. Halaman *Dashboard*

Halaman *dashboard* adalah halaman utama yang akan dieksekusi ketika petugas pemantauan sudah melakukan proses login dan sukses. Di halaman ini pemantauan dapat melihat informasi arus tegangan listrik dan debit air pada PLTMH, dan juga terdapat tabel histori pemantauan.





**Gambar 4.5** Halaman *Dashboard*

Data monitoring PLTMH secara real time dan histori monitoring

**Segmen Program 4.1** Sinkode *Index DashboardContoller*

```

1. @extends('layouts.main')
2. @section('title', 'Dashboard')
3. @section('content')
4. <!-- Content Wrapper. Contains page content -->
5. <div class="content-wrapper">
6.     <!-- Content Header (Page header) -->
7.     <div class="content-header">
8.         <div class="container-fluid">
9.             <div class="row mb-2">
10.                <div class="col-sm-6">
11. <h1 class="m-0 text-dark">Dashboard Monitoring PLTMH</h1>
12.                </div><!-- /.col -->
13.            </div><!-- /.row -->
14.        </div><!-- /.container-fluid -->
15.    </div>
16. <!-- /.content-header -->
17. <!-- Main content -->
18. <section class="content">
19.     <div class="container-fluid">
20.         <div class="row">
21.             <div class="col-12 col-sm-6 col-md-2">
22.                 <div class="info-box">

```

```

23. <span class="info-box-icon bg-info elevation-1"><i class="fas fa-
    charging-station"></i></span>
24.         <div class="info-box-content">
25.             <span class="info-box-text">Arus Listrik</span>
26.             <span class="info-box-number" id="tegangan">
27.                 {{ $monitor->tegangan }}
28.             </span>
29.             <small>Volt</small>
30.         </div>
31.         <!-- /.info-box-content -->
32.     </div>
33.     <!-- /.info-box -->
34. </div>
35. <!-- /.col -->
36. <div class="col-12 col-sm-6 col-md-2">
37.     <div class="info-box mb-3">
38.         <span class="info-box-icon bg-danger elevation-1"><i class="fas
    fa-water"></i></span>
39.             <div class="info-box-content">
40.                 <span class="info-box-text">Debit Air</span>
41.                 <span class="info-box-number" id="debid">{{ $monitor->debid }}
    </span><small>M3/s</small>
42.             </div>
43.         </div>
44.     </div>
45.     <div class="clearfix hidden-md-up"></div>
46.     <div class="col-12 col-sm-6 col-md-8">
47.         <div class="card">
48.             <div class="card-header">
49.                 <h3 class="card-title">Histori Monitoring PLTMH</h3>
50.                 <button type="submit" class="btn btn-outline-primary btn-
    xs float-right"><i
51.                     class="fas fa-redo" onClick="refreshPage()"></i></button>
52.             </div>
53.             <div class="card-body">
54.                 <form role="form">
55.                     <div class="input-group input-group-sm mb-3">
56.                         <input type="date" class="form-control" name="tgl">
57.                         <input id="appt-time" class="form-control" type="time"
    name="jamAwal" step="2">
58.                         <input id="appt-time" class="form-control" type="time"
    name="jamAkhir" step="2">
59.                     <span class="input-group-append">

```



```

60. <button type="submit" class="btn btn-info btn-flat">Cari</button>
61.     </span>
62.     </div>
63. </form>
64.     <table class="table table-bordered">
65.         <thead>
66.             <tr>
67.                 <th style="width: 10px">#</th>
68.                 <th>Tanggal</th>
69.                 <th>Arus Listrik</th>
70.                 <th>Debit Air</th>
71.             </tr>
72.         </thead>
73.         <tbody>
74.             @forelse ($monitorTable as $mt)
75.                 <tr>
76.                     <td>{{ $loop->iteration }}</td>
77.                     <td>{{ $mt->created_at }}</td>
78.                     <td>{{ $mt->tegangan }} <span class="badge bg-danger">Volt</span>
79.                     </td>
80.                     <td>{{ $mt->debid }}<span class="badge bg-danger">M3/s</span>
81.                     </td>
82.                 </tr>
83.             @empty
84.                 <tr>
85.                     <td colspan="4">Data Tidak Ada</td>
86.                 </tr>
87.             @endforelse
88.         </tbody>
89.     </table>
90. </div>
91. </div>
92.     <!-- /.info-box -->
93. </div>
94. <!-- /.col -->
95. <!-- /.col -->
96. </div>
97. <button type="button" id="alert" class="btn btn-info swalDefaultInfo"
    hidden>
98.     Launch Info Toast
99. </button>
100. </div>
101. </section>

