# BAB III

PERANCANGAN ALAT

1. Rekayasa Kebutuhan

Supaya alat dapat berjalan sebagiamana mestinya, maka ada beberapa hal yang dibutuhkan untuk merancang purwarupa sistem *monitoring* dan peringatan jumlah pemakaian air PDAM yang berlebihan berbasis *Internet of Things* yang meliputi kebutuhan perangkat keras dan kebutuhan perangkat lunak.

1. Kebutuhan Perangkat Keras

Dalam perancangan projek ini dibutuhkan beberapa perangkat keras diantaranya terdiri dari sensor *water flow*, mikrokontroler *NodeMCU ESP8266*, LCD, *push button*, *buzzer*, *potensiometer,* *power supply*, *smartphone* dan box plastik. Untuk lebih jelasnya dapat dilihat pada Tabel 3.1.

Tabel 3. Daftar Kebutuhan Perangkat Keras

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Perangkat Keras** | **Jumlah** | **Fungsi** |
| 1 | Sensor *Water Flow* YF-S201 | X1 | Untuk mengukur berapa banyak jumlah air yang mengalir pada sensor ini dam selanjutnya nilai tersebut akan diolah menjadi debit dan volume menggunakan *NodeMCU*. Dalam perancangan projek ini digunakan sensor *water flow* tipe YF-S201 |
| 2 | *NodeMCU ESP8266* V3 *Lolin* | X1 | Pada projek ini, mikrokontroler N*odeMCU* digunakan untuk mengolah hasil pembacaan jumlah aliran air yang dideteksi sensor *water flow* menjadi debit, volume air, pembacaan dari *input push button,* dan *potensiometer*. |

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Perangkat Keras** | **Jumlah** | **Fungsi** |
| 3 | *NodeMCU ESP8266* *LoLin* *Base* | X1 | Ini merupakan sebuah *shield* yang akan dihubungkan dengan *board NodeMCU*. *Shield* ini dibutuhkan untuk menambah jumlah *pin* pada *NodeMCU* |
| 4 | LCD 20x4 | X1 | LCD akan menampilkan beberapa informasi yang telah diproses oleh mikrokontroler *NodeMCU ESP8266* |
| 5 | *Buzzer* | X1 | *Buzzer* pada projek ini digunakan sebagai alarm untuk sistem peringatan pemakaian air yang berlebihan |
| 6 | *Push Button* | X2 | *Push Button* yang pertama digunakan untuk mematikan sistem peringatan jumlah pemakaian air berlebihan yang berupa suara *buzzer.* Untuk *push button* yang satunya digunakan untuk me-*reset* volume air menjadi 0 liter |
| 7 | *Potensiometer* 1KΩ | X1 | *Potensiometer* digunakan untuk mengatur nilai batas volume air untuk mengaktifkan sistem peringatan |
| 8 | *Jack* DC & Saklar | X2 | Pada projek ini dibutuhkan dua buah *jack* DC diantaranya untuk menghubungkan sumber listrik ke *jack* DC *Female* yang nanti nya terhubung ke saklar sehingga dapat memutus dan menyambungkan arus listrik dan akan terhubung ke mikrokontroler *NodeMCU* menggunakan *jack* DC *Male* |
| 9 | *Power Supply* | X1 | Pada projek ini *power supply* yang digunakan memiliki tegangan 9V dengan arus sebesar 2A |

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Perangkat Keras** | **Jumlah** | **Fungsi** |
| 10 | *Smartphone* | X1 | *Smartphone* digunakan untuk membuka aplikasi *blynk*. Untuk sistem operasi bisa menggunakan sistem operasi *Android* atau *iOS* |
| 11 | Box Plastik | X1 | Box plastik dibutuhkan untuk membuat kotak untuk meletakan seluruh komponen yang sudah dirangkaian. |

1. Kebutuhan Perangkat Lunak

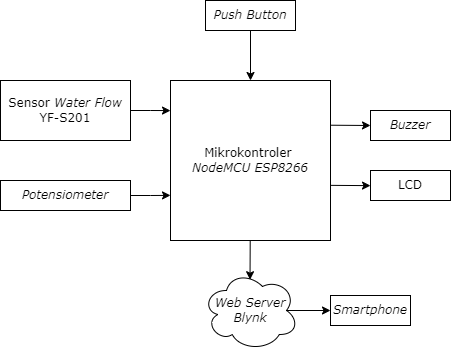
Selain perangkat keras, dibutuhkan juga beberapa perangkat lunak atau aplikasi yang digunakan sehingga perancangan dari alat sesuai dengan yang diinginkan. Untuk lebih jelasnya dapat dilihat pada Tabel 3.2.

Tabel 3. Daftar Kebutuhan Perangkat Lunak

|  |  |  |
| --- | --- | --- |
| **No** | **Perangkat Lunak** | **Fungsi** |
| 1 | *Arduino IDE* | Aplikasi *Arduino IDE* untuk menulis program dan meng-*upload* program (*sketch*) dari komputer ke *board NodeMCU* |
| 2 | *Blynk* | Aplikasi *Blynk* pada projek ini akan di instal pada *smartphone*. Aplikasi ini digunakan untuk melihat informasi yang diolah oleh mikrokontroler *NodeMCU* sehingga informasi tersebut tidak hanya dapat dilihat dari LCD, melainkan dapat dilihat melalui *smartphone.* Dalam perancangan alat ini menggunakan aplikasi *blynk* versi 2.27.32 |

1. Perancangan Alat

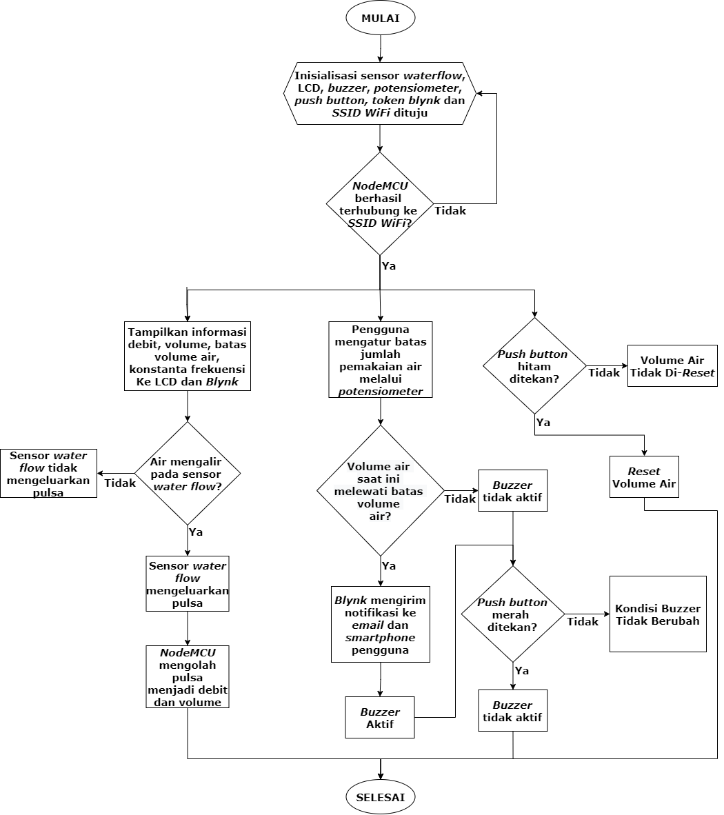
Alat yang akan dirancang adalah sebuah purwarupa sistem *monitoring* dan peringatan jumlah pemakaian air PDAM yang berlebihan berbasis *internet of things*. Konsep dari alat yang dirancang adalah untuk melihat debit air dan jumlah pemakaian air atau volume air secara langsung. Nilai debit dan volume air akan ditampilkan pada alat secara langsung melalui LCD 20x4. Karena alat yang dirancang berbasis *IoT*, maka nilai debit dan volume air juga dapat dilihat dari *smartphone* dengan menggunakan aplikasi *blynk*. Mengenai sistem peringatan jumlah pemakaian air yang berlebihan akan memanfaatkan parameter nilai batas volume air yang diatur oleh pengguna melalui *potensiometer*. Ketika sistem peringatan aktif, maka akan muncul sebuah *buzzer* yang terus menyala sampai ada yang mematikkan nya melalui *push button*. Selain sistem peringatan yang muncul melalui *buzzer*, dibutuhkan juga sistem peringatan berupa notifikasi ke *smartphone* pengguna. Selain itu nilai jumlah pemakaian air atau volume air juga dapat di-*reset* kembali menjadi kondisi awal dengan memanfaatkan sebuah *push button* yang lain. Dari konsep yang diinginkan tersebut, maka dirancanglah sebuah blok diagram yang ditampilkan pada Gambar 3.1.



Gambar 3. Blok Diagram Sistem

Gambar 3.1 menampilkan blok diagram dari sistem alat yang dirancang. Pada blok diagram tersebut terdapat tiga buah *input* berupa *input* dari sensor *water flow*, *push button,* dan *potensiometer*. Sensor *water flow* yang digunakan adalah tipe YF-S201. Sensor *water flow* akan menghitung berapa banyak air yang mengalir melalui sensor. Kemudian untuk mengatur nilai batas volume air dapat melalui *potensiometer.* Pada perancangan alat ini dibutuhkan dua buah *push button*. *Push button* yang pertama berfungsi untuk mematikkan sistem peringatan yang muncul dari *buzzer*. Kemudian *push button* yang kedua berfungsi untuk me-*reset* nilai volume air yang terukur kembali ke 0 Liter. Dari ketiga *input* tersebut akan diproses oleh mikrokontroler *NodeMCU ESP8266*.

Selanjutnya setelah mikrokontroler *NodeMCU ESP8266* menerima *input* tersebut, maka akan diproses sehingga menjadi sebuah sistem peringatan yang aktif dengan menggunakan *buzzer* dan selanjutnya nilai dari debit air, volume air, dan nilai batas pemakaian air yang ditampilkan ke LCD dan ke aplikasi *blynk*. Dari blok diagram tersebut maka disusunlah sebuah *flowchart* yang akan merepresentasikan alur kerja sistem secara keseluruhan yang akan ditampilkan pada Gambar 3.2.



Gambar 3. *Flowchart* Sistem Kerja Keseluruhan

Berdasarkan alur kerja yang ditampilkan pada Gambar 3.2, setelah alat menyala, pertama dilakukan proses inisialisasi dari beberapa komponen yang digunakan diantaranya seperti sensor *water flow*, *buzzer*, *potensiometer*, 2 buah *push button*, *blynk* dan *SSID WiFi* yang dituju. Mikrokontroler *NodeMCU ESP8266* akan mengenali 3 *input*. *Input* sensor *water flow* akan digunakan untuk mengukur debit dan volume air, *input potensiometer* akan diubah menjadi nilai batas pemakaian air sehingga pengguna dapat mengatur nilai batas jumlah pemakaian air atau batas jumlah volume air melalui *potensiometer*, *input push button* merah digunakan untuk mematikkan *buzzer*, dan *input push button* hitam digunakan untuk me-*reset* volume air menjadi 0 Liter. Setelah berhasil terhubung dengan *SSID WiFi* yang dituju, maka LCD dan *blynk* akan menampilkan informasi mengenai debit air, volume air, batas volume air serta kondisi dari *push button* merah apakah sudah ditekan atau belum.

