

BAB III METODE PENELITIAN

3.1 ALUR PENELITIAN

3.1.1 Studi Literatur Dan Pengumpulan Bahan

Penelitian ini dilakukan dalam beberapa tahapan diantaranya yaitu studi literatur, perancangan alat, perakitan dan pemrograman alat dengan *Node MCU*, pengujian alat, pengumpulan hasil data. Alur penelitian ini dapat dilihat pada gambar berikut.

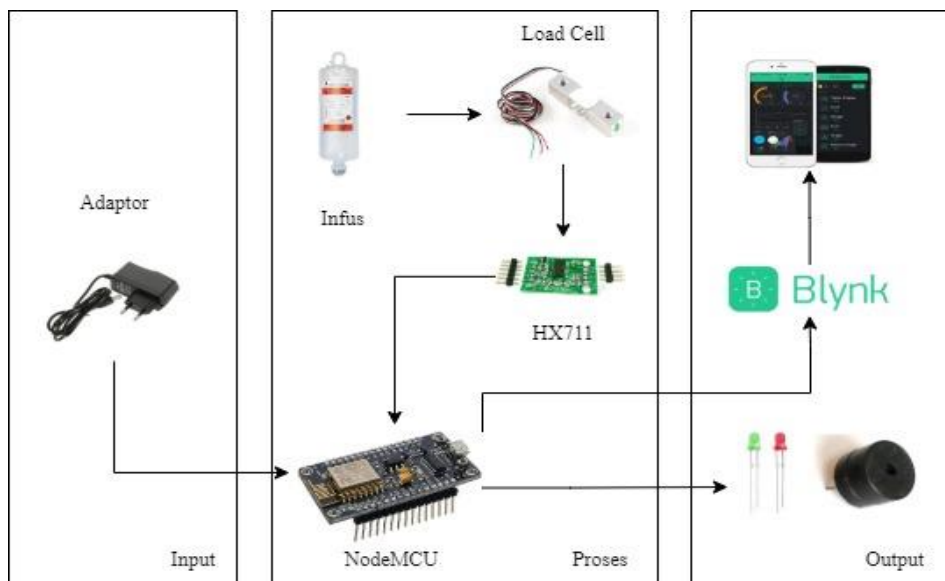


Gambar 3. 1 Blog Diagram Alur Penelitian

Perancangan alur dilakukan untuk mempermudah proses yang nantinya akan dilakukan secara bertahap. Perancangan perangkat keras (*hardware*), perangkat lunak (*software*) dan instalasi program, dimana pada alur ini dilakukan dengan cara membuat *flowchart*. Pada *flowchart* terdapat beberapa *step by step* proses jalannya perancangan alat. Pada perancangan perangkat lunak ini digunakan *software* pemrograman *Arduino IDE*. Kemudian pada pengujian alat, dimana proses ini bertujuan untuk melakukan pengujian perangkat yang telah dibuat sudah sesuai dengan skematik dan dapat berfungsi dengan baik. Pengujian yang telah dilakukan pada setiap sistem dengan mengecek fungsionalitas dan validitas tiap sistem. Langkah selanjutnya yaitu melakukan pengumpulan data dari hasil pengujian yang akan dianalisis.