```

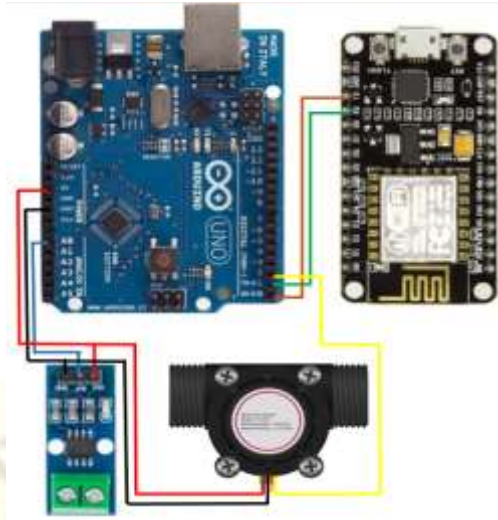
```

102.         <div tabindex="-1" class="modal bs-example-modal-sm"
           role="dialog" aria-hidden="true">
103.             <div class="modal-dialog modal-sm">
104.                 <div class="modal-content">
105.                     <div class="modal-header">
106.                         <h4>Logout <i class="fa fa-lock"></i></h4>
107.                     </div>
108.                 <div class="modal-body"><i class="fa fa-question-circle"></i>
           Anda Yakin Ingin Keluar ?
109.                 </div>
110.                 <div class="modal-footer">
111.                     <a href="{{ route('logout') }}" class="btn btn-danger btn-
           block" btn-block">Logout</a>
112.                 </div>
113.             </div>
114.         </div>
115.     </div>
116.     <!-- /.content -->
117. </div>
118. @endsection
119. @push('scripts')
120.     {{-- <script src="https://code.jquery.com/jquery-
           3.4.1.min.js"></script> --}}
121.     <script type="text/javascript"
           src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.0/jquery.min.js"
           ></script>
122.     <script type="text/javascript">
123.         function getRealData() {
124.             let tegangan = $("#tegangan").text();
125.             let debid = $("#debid").text();
126.             $.ajax({
127.                 url: "{{ URL::to('/reload') }}",
128.                 type: "GET",
129.                 dataType: "json",
130.                 cache: false,
131.                 success: function (res) {
132.                     console.log(res.tegangan);
133.                     console.log(res.debid);
134.                     console.log(res.notification);
135.                     $("#tegangan").text(res.tegangan);
136.                     $("#debid").text(res.debid);
137.                     if (res.notification == "0") {
           $('#alert').addClass('active');

```

```
138.             }else{
139.                 $('#alert').removeClass('active');
140.             }
141.         }
142.     });
143. }
144. $('document').ready(function () {
145.     setInterval(function () {
146.         getRealData();
147.         }, 1000);//request every x seconds
148.     setInterval(function () {
149.         notif();
150.     }, 5000);
151. });
152.     function refreshPage(){
153.         window.history.replaceState(null, null,
154.             window.location.pathname);
155.         window.location.reload();
156.     }
157. </script>
@endpush
```

### 3. Rancang Alat



Gambar 4. 6 Rancang Alat

#### Segmen Program Alat 4. 2 Sinkode *Nodemcu*

```
1. #include <ESP8266WiFi.h>
2. const short int BUILTIN_LED1 = 2;
3. //===== Variabel WiFi =====//
4. const char* ssid      = "mi";
5. const char* password = "09876543210";
6. //===== Variabel Server =====//
7. const char* host = "lcdmonitoring.000webhostapp.com";
8. const int httpPort = 80;
9. String pathfile;
10.
11. String protokol;
12. int sensor1, sensor2, notifikasi;
13.
14. void setup() {
15. Serial.begin(9600);
16. Serial.print("Connecting to ");
17.   Serial.println(ssid);
18.   WiFi.mode(WIFI_STA);
19.   WiFi.begin(ssid, password);
20. while (WiFi.status() != WL_CONNECTED) {
21.     delay(500);
22.     Serial.print(".");
23. } }
```

```

24. Serial.println("IP address: ");
25. Serial.println(WiFi.localIP());
26. }
27. void loop() {
28. if (sensor2 <= 15){
29. notifikasi = 0;
30. if (sensor2 > 15){
31. notifikasi = 1;}
32. if (Serial.available()>0){
33. String datamasuk = Serial.readStringUntil('\n');//membaca data
masuk secara keseluruhan sampai terdapat enter
34. int a = datamasuk.indexOf("A");
35. int b = datamasuk.indexOf("B");
36. int c = datamasuk.indexOf("C");
37. int d = datamasuk.indexOf("D");
38.
39. int fcs = datamasuk.substring(c+1,d).toInt();
40. int jumlahdata = datamasuk.substring(a, c+1).length();
41. if (fcs == jumlahdata && fcs>0 && jumlahdata >0){
42. sensor1 = datamasuk.substring(a+1, b).toInt();
43. sensor2 = datamasuk.substring(b+1, c).toInt();
44. send_data(sensor1,sensor2, notifikasi);
45. }
46. }
47. }
48. void send_data(int a1,int a2, int a3){
49. WiFiClient client;
50. pathfile =
"/public/update?tegangan="+String(a1)+"&debid="+String(a2)+"&notifica
tion="+String(a3);
51. if (!client.connect(host, httpPort)) {
52. Serial.println("connection failed");
53. return;
54. }
55. client.print(String("GET ") + pathfile + " HTTP/1.1\r\n" +
56. "Host: " + host + "\r\n" +
57. "Connection: close\r\n\r\n");
58. unsigned long timeout = millis();
59. while (client.available() == 0) {
60. if (millis() - timeout > 5000) {
61. Serial.println(">>> Client Timeout !");
62. client.stop();
63. return;