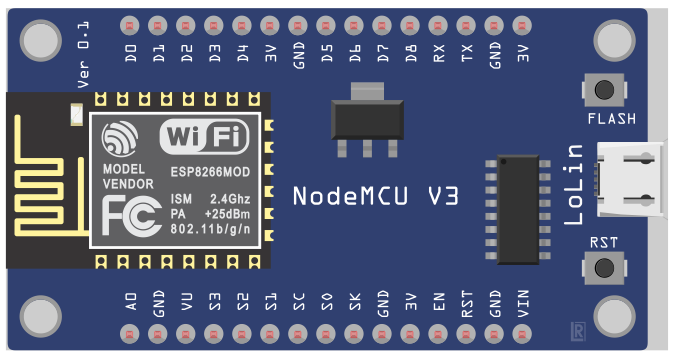
Jika terdapat air yang mengalir pada sensor *water flow*, maka akan membuat sensor *water flow* mengeluarkan pulsa. *NodeMCU* akan mengolah pulsa menjadi debit dan volume air. Untuk mengatur nilai batas jumlah pemakaian air, pengguna dapat mengaturnya melalui *potensiometer*. Jika volume air sudah melewati nilai batas jumlah pemakaian air atau batas volume air, maka akan terdapat dua skenario yang berbeda. Skenario pertama, jika *push button* merah sebelumnya sudah ditekan, maka *buzzer* hanya aktif sebentar dan langsung mati dan *blynk* mengirimkan notifikasi ke *smartphone* pengguna dan ke *email* pengguna sebagai peringatan bahwa jumlah pemakaian air saat ini telah melewati batas jumlah pemakaian air yang diatur oleh pengguna. Dan skenario kedua, jika *push button* merah sebelumnya belum ditekan, maka ketika sistem peringatan aktif, maka *buzzer* akan menyala terus terusan sampai *push button* merah ditekan. Lalu untuk *push button* hitam digunakan untuk me-*reset* volume air kembali menjadi 0 liter, setelah di­­-*reset*, nilai volume air yang terukur sebelum di-*reset* akan dikirimkan ke *email* pengguna.

1. Perancangan Perangkat Keras

Berdasarkan pada Tabel 3.1, ada beberapa komponen perangkat keras yang dibutuhkan. Perancangan perangkat keras dalam pengerjaan projek ini dilakukan secara bertahap dari setiap komponennya.

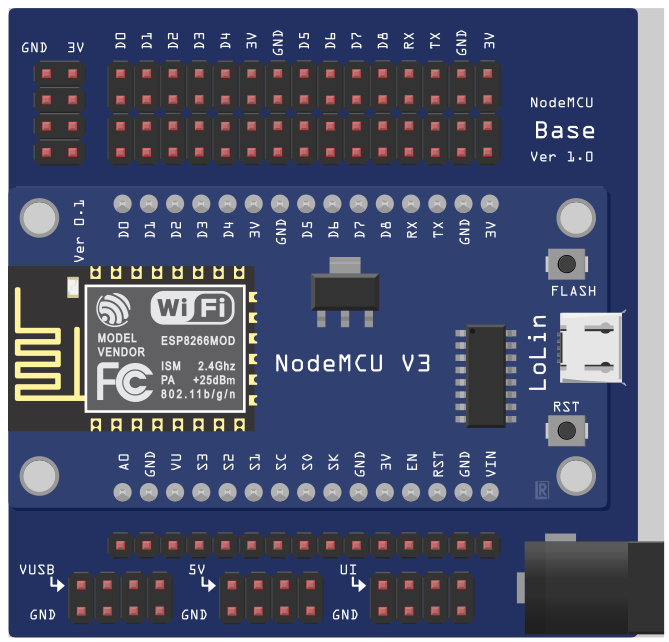
1. Perancangan Skema Rangkaian *NodeMCU ESP8266* V3 *Lolin Base*

Pada perancangan projek ini akan digunakan mikrokontroler *NodeMCU* ESP8266 V3 *LoLin*. Seperti yang ditampilkan pada Gambar 3.3, dapat dilihat bahwa *NodeMCU* hanya memiliki 3 buah pin 3V, 1 buah pin VV dan 1 buah pin VIN.



Gambar 3. Tampilan *Board NodeMCU* V3 *Lolin*

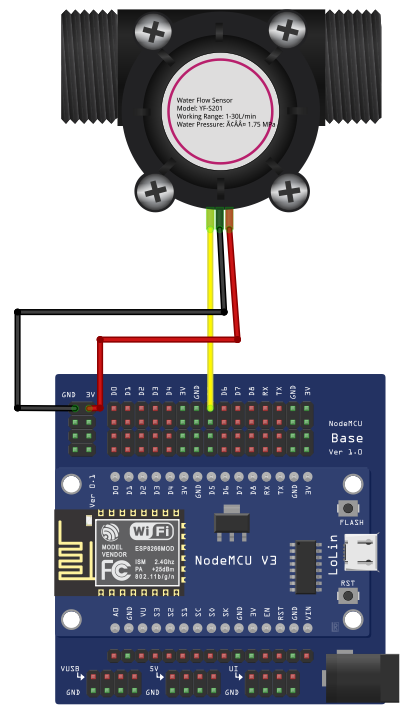
Dikarenakan pada projek ini akan digunakan komponen sensor *water flow*, LCD I2C, *buzzer, potensiometer dan push* button, maka kebutuhan akan pin untuk sumber tegangan dan *ground* dibutuhkan juga pada rangkaian dalam pembuatan projek. Maka dari itu, *NodeMCU* akan dihubungkan dengan sebuah *NodeMCU Base* sehingga dapat menambah jumlah pin untuk sumber tegangan dan *ground*. Gambar 3.4 akan merepresentasikan *NodeMCU* yang dihubungkan dengan *NodeMCU Base*.



Gambar 3. *NodeMCU* yang Dihubungkan Dengan *NodeMCU Base*

1. Perancangan Skema Rangkaian Sensor *Water Flow*

Gambar 3.5 menunjukkan skema rangkaian dalam perancangan alat untuk menghubungkan sensor *water flow* dengan mikrokontroler *NodeMCU*. Skema rangkaian ini dibutuhkan sehingga bisa mendapatkan data mengenai debit air yang didapat dari sensor *water flow* yang kemudian data nya diolah menjadi volume air menggunakan mikrokontroler *NodeMCU.*



Gambar 3. Skema Rangkaian untuk Menghubungkan *NodeMCU Base* dan Sensor *Water Flow*

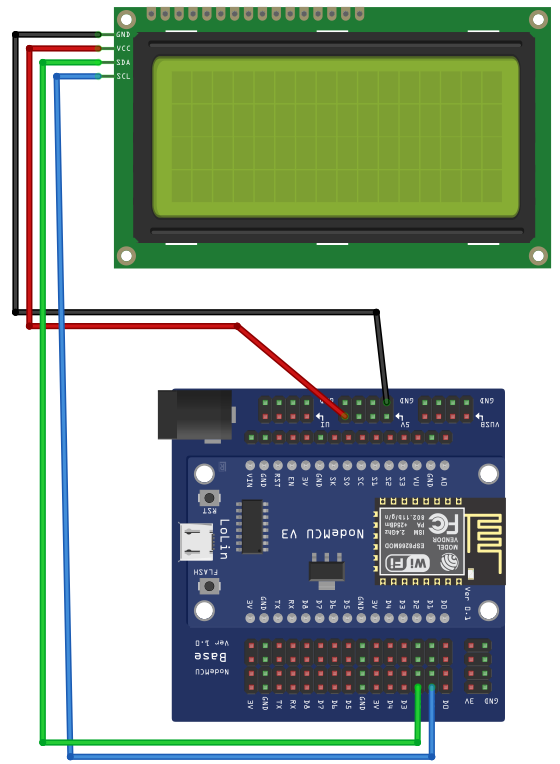
Adapun konfigurasi pin *wiring* dari skema rangkaian sensor *water flow* dan mikrokontroler *NodeMCU* ditunjukkan pada Tabel 3.3.

Tabel 3. Konfigurasi *Wiring NodeMCU* dan Sensor *Water Flow*

|  |  |
| --- | --- |
| ***NodeMCU Base*** | **Sensor *Water Flow*** |
| Pin D5 | *Signal* |
| 3V | VCC |
| GND | GND |

1. Perancangan Skema Rangkaian LCD 20x4 I2C

Gambar 3.6 menampilkan skema rangkaian dalam perancangan alat untuk menghubungkan LCD 20x4 I2C dengan mikrokontroler *NodeMCU*. Skema rangkaian ini dibutuhkan untuk menampilkan informasi mengenai data debit dan volume air yang akan ditampilkan melalui LCD 20x4 I2C.



Gambar 3. Skema Rangkaian Untuk Menghubungkan *NodeMCU Base* dan LCD 20x4 I2C

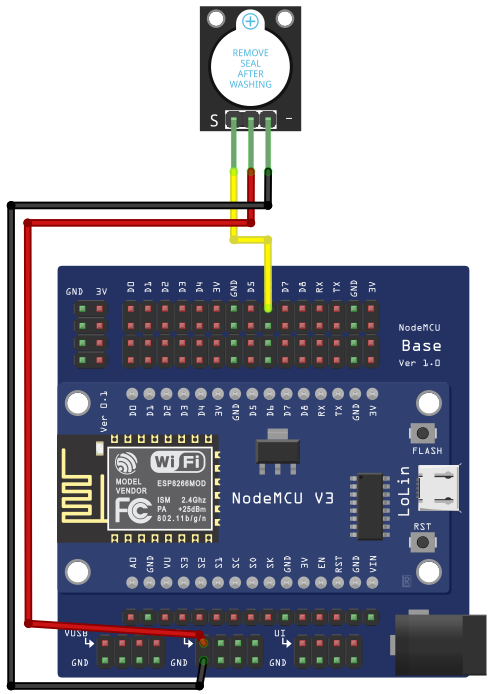
Adapun konfigurasi pin *wiring* dari skema rangkaian LCD 20x4 I2C dan mikrokontroler *NodeMCU* dapat dilihat pada Tabel 3.4 dibawah ini.

Tabel 3. Konfigurasi *Wiring NodeMCU* dan LCD I2C

|  |  |
| --- | --- |
| ***NodeMCU Base*** | **LCD 20x4 I2C** |
| D2 | SDA |
| D1 | SCL |
| 3V | VCC |
| GND | GND |

1. Perancangan Skema Rangkaian *Buzzer*

Pada perancangan alat di projek ini membutuhkan sebuah *buzzer*. *Buzzer* pada alat ini akan digunakan sebagai sistem peringatan. Untuk skema rangkaian nya dapat dilihat pada Gambar 3.7.



Gambar 3. Skema Rangkaian Untuk Menghubungkan *NodeMCU Base* dan *Buzzer*

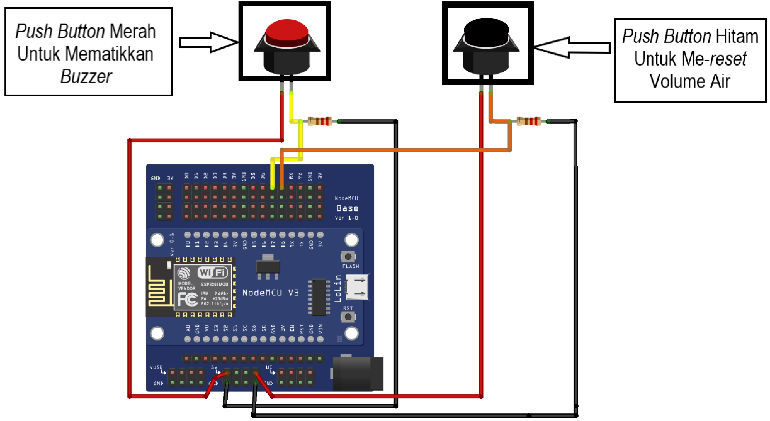
Untuk konfigurasi *wiring* dari rangkaian skematik *buzzer* dan mikrokontroler *NodeMCU* dapat dilihat pada Tabel 3.5.