3.1.2 Blok Diagram Perancangan Sistem

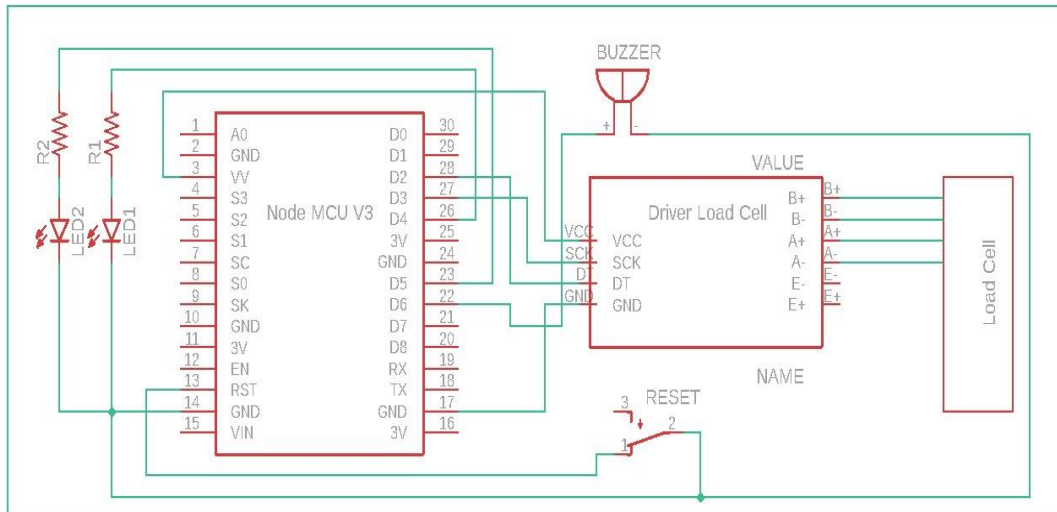
Sesuai pada gambar 3.1 alur penelitian salah satunya yaitu meliputi studi literatur. Dalam proses ini mengumpulkan berbagai macam jurnal sebagai sumber referensi dalam menyusun proposal skripsi ini. Hal ini dilakukan untuk menunjang dasar teori dan acuan dalam menyusun perancangan sistem ini. Sumber dari referensi yang digunakan yaitu berupa jurnal baik ITU-T nasional. Tahap berikutnya yaitu melakukan sebuah perancangan perangkat keras (*hardware*) dan perangkat lunak (*software*) meliputi mikrokontroler, modul pendukung dan aplikasi yang digunakan.



Gambar 3. 2 Blok Diagram Perancangan Sistem

Pada gambar 3.2 Pada tahap penjelasan sistem keseluruhan alat *monitoring* volume cairan infus berbasis *Internet of things*. Dengan dibekali beberapa komponen dan memiliki fungsi tersendiri. Komponen pertama yang digunakan seperti sensor *Load Cell*, driver HX711, akan selalu mengirimkan data kepada *NodeMCU*, kemudian komponen kedua yaitu mikrokontroler *NodeMCU ESP8266* mengambil dan mengirimkan data ke *Blynk server platform*. Setelah menerima data, selanjutnya akan diproses melalui *server firebase* dan diteruskan menuju *output*. *Output* dari sistem yaitu berupa aplikasi pada *android* berupa hasil data pembacaan dari setiap sensor. Aplikasi *android* ini berfungsi untuk *monitoring* volume cairan infus dari jarak jauh. untuk lampu LED sendiri menggunakan ukuran 5mm. led warna hijau akan menyala ketika berhasil terkoneksi ke *Blynk server platform*, begitu juga sebaliknya led warna merah akan menyala ketika belum berhasil terkoneksi atau terhubung ke *Blynk server platform*. Pada saat infus habis maka buzzer akan berbunyi. Dibawah ini pada gambar 3.3 merupakan alur flowchart aplikasi sistem.

3.1.3 Perancangan Perangkat Keras



Gambar 3. 3 Skematik Rangkaian

Pada rangkaian skematik pada gambar 3.3 menggunakan mikrokontroler *NodeMCU* sebagai mikro pemantau sisa volume cairan infus. Pada pin VV terhubung ke driver *Load Cell* pada bagian VCC berfungsi sebagai sumber tegangan 5v 1a dari *NodeMCU* yang akan terhubung ke driver *Load Cell HX711*.

Pada pin D2 di *NodeMCU* terhubung ke pin DT pada driver *Load Cell*. Pada pin D3 *NodeMCU* terhubung ke SCK pada driver *Load Cell* berfungsi untuk *serial clock* dalam menyelaraskan data yang dikirim oleh *Load Cell* ke *NodeMCU ESP8266*. Pada pin D4 terhubung ke resistor R1 berfungsi untuk menahan arus listrik yang selanjutnya menuju ke LED 1. LED 1 disini berwarna merah lalu menuju ke tombol reset pada *NodeMCU ESP8266*. Tombol reset disini berfungsi untuk jika alat dalam keadaan hidup namun tidak dapat terhubung dengan aplikasi *Blynk* maka dilakukan reset. Pada pin D5 terhubung ke resistor R2 dan LED 2. LED 2 yang digunakan berwarna hijau. Ketika aplikasi *Blynk* berhasil terhubung dengan alat ini maka LED 2 akan hidup. Selanjutnya pada pin D6 terhubung dengan *buzzer* di kutub positif dan kutub negatifnya terhubung dengan tombol reset, lalu tombol reset terhubung dengan LED 1. Kemudian pada bagian driver *Load Cell HX711* B+ terhubung ke kabel warna merah. B- terhubung ke kabel warna hitam, A+ terhubung ke kabel warna hijau, dan A- terhubung ke kabel warna putih. Dibawah ini merupakan table konfigurasi mikrokontroler *NodeMCU* dengan sensor driver *Load Cell*.