```

```

64.     }
65. }
66. }

```

### Segmen Program Alat 4.3 Sinkode Arduino

```

1. String protokol;
2. //-----Sensor Tegangan-----//
3. #define sampling 300
4. #define sampling1 200
5. float banyak_data1;
6. int rata_R;
7. //-----Sensor Flow-----//
8. byte sensorInt = 0;
9. byte flowSensor_pin = 2;
10. float konst = 4.5;
11. float debit_air;
12. int debit_air1;
13. volatile byte count;
14. unsigned int flow_mlt;
15. unsigned long total_volume;
16. unsigned long oldTime;
17. void setup() {
18. Serial.begin(9600);
19. pinMode(flowSensor_pin, INPUT);
20. digitalWrite(flowSensor_pin, HIGH);
21. count          = 0;
22. debit_air      = 0.0;
23. flow_mlt       = 0;
24. total_volume  = 0;
25. oldTime        = 0;
26. attachInterrupt(sensorInt, countPulse, FALLING);
27. }
28. void loop() {
29. //-----Sensor Tegangan-----//
30. for (int i =0; i<=sampling; i++){
31. float R = map(read_teg1(), 3,270, 0, 220);
32. banyak_data1 += R;
33. //-----Sensor Flow-----//
34. if ((millis() - oldTime) > 1000) {
35.     detachInterrupt(sensorInt);
36.     debit_air = ((1000.0 / (millis() - oldTime)) * count) / konst;

```

```

37.     oldTime = millis();
38.     flow_mlt = (debit_air / 60) * 1000;
39.     total_volume += flow_mlt;
40.     unsigned int frac;
41.     debit_air1 = int(debit_air);
42.     attachInterrupt(sensorInt, countPulse, FALLING);
43. }
44. //-----Kirim Data Serial-----//
45. protokol = "A"+String(rata_R)+"B"+String(debit_air1)+"C";
46. protokol = protokol + String(protokol.length())+ "D";
47. Serial.println(protokol);
48. }
49. rata_R = banyak_data1 / sampling;
50. //Serial.print("VR :");
51. //Serial.println(rata_R);
52. banyak_data1=0;
53. }

54. float read_tegl(){
55.     int val_min = 1024;
56.     int val_max = 0;
57.     for (int i=0; i<=sampling; i++){
58.         int val = analogRead(A0);
59.         if (val < val_min ){
60.             val_min = val;
61.         }
62.         if (val > val_max ){
63.             val_max = val;
64.         }
65.     }
66.     return val_max - val_min;
67. }

68. void countPulse(){
69.     count++;
70. }

```



## 4.4 Uji Coba

### 4.4.1 Pengujian Internal (*Black Box Testing*)

**Tabel 4. 3** Hasil Dari *Black Box Testing*

No	Item pengujian	Skenario Pengujian	Hasil Yang Diharapkan	Kesesuaian		Ket.
				Ya	Tidak	
1	Form Login	Klik tombol login	<ul style="list-style-type: none"><li>- Jika salah satu antara username dan password salah maka akan mengualangi proses login.</li><li>- Jika username dan password sesuai maka halaman pindah pada menu utama</li></ul>			
2.	Tampilan <i>Dashboard</i>	Data arus tegangan listrik.	Menampilkan keluaran arus listrik yang di hasilkan PLTMH.			
		Data informasi debit air.	Menampilkan informasi naik turunnya debit air.			
		Data informasi monitorinnng Arus tegangan listrik.	Menampilkan keseluruhan data arus tegangan listrik.			
		Data informasi	Menampilkan keseluruhan data naik turunnya debit air.			

		monitoring debit air.				
--	--	--------------------------	--	--	--	--