Tabel 3. Konfigurasi *Wiring NodeMCU* dan *Buzzer*

|  |  |
| --- | --- |
| ***NodeMCU Base*** | ***Buzzer*** |
| D4 | *Signal* |
| 5V | VCC |
| GND | GND |

1. Perancangan Skema Rangkaian *Push Button*

Dalam perancangan projek ini dibutuhkan dua buah *push button*. *Push button* yang pertama adalah *push button* merah digunakan untuk mematikan *buzzer* yang menyala. Untuk *push button* kedua adalah *push button* hitam digunakan untuk me-*reset* volume air. Untuk skema rangkaian yang digunakan dapat dilihat pada Gambar 3.8.

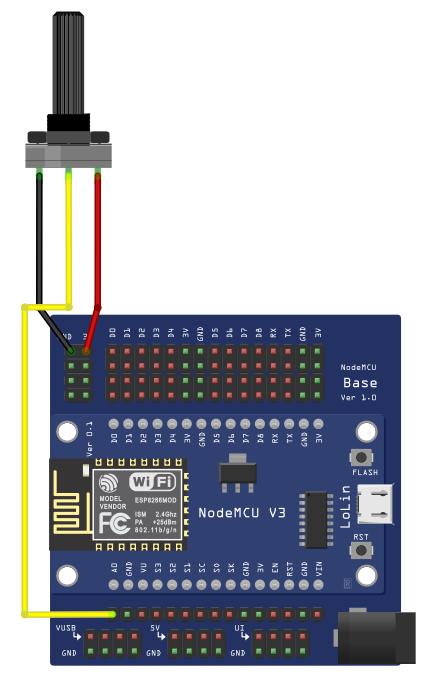


Gambar 3. Skema Rangkaian *NodeMCU Base* dan *Push Button*

Gambar 3.8 menampilkan skema rangkaian untuk menghubungkan mikrokontroler *NodeMCU Base* dan *push button*. Untuk *push button* merah pada kaki sebelah kiri dihubungkan ke sumber listrik 5V. Kemudian untuk kaki sebelah kanan dihubungkan ke *resistor* 220Ω dan dihubungkan juga ke pin D7. Untuk kaki *resistor* yang satunya dihubungkan ke GND. Untuk *push button* hitam pada kaki kiri dihubungkan ke sumber listrik 5V. Kemudian untuk kaki sebelah kanan dihubungkan ke *resistor* 220Ω dan dihubungkan juga ke pin D8. Untuk kaki *resistor* yang satunya dihubungkan ke GND.

1. Perancangan Skema Rangkaian *Potensiometer*

Dalam perancangan ini dibutuhkan sebuah *potensiometer* untuk mengatur nilai batas volume pemakaian air untuk sistem peringatan jumlah pemakaian air berlebihan. Untuk skema rangkaiannya dapat dilihat pada Gambar 3.9.

****

Gambar 3. Skema Rangkaian *NodeMCU Base* dan *Potensiometer*

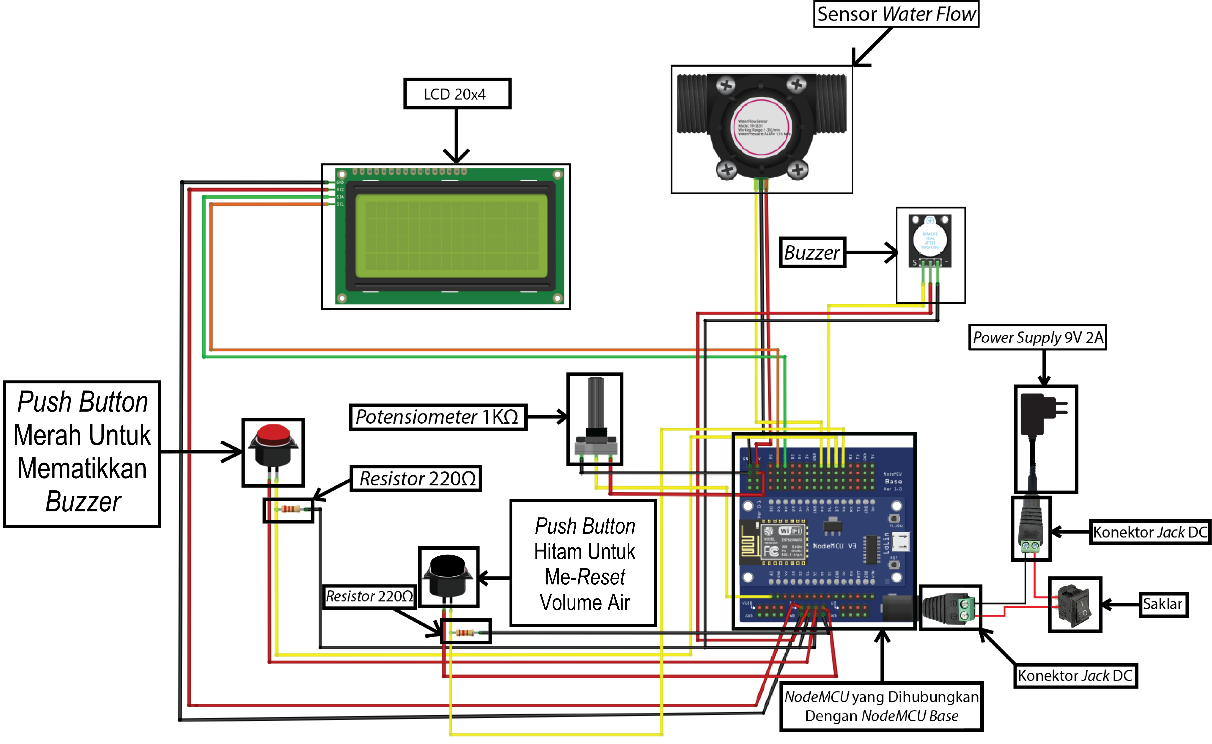
Pada gambar 3.9 adalah skema rangkaian untuk menghubungkan *potensiometer* ke mikrokontroler *NodeMCU Base*. Untuk jenis *potensiometer* yang digunakan pada skema rangkaian ini adalah *potensiometer* jenis *trimmer* 1KΩ. Untuk konfigurasi *wiring* dari skema rangkaian diatas dapat dilihat pada Tabel 3.6.

Tabel 3. Konfigurasi *Wiring NodeMCU* dan *Potensiometer*

|  |  |
| --- | --- |
| ***NodeMCU Base*** | ***Potensiometer*** |
| 3V | VCC |
| A0 | *Signal* |
| GND | GND |

1. Perancangan Skema Rangkaian Keseluruhan

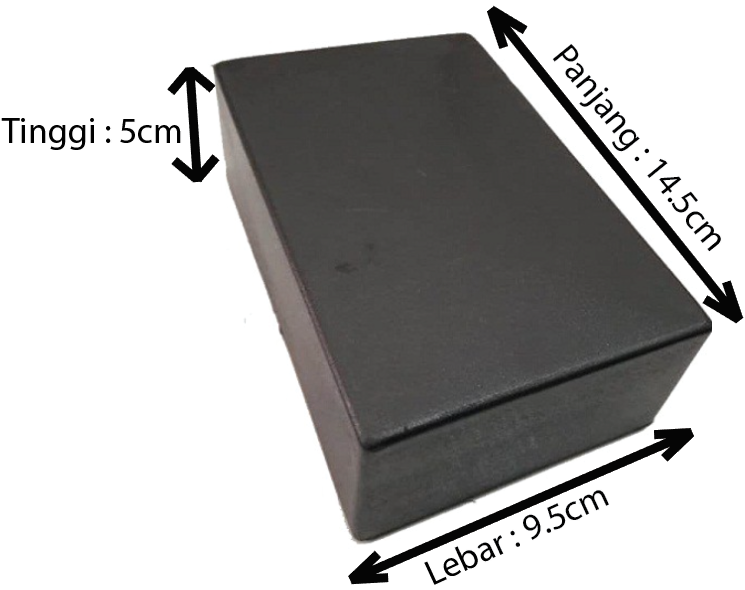
Pada perancangan skema rangkaian keseluruhan ini adalah menggabungkan seluruh rangkaian dari komponen perangkat keras yang telah dirangkai sebelumnya. Namun pada skema rangkaian keseluruhan ini terdapat beberapa komponen tambahan diantaranya adalah dua buah konektor *jack DC*, saklar dan *power supply*. Untuk konektor *jack DC* akan disambung ke *barrel jack* pada *NodeMCU Base*, kemudian *jack DC* yang sudah disambung akan diberi kabel yang terhubung ke sebuah saklar sehingga dapat menyambung dan memutus aliran listrik ke *NodeMCU*. Selanjutnya saklar akan disambung lagi ke sebuah konektor *jack DC* yang akan dihubungkan ke sumber listrik yaitu *power supply*. Untuk *power supply* yang digunakan adalah *power supply* dengan tegangan 9V dan arus sebesar 2A. Untuk gambar dari rangkaian keseluruhan dapat dilihat pada Gambar 3.10.



Gambar 3. Skema Rangkaian Keseluruhan

1. Perancangan *Casing*

Setelah perancangan skema rangkaian sudah selesai, maka rangkaian dari alat akan dibuatkan sebuah *casing* untuk meletakkan seluruh komponen tersebut. Untuk ukuran kotak yang digunakan menggunakan sebuah kotak berbentuk persegi dengan dimensi panjang 14.5cm, lebar 9.5cm dan tinggi 5cm. Selanjutnya kotak tersebut akan dilubangi pada beberapa bagian untuk meletakkan beberapa komponen yang masing – masing dibuatkan tempat meletakkan komponen tersebut.



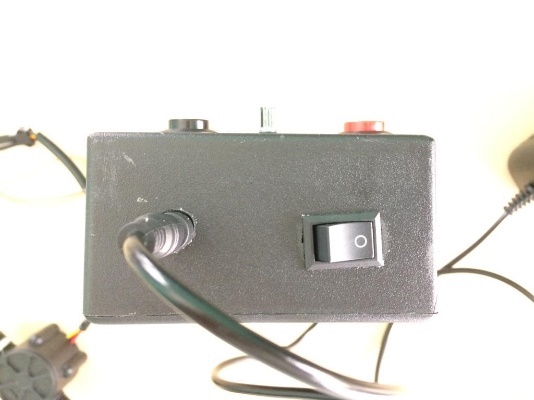
Gambar 3. Bentuk Awal Kotak Sebelum Dibuat *Casing*

Pada bagian depan kotak dibuatkan sebuah lubang persegi dengan diameter 10x4cm guna untuk meletakkan sebuah LCD 20x4. Selanjutnya dibuatkan juga lubang di masing – masing sudut dekat lubang persegi tersebut untuk memasangkan baut pada *board* LCD ke kotak. Untuk komponen berikutnya dibuatkan juga masing – masing lubang untuk meletakkan dua buah *push button*, *potensiometer*, *jack* DC, *buzzer*, dan lubang untuk memasukkan kabel sensor *water flow* yang terhubung ke mikrokontroler didalam kotak, dan juga sebuah lubang berbentuk persegi untuk meletakkan sebuah saklar.



Gambar 3. Tampilan Depan Dari Alat

Seperti yang ditampilkan pada Gambar 3.12, dibagian depan dari alat terdapat 3 buah komponen utama yaitu LCD 20x4, 2 buah *push button* yang pertama *push button* berwarna merah berguna untuk mematikkan *buzzer*, dan *push button* kedua yang berwarna hitam berguna untuk me-*reset* volume air menjadi 0 liter. Ditengah dari kedua *push button* tersebut terdapat sebuah *potensiometer* yang digunakan untuk mengatur berapa batas jumlah volume air untuk sistem peringatan jumlah pemakaian air berlebihan.