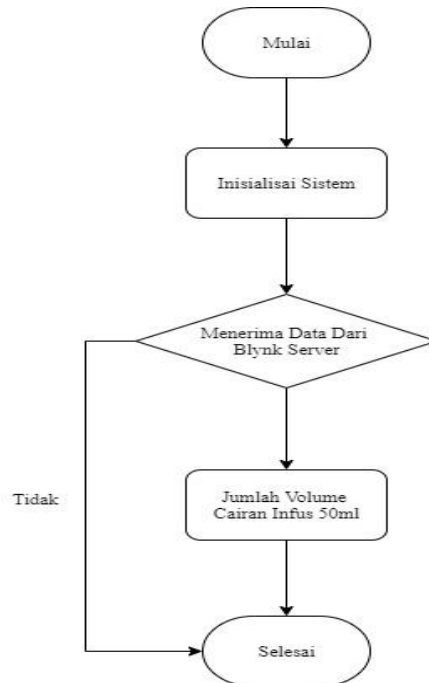
Tabel 3. 1 Konfigurasi *NodeMCU*, HX711

NO	PIN	Fungsi
1	VV	Catu Daya 5 Volt 1 Ampere
2	D3	Serial clock / Pembaca data dari sensor <i>Load Cell HX711</i>
3	D2	Serial Data Output
4	GND	Grounding

Tabel 3.2 menunjukkan konfigurasi Setiap pin yang digunakan oleh *NodeMCU*. Dimana pin VV berfungsi sebagai sumber daya tegangan pada *NodeMCU* yang akan terhubung ke sumber arus atau tegangan positif. Pin *Ground* terhubung ke kutub negatif sebagai *grounding*. Kemudian nilai analog ada pada pin D3 dan bukan dibaca sebagai arus tetapi akan dibaca sebagai tegangan, jadi sensor arus membaca arus yang dirubah ke tegangan dan nilai tegangan tersebut akan dibaca oleh pin D3 di *NodeMCU* tersebut.

3.1.4 Alur Aplikasi *Android*

Pada alur tahapan ini akan dipaparkan sebuah proses atau jalannya suatu aplikasi dalam bentuk flowchart.



Gambar 3. 4 *Flowchart Aplikasi Android*

Pada gambar 3.4 proses pertama yang dilakukan yaitu inisialisasi sistem untuk pemberian nilai awal kondisi saat deklarasi variable. Kemudian untuk proses selanjutnya yaitu menerima data yang dikirimkan oleh *Blynk server platform*, dimana data tersebut berasal dari data yang diperoleh pada mikrokontroler *NodeMCU* dan kemudian akan diteruskan ke aplikasi *Blynk android*. Setelah data diperoleh dari *Blynk server* jika sisa volume cairan infus tinggal 10 ml maka notifikasi akan muncul pada aplikasi *android*.

3.2 ALAT DAN BAHAN

Pada penelitian ini akan membahas mengenai pengujian parameter *Qos* (*Quality of Service*) komunikasi antara sistem yang akan dibuat dengan *user interface* melalui aplikasi berbasis *android*, dalam penelitian yang berjudul “*IMPLEMENTASI SISTEM MONITORING DETEKSI VOLUME INFUS SECARA REAL-TIME MENGGUNAKAN INTERFACE ANDROID*” ini perlu adanya sebuah

sistem dengan penunjang alat dan bahan sesuai dengan kebutuhan, dalam proses pembuatan dan perancangan alat dan bahan yang digunakan meliputi perangkat penyusun *hardware* dan perangkat penyusun *software* yang dapat dilihat pada tabel 3.1.

Tabel 3. 2 Daftar Alat dan Bahan

No	Alat dan Bahan	Jumlah
1	Laptop	1
2	<i>Load Cell</i> 1kg	1
3	Smartphone	1
4	<i>NodeMCU ESP8266</i>	1
5	Software Aplikasi Blynk	1
6	Software Arduino IDE	1
7	LED merah 5mm	4
8	LED hijau 5mm	4
9	PCB bolong IC 18.5 x 7.2 cm	1
10	Spacer 1cm	8
11	Box hitam 10 x 7.5 x 3.5 cm	2
12	Adaptor 5V 1A	1
13	kabel data	1
14	Kabel jumper 1 meter	1
15	Resistor 220 Ohm	1
16	<i>Buzzer</i>	1
17	Modul HX711	1

3.2.1 Laptop

Pada perancangan sistem *Internet of things* ini, laptop digunakan sebagai alat dalam mengolah suatu data, selain itu laptop juga digunakan untuk memberikan pengkodean pada sistem serta sebagai media pengambilan hasil data. Spesifikasi laptop yang digunakan pada tugas akhir ini yaitu prosessor AMD Ryzen 5 2500U kecepatan *clock* sebesar 3,6 GHz, dan RAM *memory* sebesar 8GB *dual channel*.

3.2.2 Smartphone

Pada perancangan sistem *Internet of things* ini, *smartphone* adalah *device* yang digunakan sebagai alat yang mempunyai aplikasi *android* dalam *monitoring* sistem dari perancangan alat. Spesifikasi pada *smartphone* yang digunakan yaitu

processor Exynos 7885, *speed core* sebesar 1,59 GHz, penggunaan OS *android v9 l* PIE, RAM sebesar 4GB.

3.2.3 NodeMCU ESP8266

Pada perancangan sistem *Internet of things* ini, *NodeMCU ESP8266* sebagai mikrokontroler, *NodeMCU* akan digunakan sebagai pengendali utama pada sistem yang akan dibuat. Selain itu *Node MCU* juga digunakan sebagai media untuk pengiriman hasil data ke *server firebase*. Dalam pengiriman data menggunakan media modul *wifi ESP8266* yang telah terpasang pada *NodeMCU* dengan koneksi jaringan *internet*. *NodeMCU* yang akan digunakan sudah dilengkapi *GPIO, PWM, IIC, I-wire*, dan *ADC*.

1.2.4 Software Arduino Ide

Pada perancangan sistem *Internet of things* ini, *software arduino IDE* digunakan untuk memprogram sistem kepada setiap masing-masing perangkat. *Arduino* pemrograman bertugas melakukan fungsi-fungsi yang sudah dinamakan melalui sintaks pemrograman. *Arduino* menggunakan bahasa pemrograman sendiri yang menyerupai bahasa *C++*. Bahasa pemrograman *Arduino Sketch* sudah dilakukan *update* perubahan untuk memudahkan dalam melakukan pemrograman dari bahasa aslinya.

1.2.5 Led

Pada perancangan sistem *Internet of things*, komponen led digunakan untuk lampu indikator peringatan. Lampu led yang digunakan adalah warna merah dan hijau dengan ukuran 5mm. Ketika *Blynk* sudah berhasil terhubung dengan mikrokontroler maka led warna hijau akan menyala. Lampu led warna merah akan menyala ketika cairan infus tersisa 10mL.

1.2.6 Pcb (Printed Circuit Board)

Pada perancangan sistem *Internet of things* ini, PCB yang digunakan adalah PCB bolong IC dengan ukuran 18.5 x 7.2 cm. berfungsi sebagai tempat untuk menempelkan

komponen elektronik agar setiap komponen yang saling terhubung dapat berfungsi sesuai dengan skema yang diinginkan.

1.2.7 Software Aplikasi Blynk

Digunakan untuk kendali modul *Node MCU* melalui jaringan internet seperti wifi.

1.2.8 Spacer

Digunakan untuk mengunci atau mengeratkan tiap-tiap komponen yang akan diletakkan ke PCB.

1.2.9 Adaptor 5V 1A

Pada penelitian alat untuk tugas akhir ini digunakan adaptor 5volt 1ampere sebagai penghubung tegangan arus *Alternating Current (AC)* ke tegangan dengan arus *Direct Current (DC)*.

1.2.10 Box Kotak Hitam

Pada perancangan alat untuk tugas akhir ini menggunakan 2 box kotak hitam dengan ukuran 10 x 7.5 x 3.5 cm digunakan untuk menaruh atau meletakkan rangkaian *monitoring* volume cairan infus.