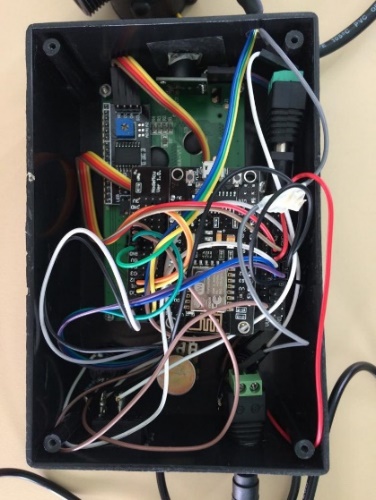
Gambar 3. Tampilan Samping Kanan Dari Alat

Gambar 3.13 menampilkan tampilan samping kanan dari alat yang dirancang. Dibagian samping kanan dari alat terdapat dua buah komponen yaitu sebuah *jack* DC dan saklar. *Jack* DC berguna untuk menghubungkan alat ke sumber tegangan. *Jack* DC juga terhubung ke saklar sehingga pengguna dapat menyala dan mematikkan alat melalui saklar tersebut.



Gambar 3. Tampilan Samping Kiri Dari Alat

Pada bagian samping kiri dari alat hanya terdapat dua buah lubang yaitu lubang pertama untuk tempat keluarnya suara yang muncul dari *buzzer*, dan untuk lubang yang kedua untuk memasukkan kabel dari *sensor water flow* yang dimana kabel tersebut terhubung ke mikrokontroler *NodeMCU* yang ada didalam *casing* tersebut. Di bagian belakang terdapat sebuah penutup dan ketika penutup tersebut dibuka maka terlihat rangkaian dari seluruh komponen yang digunakan



Gambar 3. Tampilan Dalam Alat

1. Perancangan Perangkat Lunak Sistem

Perancangan perangkat lunak sistem adalah berbagai perancangan yang digunakan untuk memprogram perangkat keras yang sudah dirangkain sehingga alat dapat bekerja dengan semestinya. Bagian ini akan meliputi berbagai perancangan untuk memprogram perangkat keras diantaranya perancangan untuk sistem pembacaaan debit dan volume, sistem untuk mengatur batas jumlah volume air dan sistem peringatan jumlah pemakaian air berlebihan.

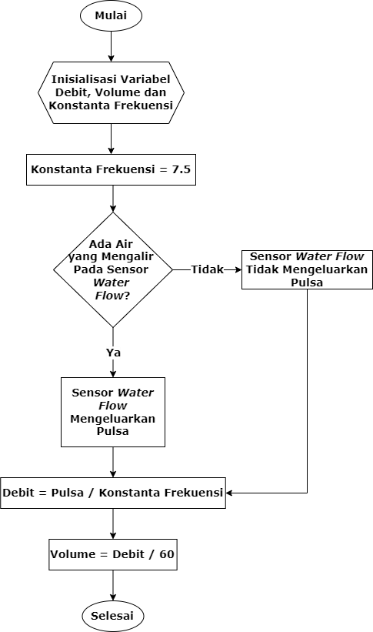
1. Perancangan Sistem Pengukuran Debit dan Volume Air

Sistem pada alat ini akan menggunakan sensor *water flow* untuk mengukur debit dan volume air. Sensor *water flow* memiliki prinsip kerja dengan memanfaatkan *hall effect*. Prinsipnya adalah ketika ada air yang mengalir melalui katup plastik pada sensor, maka akan membuat rotor berputar sehingga medan magnet yang terdapat pada rotor akan memberikan efek terhadap sensor *hall effect* dan akan menghasilkan sinyal pulsa berupa tegangan. Perancangan program sensor *water flow* dibutuhkan untuk mengubah pulsa yang dihasilkan oleh sensor *water flow* untuk diolah menjadi debit dan volume air. Untuk memperoleh nilai dari debit dan volume air dari sensor *water flow* tersebut diperoleh melalui rumus berikut :

Debit =

Volume =

Ketika *sensor water flow* dialiri oleh air sehingga sensor *water flow* menghasilkan sebuah pulsa, pulsa tersebut akan diolah untuk mengukur nilai debit dengan cara membagi pulsa dengan konstanta frekuensi. Setiap sensor *water flow* memiliki nilai konstanta frekuensi yang berbeda – beda untuk mengukur debit air. Dalam perancangan alat ini, sensor *water flow* yang digunakan adalah sensor *water flow* dengan tipe YF-S201, menggunakan konstanta frekuensi yaitu 7.5. Dari kedua rumus tersebut, disusunlah *flowchart* seperti yang ditampilkan pada Gambar 3.16.

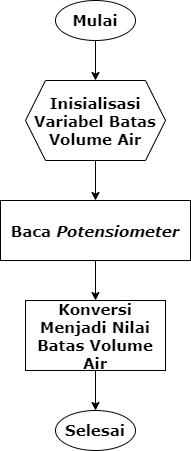


Gambar 3. *Flowchart* Perhitungan Debit dan Volume

Dari *flowchart* yang ditampilkan pada Gambar 3.16, dapat dilihat bahwa alur dari prinsip kerja untuk program sensor *water flow* diawali dengan melakukan inisialisasi variabel debit, volume dan konstanta frekuensi. Nilai konstanta frekuensi diatur menjadi 7.5 sesuai dengan spesifikasi dari sensor *water flow* tipe YF-S201. Selanjutnya diikuti dengan percabangan apakah ada air yang mengalir pada sensor *water flow* atau tidak. Jika tidak ada air yang mengalir, maka sensor *water flow* tidak mengeluarkan pulsa. Namun jika ada air yang mengalir pada sensor *water flow* maka sensor *water flow* akan mengeluarkan pulsa. Ketika sensor *water flow* sudah mengeluarkan pulsa, maka nilai pulsa tersebut akan diolah menjadi nilai debit dengan cara nilai pulsa dibagi dengan nilai konstanta frekuensi yaitu dibagi dengan 7.5. Setelah mendapatkan nilai dari debit selanjutnya diikuti dengan mencari nilai volume dengan cara nilai debit yang sudah didapatkan dibagi dengan 60. 60 disini adalah 60 detik karena satuan dari sensor *water flow* adalah liter per menit dan nilai volume akan dijumlahkan dengan volume periode berikutnya sehingga dapat menghasilkan total volume air.

1. Perancangan Sistem Mengatur Batas Volume Air

Perancangan program untuk mengatur batas volume air diperlukan dalam perancangan ini untuk sistem peringatan jumlah pemakaian air yang berlebihan. Prinsip kerja dari sistem ini adalah pengguna mengatur berapa batas volume pemakaian air untuk sistem peringatan pada alat ini. Pengguna dapat mengatur batas jumlah pemakaian air melalui *potensiometer*. Untuk penentuan *range* dari nilai batas volume air adalah dapat diatur dengan nilai maksimum yaitu 300 Liter. Untuk *flowchart* dari program ini dapat dilihat pada Gambar 3.17.

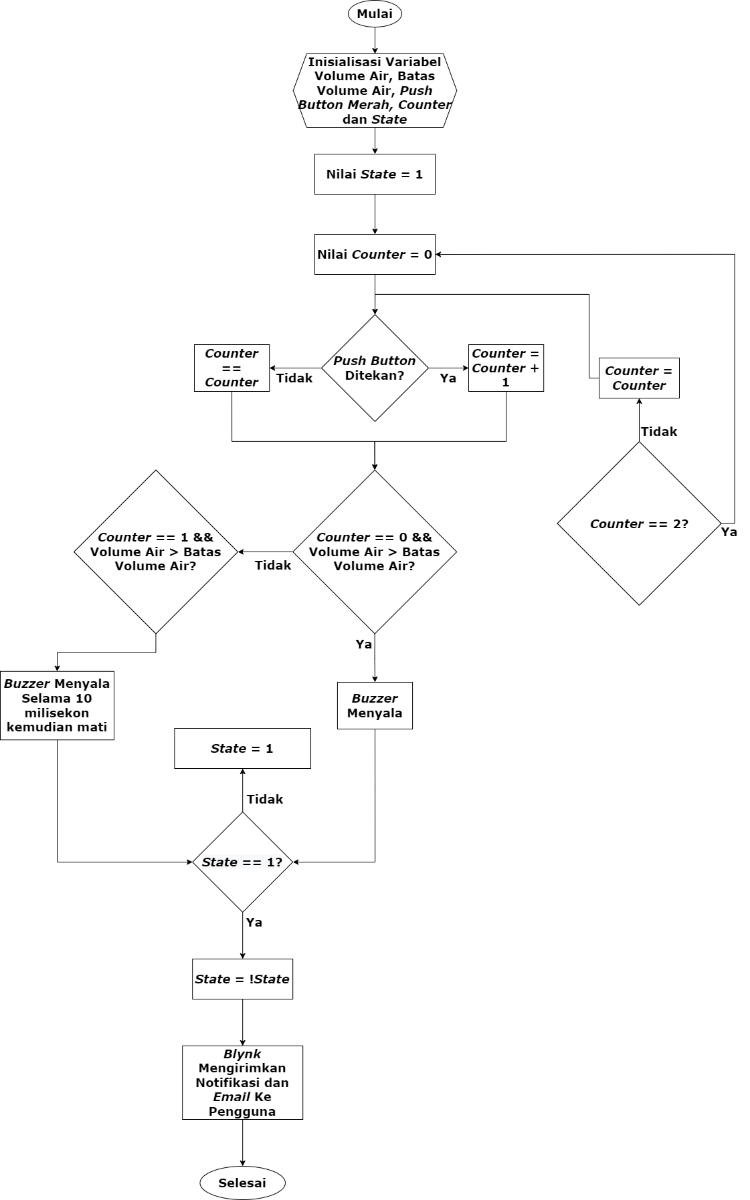


Gambar 3. *Flowchart* Program Mengatur Batas Volume Air

Pada *flowchart* tersebut, pertama hal yang dilakukan dari program sistem ini adalah inisialisasi variabel batas volume air. Nilai batas volume air ini dibaca berdasarkan kondisi dari *potensiometer*. Ketika *potensiometer* diputar kearah kiri maka nilai dari batas volume air menjadi mengecil dan ketika *potensiometer* diputar kearah kanan maka nilai dari batas volume air menjadi membesar.

1. Perancangan Sistem Peringatan Pemakaian Air Berlebihan

Prinsip dari sistem peringatan pemakaian air berlebihan akan memanfaatkan bunyi *buzzer* dan notifikasi dari aplikasi *blynk*. Ketika pengguna sudah mengatur nilai batas volume air melalui *potensiometer*, maka nilai batas volume air tersebut adalah acuan untuk aktif atau tidaknya sistem peringatan. Ketika volume air yang terukur sudah melewati batas volume air, maka sistem peringatan akan aktif. Namun ketika volume air yang terukur tidak melewati batas volume air, maka pemakaian air dianggap hemat sehingga sistem peringatan tidak aktif. Berikut *flowchart* untuk program sistem peringatan pemakaian air berlebihan.

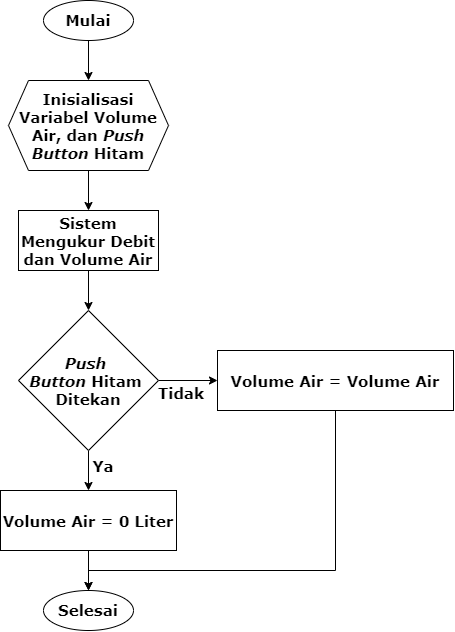


Gambar 3. *Flowchart* Sistem Peringatan Pemakaian Air Berlebihan

Gambar 3.18 menampilkan sebuah *flowchart* untuk sistem peringatan jumlah pemakaian air berlebihan. Dapat dilihat bahwa pada kondisi awal terdapat dua buah variabel tambahan yaitu *state* dan *counter*. Variabel *state* digunakan untuk pengkondisian pada saat sistem peringatan aktif agar aplikasi *blynk* hanya mengirimkan email dan notifikasi ke pengguna sekali saja. Karena jika variabel *state* tersebut dihapus, maka jika sistem peringatan sedang aktif maka *blynk* akan secara terus menerus mengirimkan *email* dan notifikasi ke pengguna tanpa henti sampai sistem terputus dari internet atau mati. Sedangkan variabel *counter* digunakan untuk pengkondisian pada *push button*. *Push button* yang digunakan dalam program ini adalah *push button* yang berwarna merah untuk mematikkan sistem peringatan yang muncul dari *buzzer*. Jika *push button* merah ini ditekan sekali maka akan menambahkan nilai *counter* menjadi 1. Dalam kondisi awal nilai *counter* adalah 0. Jika nilai *counter* adalah 0 maka ketika sistem peringatan aktif, *buzzer* akan menyala terus sampai *push button* ditekan sehingga membuat *counter* menjadi 1. Ketika nilai *counter* 1 dan sistem peringatan aktif, maka *buzzer* yang tadinya menyala akan mati. Namun ketika *counter* bernilai 1 dan sistem peringatan belum aktif, maka *buzzer* tetap mati dan ketika nanti sistem peringatan aktif, *buzzer* hanya akan menyala selama 10 milisekon kemudian mati. Ketika *push button* merah ditekan lagi sampai nilai *counter* adalah 2, maka nilai *counter* akan kembali diatur menjadi 0.

1. Perancangan Sistem *Reset* Volume Air

Sistem *reset* volume air ini digunakan untuk mengatur kembali nilai volume air yang sudah terukur menjadi 0 liter. Untuk bentuk dari *flowchart* dapat dilihat pada Gambar 3.19.



Gambar 3. *Flowchart* Sistem *Reset* Volume Air

Pada *flowchart* tersebut, sistem ini akan memanfaatkan kondisi dari *push button* hitam. Ketika *push button* hitam ditekan maka akan membuat nilai volume air diatur menjadi 0 liter. Namun jika *push button* hitam tidak ditekan maka nilai volume air tidak berubah.

1. Perancangan Sistem Tampilan Pada LCD

Setelah seluruh perangkat keras berhasil di program sehingga dapat bekerja dengan semestinya, maka dilanjutkan juga dengan perancangan sistem tampilan pada LCD. LCD akan menampilkan beberapa informasi seperti debit air, volume air dan batas volume air. Ketika pertama kali alat dinyalakan, maka akan muncul sebuah tulisan berupa “PURWARUPA MONITORING DAN PERINGATAN JUMLAH PEMAKAIAN AIR PDAM YANG BERLEBIHAN”.



Gambar 3. Tampilan Awal LCD Ketika Alat Pertama Kali Dinyalakan

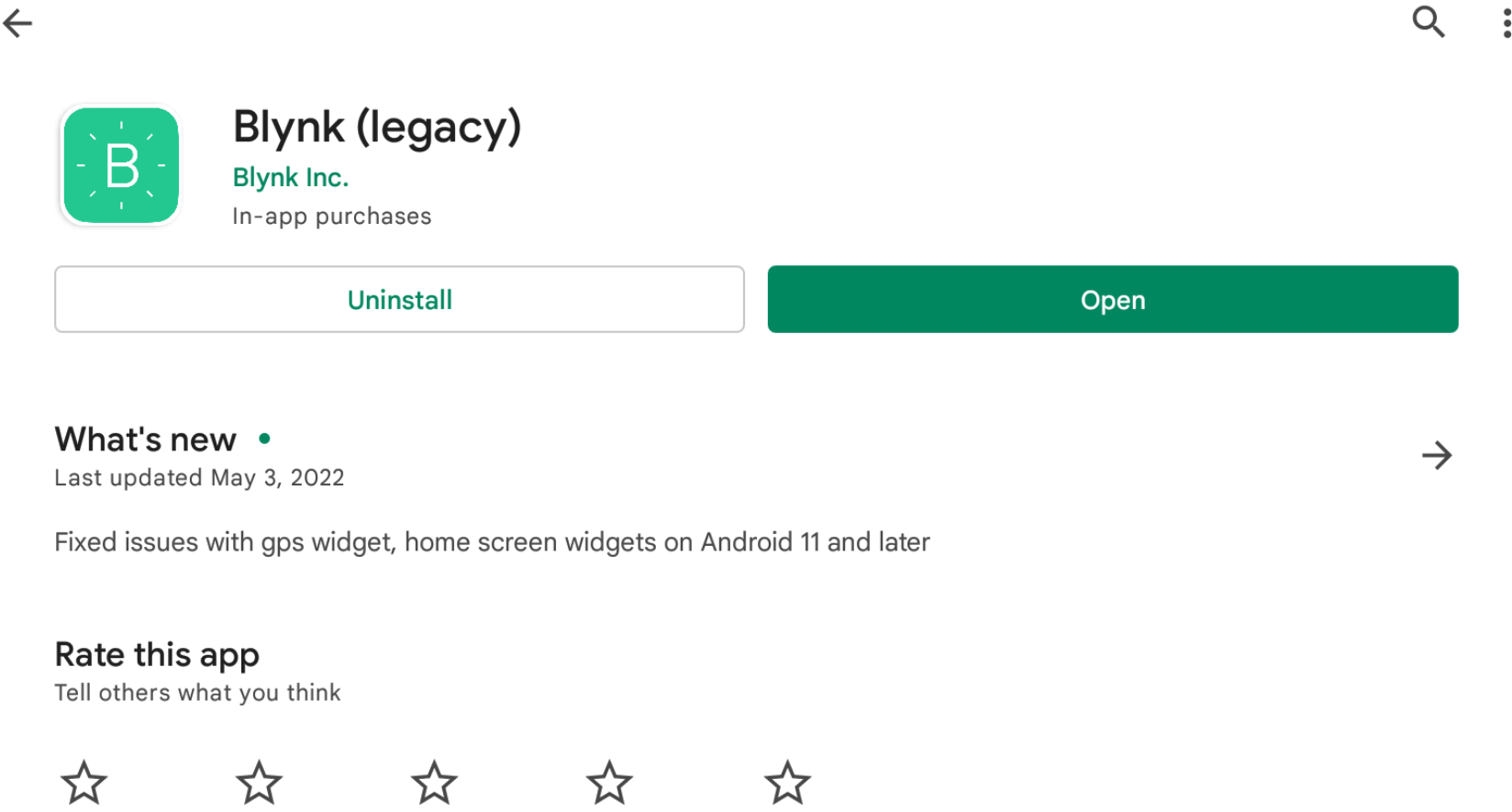
Setelah tampilan awal tersebut ditampilkan, berikutnya tampilan LCD akan berubah yang menampilkan tampilan utama. Pada tampilan utama, LCD akan menampilkan tiga informasi yaitu nilai debit air, volume air dan batas volume air.



Gambar 3. Tampilan Utama LCD

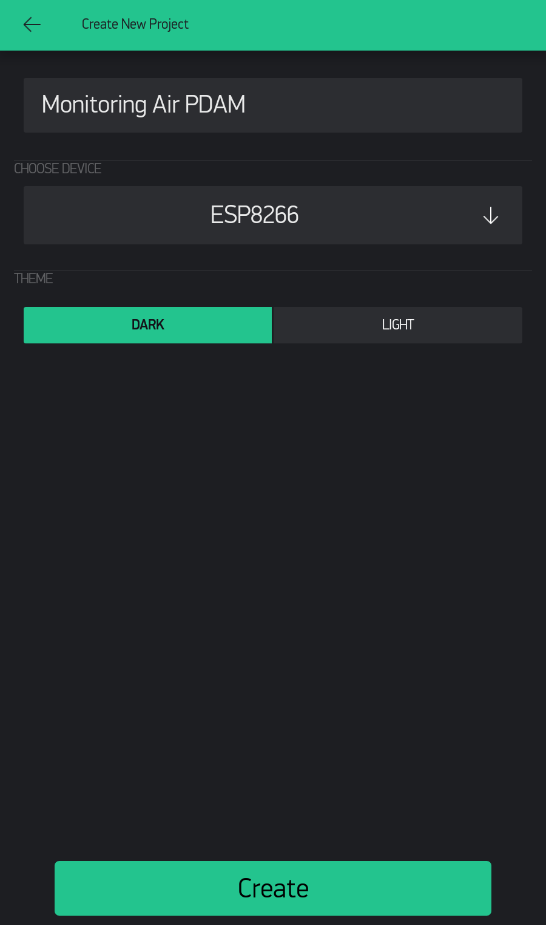
1. Perancangan Sistem Menghubungkan Alat Dengan Aplikasi *Blynk*

Setelah berhasil merancang sistem perangkat keras dan memprogram sebagaimana mestinya, selanjutnya adalah merancang sistem untuk menghubungkan alat dengan aplikasi *blynk*. Pertama *download* terlebih dahulu aplikasi *blynk* pada *smartpone* yang digunakan. Dalam perancangan projek ini, *smartphone* yang digunakan adalah *smartphone* menggunakan sistem operasi *android*.



Gambar 3. Aplikasi *Blynk* di *Play* *Store*

Seperti pada Gambar 3.22 adalah aplikasi *blynk* yang akan di unduh di *play store* dikarenakan *smartphone* yang digunakan adalah menggunakan sistem operasi *android*. Setelah berhasil mengunduh *blynk*, selanjutnya adalah mencoba membuat projek baru pada aplikasi *blynk*.



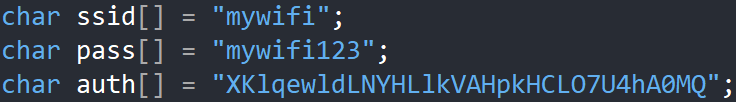
Gambar 3. Membuat Projek Baru di Aplikasi *Blynk*

Pada bagian *Create New Project* Nama projek diberi nama “Monitoring Air PDAM”. Selanjutnya bagian *Choose Device* menggunakan *ESP8266* dikarenakan perancangan projek ini menggunakan mikrokontroler *NodeMCU ESP8266*. Berikutnya, aplikasi *blynk* akan mengirimkan sebuah pesan ke *email* dari akun *blynk* yang sedang *login*.



Gambar 3. *Blynk* Mengirim *Auth* *Token* ke *Email* Pengguna

Seperti pada Gambar 3.24, aplikasi *blynk* mengirimkan sebuah *auth token* ke *email* pengguna atau *email* yang digunakan untuk *login* ke aplikasi *blynk*. *Auth token* tersebut akan dimasukkan ke aplikasi *Arduino IDE* sehingga projek yang telah dibuat pada aplikasi *bylnk* dapat diprogram sebagaimana mestinya. Selanjutnya adalah merancang sistem untuk menghubungkan *NodeMCU* ke *SSID WiFi* yang dituju. Sehingga alat dapat terhubung dengan aplikasi *blynk*.