3.2.11 Load Cell 1kg

Pada perancangan sistem *Internet of things* ini, sensor *Load Cell* 1kg digunakan untuk mengukur massa cairan infus yang tersisa. Kemudian data tersebut masuk ke driver *Load Cell* dan dikirim ke *Node MCU* untuk diproses.

3.2.12 Modul HX711

Pada perancangan sistem *Internet of things* ini, modul HX711 digunakan sebagai ADC untuk mengubah data analog menjadi data digital.

3.2.13 Kabel Data

Pada perancangan sistem *Internet of things* ini, kabel data digunakan sebagai media transmisi tegangan arus listrik yang berasal dari adaptor menuju ke

Node MCU dan juga digunakan untuk menghubungkan antara *Node MCU* ke laptop untuk memasukkan bahasa pemrograman C++.

3.2.14 Resistor

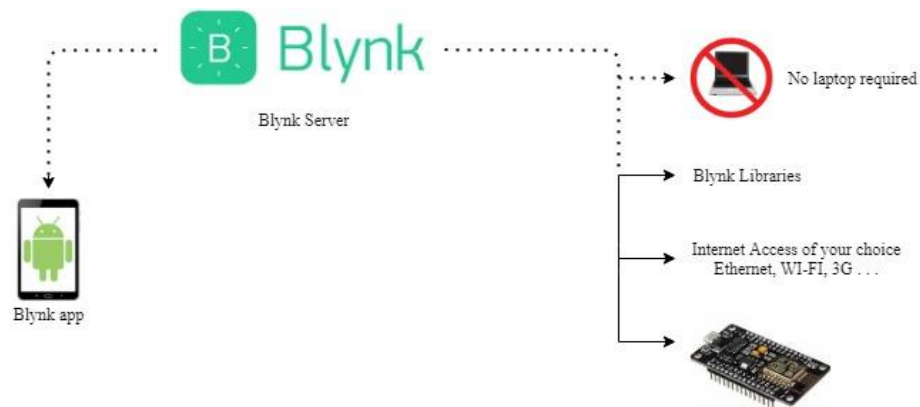
Pada perancangan sistem *Internet of things* ini, Resistor yang digunakan adalah 220 Ohm berfungsi untuk menghambat dan mengatur arus listrik yang masuk dari adaptor ke *Node MCU*.

3.2.15 Buzzer

Pada perancangan sistem *Internet of things* ini, *Buzzer* digunakan untuk memberikan suara bunyi beep ketika berhasil terhubung ke internet.

3.3 PERANCANGAN SOFTWARE

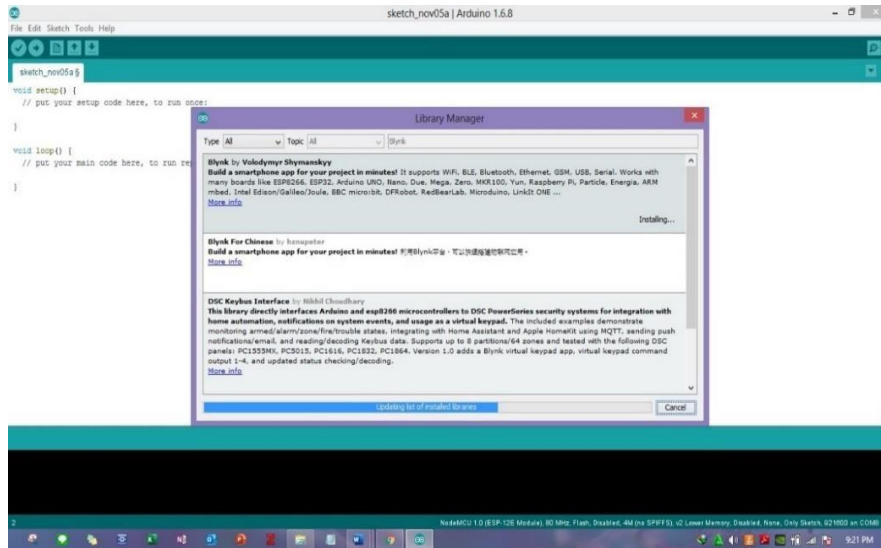
3.3.1 Arsitektur *Blynk*



Gambar 3. 5 Arsitektur *Blynk*

Pada gambar 3.5 yaitu perancangan *software* diantaranya terdapat *Blynk app* sebagai *interface* pada *user*, kemudian *Blynk server* digunakan sebagai *database* cloud antara *Blynk app* dan mikrokontroler. Pada *Blynk library* digunakan sebagai *sketc* program yang akan diinstal dan *NodeMCU* digunakan sebagai otak dari setiap data yang diambil.