Gambar 3. Inisialisasi Variabel *SSID* *WiFi*, *Password*, dan *Auth* *Token*

Seperti pada Gambar 3.25, dilakukan penulisan pada aplikasi *Arduino IDE* untuk menginisialisasi variabel dari *SSID WiFi, Password WiFi* dan *Auth Token* yang telah dikirimkan oleh *blynk* ke *email*. Selanjutnya adalah memasukkan sebuah fungsi “Blynk.begin” sehingga mikrokontroler *NodeMCU* akan terhubung ke *SSID WiFi* dan dapat terhubung ke *blynk* dengan menggunakan *Auth Token* yang diberikan oleh pihak *blynk* ke *email*.



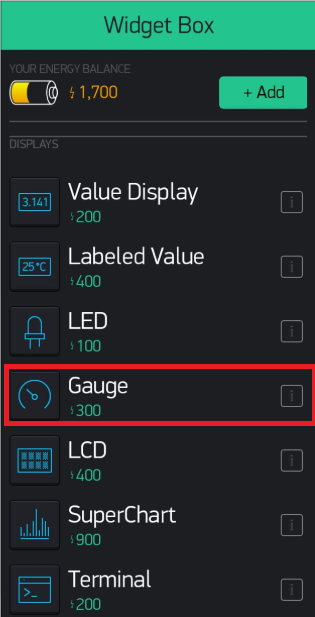
Gambar 3. Penulisan Fungsi “Blynk.begin”

1. Perancangan Sistem Tampilan Aplikasi *Blynk*

Setelah berhasil membuat sistem untuk menghubungkan alat dengan aplikasi *blynk*, maka selanjutnya adalah membuat tampilan dari aplikasi *blynk*. Tampilan pada aplikasi akan menampilkan beberapa informasi yang ditampilkan pada LCD seperti debit, volume air, batas volume air dan tampilan sistem peringatan yang muncul pada aplikasi *blynk*.

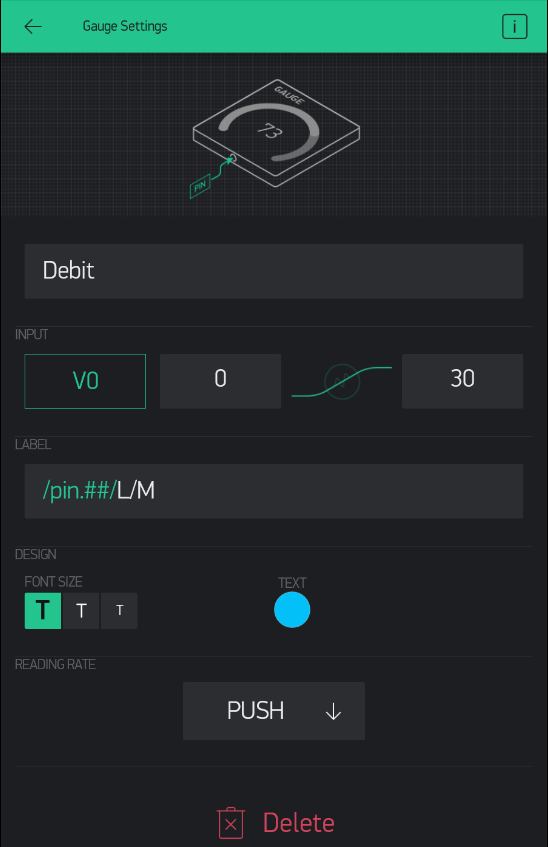
1. **Sistem Tampilan Debit Air**

Dalam perancangan sistem tampilan debit air, digunakan sebuah *widget* berupa *gauge*.



Gambar 3. *Widget Gauge*

Setelah memasukkan *widget gauge* ke tampilan *blynk*, selanjutnya akan dilakukan pengaturan terhadap *widget* tersebut. Untuk pengaturan yang dilakukan, dapat dilihat pada Gambar 3.28.



Gambar 3. Pengaturan *Widget* *Gauge* Untuk Debit Air

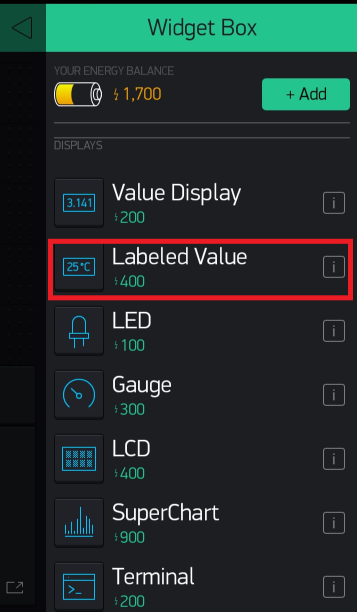
Gambar 3.28 menampilkan sebuah *gauge setting* untuk melakukan pengaturan terhadap *widget gauge*. Pada bagian nama *gauge* diberi nama “Debit” dikarenakan *widget gauge* ini digunakan untuk menampilkan informasi debit air. Selanjutnya pada bagian *input* diatur menggunakan pin V0 dan nilai minimum diatur dari 0 dan nilai maksimum di 30. Nilai maksimum diatur menjadi 30 dikarenakan sensor *water flow* yang digunakan adalah senor *water flow* tipe YF-S201 yang memiliki *range* debit sampai 30 L/menit. Pada bagian *label* diatur dengan penulisan “/pin.##/L/M”. “pin.##” adalah nilai *input* dari V0 yang ditampilkan menggunakan bilangan desimal dua angka dibelakang titik dan “L/M” akan menampilkan sebuah tulisan “L/M” dibelakang nilai *input* V0. Lalu untuk *Reading Rate* diatur menjadi *“Push”*. Kemudian *TEXT* diatur menjadi warna biru.



Gambar 3. Tampilan *Widget* *Gauge* Untuk Informasi Debit

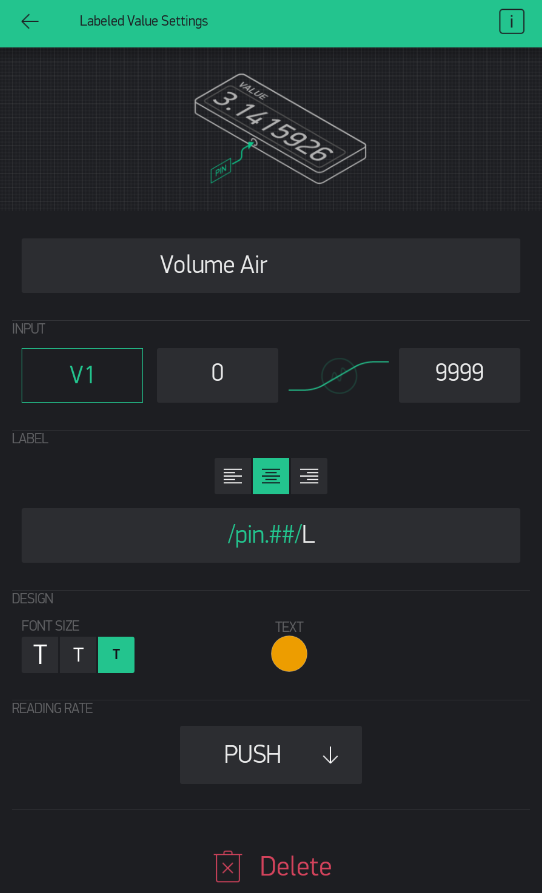
1. **Sistem Tampilan Volume Air**

Dalam perancangan sistem tampilan volume air akan menggunakan sebuah *widget* berupa *labeled value*.



Gambar 3. *Widget* *Labeled* *Value*

Setelah memasukkan *widget labeled value* ke tampilan *blynk*, maka pada *widget labeled value* tersebut perlu dilakukan sebuah pengaturan.



Gambar 3. Pengaturan *Widget* *Labeled* *Value* Untuk Volume Air

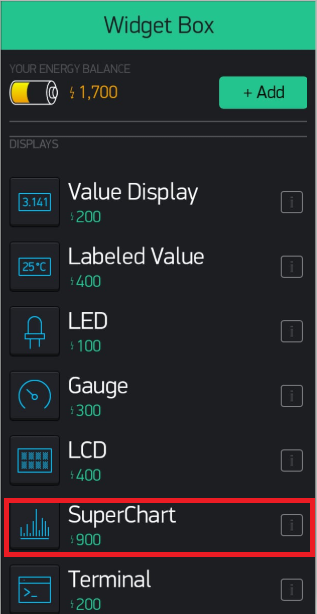
Gambar 3.31 menampilkan sebuah *labeled value settings* untuk melakukan beberapa pengaturan terhadap *widget labeled value*. Pada bagian nama *labeled value* diberi nama “Volume Air”. Selanjutnya untuk bagian *input* diatur menggunakan pin V1 dan nilai minimum diatur menjadi 0 dan nilai maksimum adalah 9999 Liter. Pada bagian *label* diatur dengan penulisan “/pin.##/L”. “pin.##” adalah nilai *input* dari V1 ditampilkan menggunakan bilangan desimal dua angka dibelakang titik dan “L” akan menampilkan tulisan berupa “L” dibelakang nilai *input* V1. Lalu untuk *Reading Rate* diatur menjadi “*Push*”. Untuk warna *text* pada *widget* ini diatur menjadi warna oranye.



Gambar 3. Tampilan *Widget* Menampilkan Volume Air

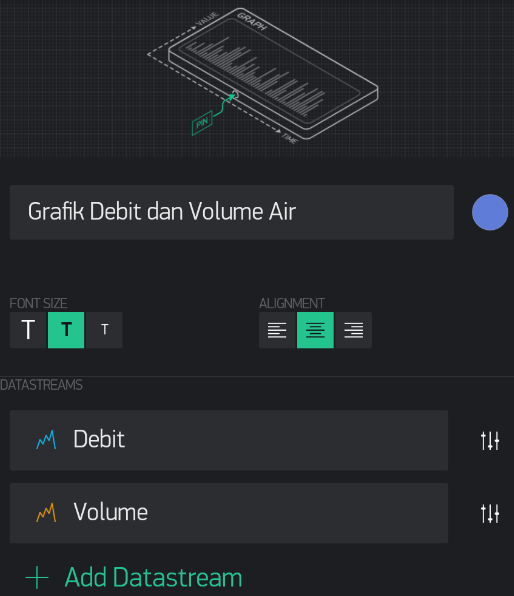
1. **Sistem Tampilan Grafik Debit dan Volume Air**

Tampilan informasi mengenai debit dan volume air juga akan ditampilkan menggunakan sebuah grafik pada aplikasi *blynk* sehingga dapat mengetahui berapa besar nilai debit dan volume air dalam periode tertentu. Untuk hal tersebut dibutuhkan sebuah *widget SuperChart* pada aplikasi *blynk*.



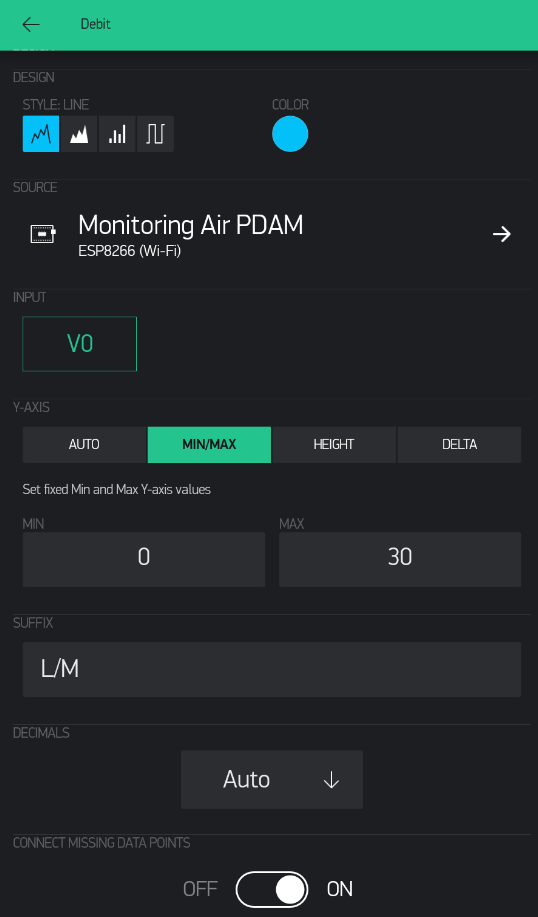
Gambar 3. *Widget* *SuperChart*

Setelah memasukkan *widget SuperChart* ke tampilan *blynk*, selanjutnya adalah melakukan pengaturan terhadap *widget* tersebut. Pertama, *widget* ini akan diberi nama “Grafik Debit dan Volume Air” diikuti dengan penambahan *datastream* yaitu parameter debit dan volume.