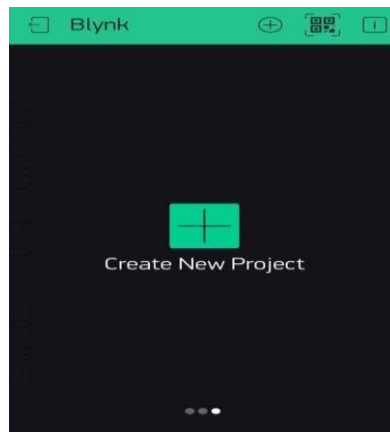
3.3.2 Setting Blynk Pada Arduino



Gambar 3. 6 Instalasi *Blynk* pada Arduino

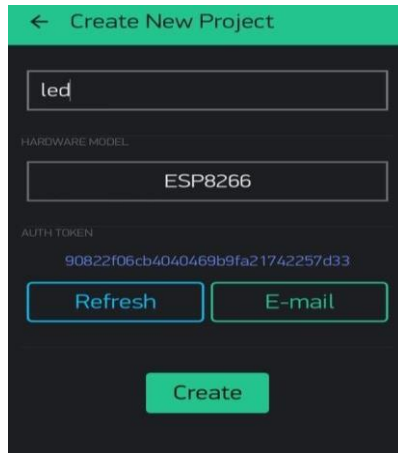
Pada gambar 3.6 adalah proses instalasi *Library Blynk* pada aplikasi *Arduino*. *Library Blynk* digunakan sebagai *sketch* awal program pada saat menghubungkan *Blynk app* dengan mikrokontroler dan *Blynk server* sebagai *database*.

3.3.3 Setting Aplikasi Blynk Pada Smartphone



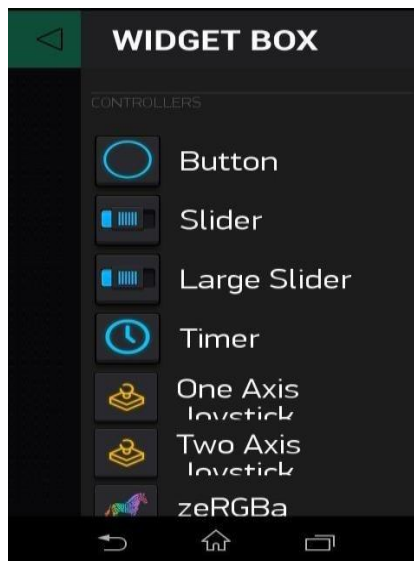
Gambar 3. 7 Tampilan Awal *Blynk Project*

Pada gambar 3.7 adalah tampilan awal untuk proses membuat *project* baru yang akan dibuat dengan cara klik *creat new project*.