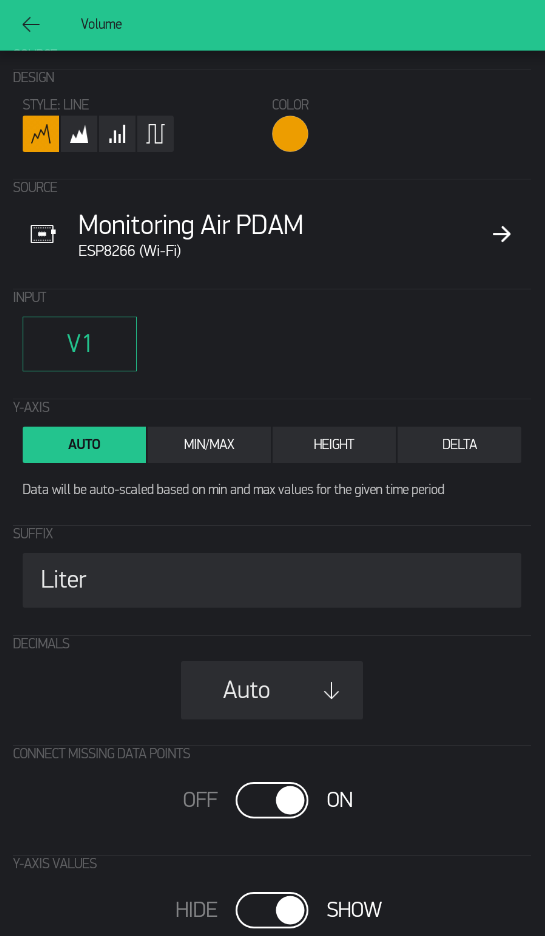
Gambar 3. Pemberian Nama *Widget* dan Penambahan *Datastream*

Setelah memberikan nama dan penambahan *datastrean* debit dan volume, berikutnya akan dilakukan pengaturan terhadap masing – masing *datasream* yang telah ditambahkan. Pada pengaturan *datastream* debit, *style* dari garis akan menggunakan jenis “line” diberi warna biru. Untuk *input* diatur dengan menggunakan *virtual* pinV0 dikarenakan variabel debit ditampung ke *virtual* pin V0. Untuk *Y-Axis* diatur menjadi “Min/Max” dengan nilai minimum yaitu 0 dan maksimum yaitu 30. Untuk *suffix* diberi nama “L/M”. Untuk *decimals* diatur menjadi “Auto”. Untuk *connect missing data points* diatur menjadi “ON” dan *Y-Axis Values* diatur menjadi “SHOW”. Untuk tampilan dari pengaturannya dapat dilihat pada Gambar 3.35.



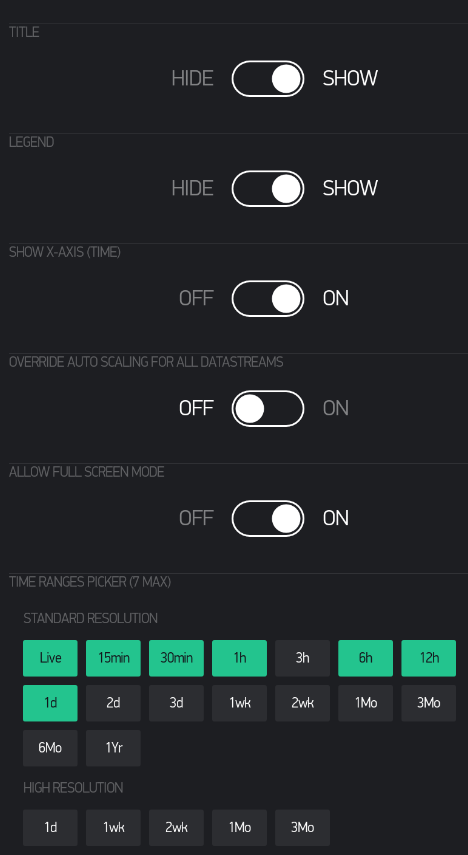
Gambar 3. Pengaturan *Datastream* Debit

Setelah selesai melakukan pengaturan pada *datastream* debit, selanjutnya akan dilakukan pengaturan pada *datastream* volume air. Pada pengaturan *datastream* volume, *style* dari garis akan menggunakan jenis “line” diberi warna oranye. Untuk *input* diatur dengan menggunakan *virtual* pinV1 dikarenakan variabel volume ditampung ke *virtual* pin V1. Untuk *Y-Axis* diatur menjadi “Auto”. Untuk *suffix* diberi nama “L/M”. Untuk *decimals* diatur menjadi “Auto”. Untuk *connect missing data points* diatur menjadi “ON” dan *Y-Axis Values* diatur menjadi “SHOW”. Untuk tampilan dari pengaturannya dapat dilihat pada Gambar 3.36.



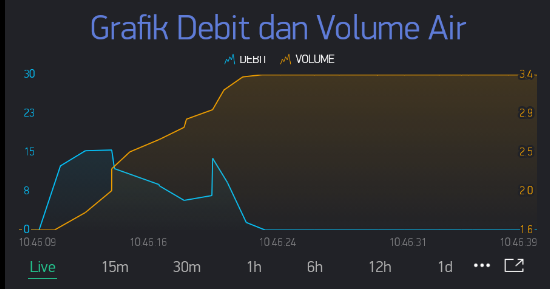
Gambar 3. Pengaturan *Datastream* Volume

Setelah melakukan pengaturan terhadap masing – masing *datastream*, berikutnya pada tampilan awal dari pengaturan *SuperChart* dilakukan beberapa pengaturan. Untuk *title* diatur menjadi “SHOW” sehingga judul dari *widget* ini ditampilkan, *legend* diatur menjadi “SHOW” agar petunjuk warna terhadap *datastream* yang digunakan muncul, *show x-axis* (*time)* diatur menjadi “ON” sehingga informasi waktu ditampilkan pada sumbu x grafik, dan *override auto scaling for all datastreams* diatur menjadi “OFF”. Untuk tampilan dari pengaturannya dapat dilihat pada Gambar 3.37.



Gambar 3. Pengaturan Pada *Widget* *SuperChart*

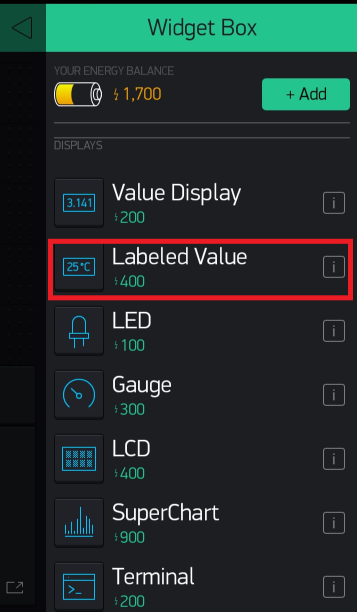
Setelah berhasil melakukan pengaturan terhadap *widget SuperChart* maka *widget* ini siap untuk digunakan. Untuk hasil dari tampilan *widget* ini dapat dilihat pada Gambar 3.38.



Gambar 3. Tampilan Grafik Debit dan Volume Air

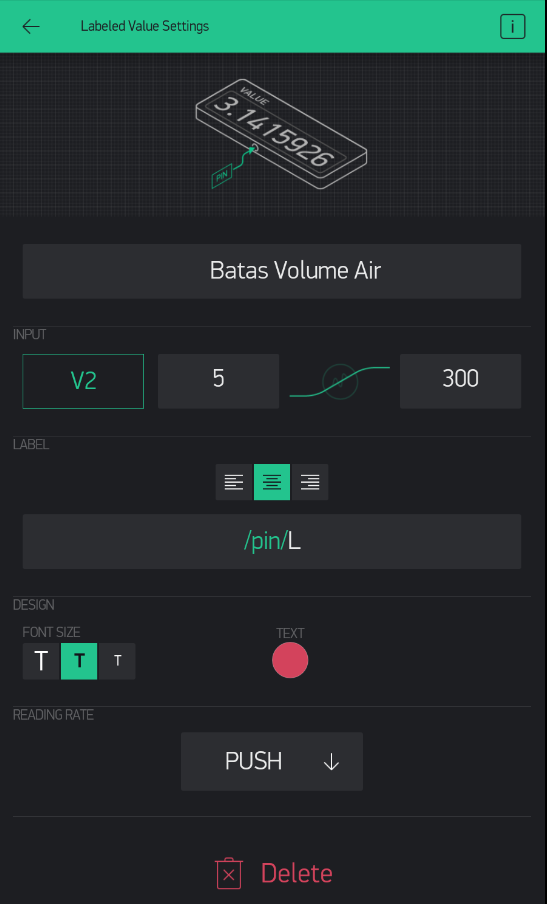
1. **Sistem Tampilan Batas Volume Air**

Dalam perancangan sistem tampilan batas pemakaian air digunakan sebuah *widget labeled value*.



Gambar 3. *Widget* *Labeled* *Value*

Setelah memasukkan *widget* tersebut, maka akan dilakukan pengaturan terhadap *widget* untuk menampilkan nilai batas volume air.



Gambar 3. Pengaturan *Widget* *Labeled* *Value* Untuk Batas Volume Air

Pada bagian nama *labeled value* diberi nama “Batas Volume Air”. Selanjutnya untuk bagian *input* diatur menggunakan pin V2 dikarenakan variabel dari batas volume air ditampung ke *virtual* pin V2. Pada bagian *label* diatur dengan penulisan “/pin/Liter” sehingga nilai yang akan ditampilkan adalah bilangan bulat titik diikuti dengan tulisan “Liter”. Untuk *text* diatur menjadi warna merah.

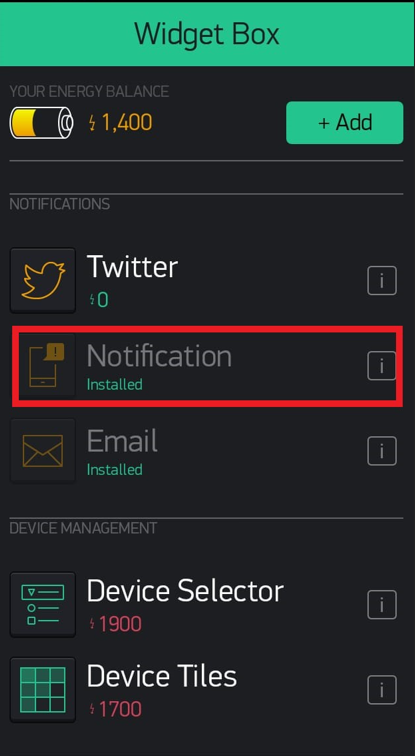


Gambar 3. Tampilan *Widget* Menampilkan Batas Volume Air

Pada tampilan *widget* yang menampilkan nilai batas volume air masih dikosongkan dengan cara tidak memasukkan variabel batas air ke pin *virtual* pada *blynk*. Hal ini dikarenakan penentuan *range* nilai batas volume air akan dilakukan setelah melakukan pengujian rata-rata volume air yang digunakan.