Gambar 3. 8 Proses Pembuatan *Blynk* project

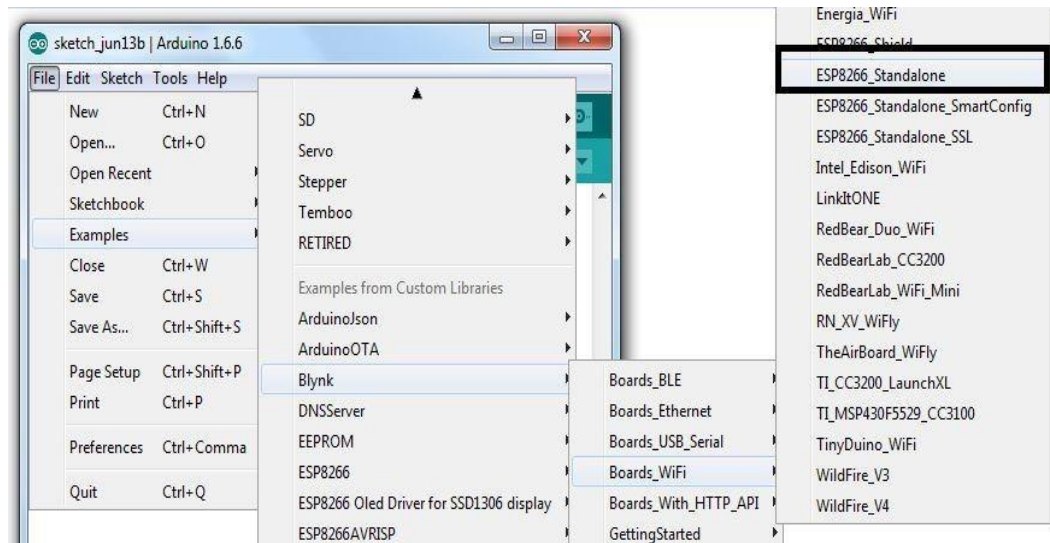
Pada gambar 3.8 adalah proses pembuatan projek. Misalkan membuat projek lampu led yang akan dihubungkan ke modul *wifi ESP8266*, kemudian token digunakan sebagai inisialisasi data base *Blynk server* ke *sketch* program *Blynk*.



Gambar 3. 9 Tampilan *Widget Box Blynk*

Pada gambar 3.9 adalah proses dalam penambahan ikon pada *Blynk* app yang dapat digunakan untuk menghidupkan, mematikan, dan dapat *memonitoring* sistem yang sudah dibuat.

3.3.4 Proses Upload Code



Gambar 3. 10 Proses Upload code Ke Modul ESP8266

Pada gambar 3.10 adalah proses pada saat akan mengupload kode program dengan menggunakan *sketch Blynk* dan memilih *Library* yang akan digunakan. Misalkan menggunakan *Board* untuk koneksi *wifi*, kemudian memilih jenis modul *wifi* ke *ESP8266Standalone*.

3.4 SKEMA PENGUJIAN

3.4.1 Pengujian *Quality of Service (QoS)*



Gambar 3. 11 Skema Pengujian *Quality of Service (QoS)*

Pada gambar 3.11 adalah skema dari pengukuran yang digunakan untuk mendapatkan parameter *Quality of service (QoS)*. Pada Tx terdapat *NodeMCU* yang akan digunakan sebagai sisi dari *transmitter* dan *Access point*. Sedangkan pada Rx (*receiver*) terdapat pada laptop yang sudah diinstal dengan aplikasi *wireshark*. Pengujian yang akan dilakukan pada *QoS* meliputi pengujian *Delay*, *throughput*, dan *Packet Loss*.

3.4.1.1 Delay

Pengujian *Delay* dilakukan bertujuan untuk mengetahui total waktu tunda suatu paket. Pada pengujian *Delay* menguji total selisih waktu yang akan diterima pada sisi Rx ke data terima selanjutnya. Proses pengujiannya dilakukan dengan mengirim data dari sisi Tx ke sisi Rx dengan jumlah data dan jarak yang berbeda. Nilai *Delay* dapat dinyatakan dalam persamaan berikut:

$$\text{Total Delay} = \text{Jumlah Delay seluruh paket} \quad [3.1]$$

3.4.1.2 Packet Loss

Pengujian *Packet Loss* dilakukan bertujuan untuk mengetahui jumlah total paket pengiriman data yang hilang dari sisi Tx ke sisi Rx. Proses pengujian *Packet Loss* dilakukan dengan mengirimkan data dari sisi Tx ke sisi Rx dengan jumlah data yang ada dan jarak yang berbeda. Dimana pada pengujian *Packet Loss* untuk melihat *loss* yang akan terjadi pada proses komunikasi antara Tx ke sisi Rx yang dapat dilihat di *wireshark*.

3.4.1.3 Throughput

Pengujian *Throughput* dilakukan bertujuan untuk mengetahui laju data pada saat proses pengiriman data berlangsung. Proses pengujian *Throughput* pada penelitian ini dilakukan dengan mengirimkan data dari sisi Tx ke sisi Rx dengan waktu dan jarak yang berbeda. Nilai *Throughput* dapat diperoleh dengan menggunakan persamaan berikut:

$$\text{Throughput} \left(\frac{\text{bit}}{\text{s}} \right) = \frac{\text{Ukuran data}}{\text{Total Delay}} \times 8 \quad [3.2]$$