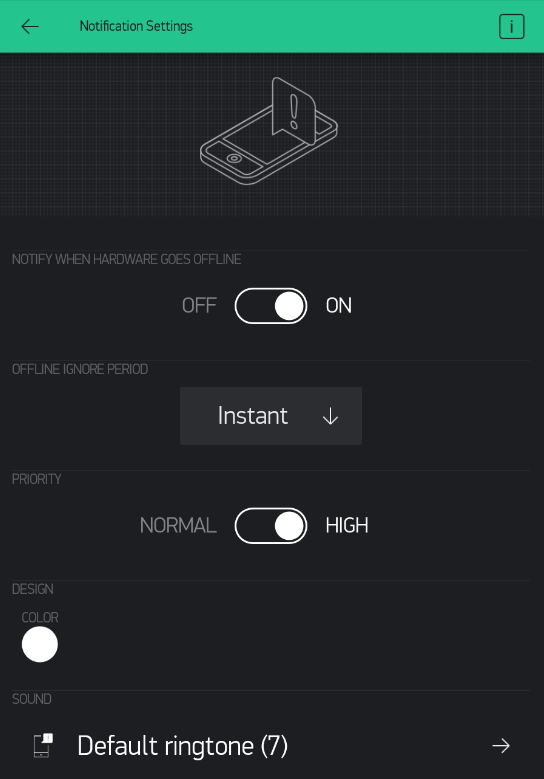
1. **Sistem Tampilan Notifikasi**

Ketika volume air melewati batas volume air yang telah ditentukan, maka sistem peringatan muncul salah satunya dengan cara *blynk* mengirimkan notifikasi ke *smartphone* pengguna. Untuk perancangan sistem tampilan notifikasi ini akan menggunakan *widget notification*.



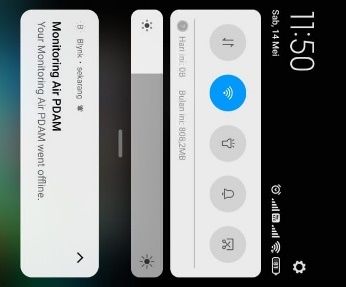
Gambar 3. *Widget Notification*

Setelah memasukkan *widget notification* tersebut, maka dilakukan sebuah pengaturan pada *widget notification* tersebut.



Gambar 3. Pengaturan *Widget* Notification

Seperti yang sudah ditampilkan pada Gambar 3.43, pada bagina *notify when hardware goes offline* diatur menjadi “ON” yang artinya, ketika alat terputus dari jaringan internet atau terputus dari sumber listrik, maka aplikasi *blynk* akan mengirimkan notifikasi. Lalu bagian *offline ignore period* diatur menjadi “Instant” dan *priority* diatur menjadi “HIGH”. Untuk warna dari *widget* ini diatur menjadi putih.



Gambar 3. Tampilan Notifikasi Ketika Alat Mati Atau Terputus Dari Internet

Seperti yang dapat dilihat pada Gambar 3.44, ketika alat terputus dari internet atau terputus dari sumber listrik, maka aplikasi *blynk* mengirimkan sebuah notifikasi dengan judul yang sama pada projek di *blynk* yaitu “Monitoring Air PDAM” dan bertuliskan “Your Monitoring Air PDAM went offline”.

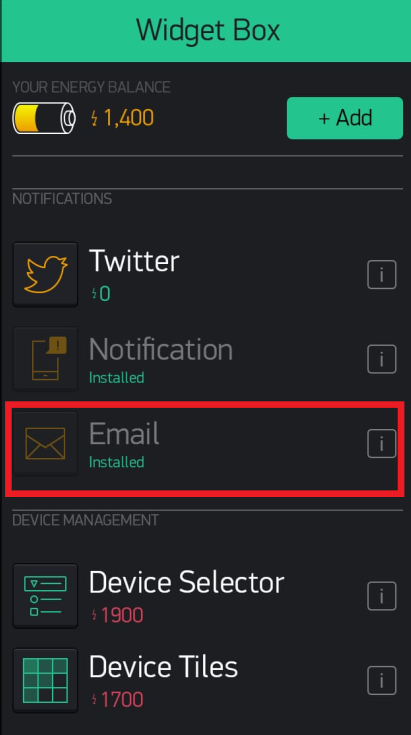


Gambar 3. Tampilan Notifikasi Ketika Sistem Peringatan Aktif

Ketika sistem peringatan jumlah pemakaian air berlebihan aktif, maka *blynk* akan menampilkan notifikasi seperti yang ditampilkan pada Gambar 3.45. Pada tampilan notifikasi tersebut, ditampilkan judul notifikasi yang sama seperti judul projek di *blynk* yang isi nya adalah “Volume Air yang Digunakan Sudah Melewati Batas”.

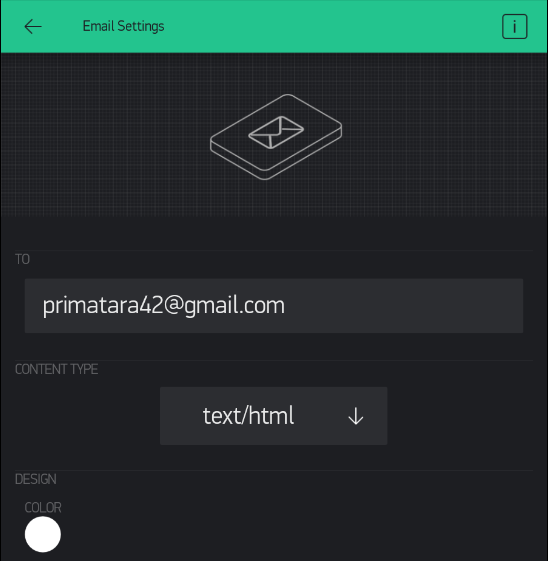
1. **Sistem Tampilan *Email***

Aplikasi *blynk* akan mengirimkan *email* dalam beberapa kondisi. Untuk hal tersebut, maka dibutuhkan sebuah *widget email* pada tampilan aplikasi *blynk*.



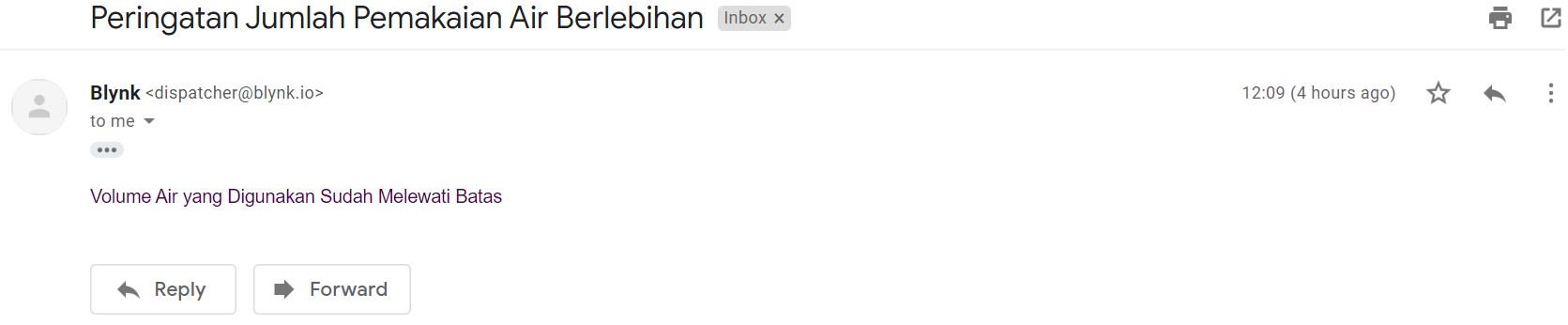
Gambar 3. *Widget Email*

Setelah memasukkan *widget email* ke tampilan *blynk*, berikutnya melakukan pengaturan terhadap *widget* tersebut.



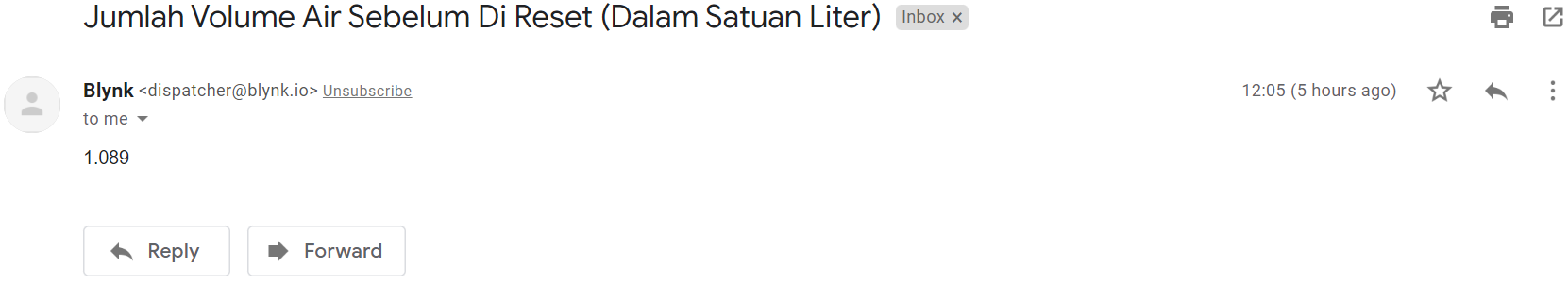
Gambar 3. Pengaturan *Widget* *Email*

Pada bagian pengaturan tersebut untuk bagian TO akan membuat ketika *blynk* mengirimkan *email*, maka akan dikirim ke *alamat email* yang diisi pada bagian TO. Ada beberapa kondisi yang membuat *blynk* mengirimkan *email* ke pengguna. Seperti contoh, ketika sistem peringatan aktif, maka *blynk* akan mengirimkan sebuah *email* ke pengguna. Seperti contoh pada Gambar 3.48.



Gambar 3. *Blynk* Mengirim *Email* Saat Sistem Peringatan Aktif

Pada Gambar 3.48, ketika sistem peringatan aktif yaitu saat volume air melewati batas volume air yang ditentukan, maka *email* yang dikirim oleh *blynk* akan berisikan pesan dengan subjek yaitu “Peringatan Jumlah Pemakaian Air Berlebihan” diikuti dengan isi pesan yaitu “Volume Air yang Digunakan Sudah Melewati Batas”. *Blynk* juga akan mengirimkan *email* ketika nilai volume air di-*reset*. Untuk hasilnya dapat dilihat pada Gambar 3.49.



Gambar 3. *Blynk* Mengirim *Email* Saat Volume Air Di-*Reset*

Seperti yang sudah ditampilkan pada Gambar 3.54, *blynk* mengirimkan sebuah *email* dengan subjek “Jumlah Volume Air Sebelum Di Reset (Dalam Satuan Liter)”. Lalu isi dari *email* tersebut langsung menampilkan nilai volume air dengan menggunakan bilangan desimal 3 angka dibelakang titik.

1. **Hasil Perancangan Tampilan Keseluruhan**

Setelah berhasil merancang seluruh bagian dari sistem tampilan dengan menggunakan berbagai *widget* yang telah difasilitasi oleh pihak pengembang aplikasi *blynk*, maka seluruh sistem tampilan tersebut akan digabung sehingga menjadi tampilan dari keseluruhan sistem yang ditampilkan menggunakan *blynk*. Untuk hasil dari tampilan keseluruhannya dapat dilihat pada Gambar 3.59.



Gambar 3. Tampilan Keseluruhan di *Blynk*