

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Pada penelitian ini, penulis melakukan tinjauan pustaka pada penelitian sebelumnya yang serupa. Sebagai pendukung penelitian yang dilakukan oleh penulis, berikut adalah tinjauan pustaka yang penulis digunakan pada tabel 2.1.

Tabel 2.1 Tinjauan Pustaka

No. Literatur	Penulis	Tahun	Judul
Literatur 1	Wisnu Jatmiko, Prahenusa Wahyu, Ciptadi, R. Hafid Hardyanto	2021	Sistem Penyiram Tanaman Otomatis Berbasis Mikrokontroler dan Panel Surya Dasar di SMK Nufa Citra Mandiri Berbasis Android
Literatur 2	Rahmat Tullah, Sutarman, Agus Hendra Setyawan	2019	Sistem Penyiraman Tanaman Otomatis Berbasis Mikrokontroler Arduino Uno Pada Toko Tanaman Hias Yopi
Literatur 3	Emir Nasrullah, Agus Trisanto, Lioty Utami	2018	Rancang bangun sistem penyiraman tanaman secara otomatis Menggunakan sensor suhu lm35 berbasis mikrokontroler atmega8535
Literatur 4	Deddy Pratama, Amelia Yolanda, Andi Wellyno Pratama	2018	Rancang Bangun Alat Pengontrol Penyiram Tanaman Otomatis Menggunakan Sensor Kelembaban Tanah Di Area Pertanian
Literatur 5	Mindit Eriyadi dan Syafrian Nugroho	2018	Prototipe Sistem Penyiram Tanaman Otomatis Berbasis Suhu Udara Dan Kelembaban Tanah

2.1.1 Literatur 1

(Jatmiko, Ciptadi and Hardyanto, 2021) melakukan penelitian “Sistem Penyiram Tanaman Otomatis Berbasis Mikrokontroler dan Panel Surya”. Sistem ini menggunakan soil moisture sensor untuk menyajikan informasi nilai kelembaban tanah setiap waktu. Selain itu, rancang bangun sistem ini membantu penyiraman lebih terjadwal setiap harinya serta mengetahui cara kerja alat dan pendekatan waterfall untuk perancangan software yang dipakai. Rancang bangun sistem penyiram tanaman otomatis ini menggunakan mikrokontroler NodeMCU dengan dilengkapi soil moisture sensor untuk mengetahui nilai kelembaban tanah setiap waktu dan hasil tersebut akan dikirim secara serial ke software yang dibuat dengan aplikasi Android Studio dan hasil tersebut akan ditampilkan di software dan akan selalu di update dalam jangka waktu tertentu.

2.1.2 Literatur 2

(Tullah, Sutarman and Setyawan, 2019) melakukan penelitian tentang Sistem “Penyiraman Tanaman Otomatis Berbasis Mikrokontroler Arduino Uno Pada Toko Tanaman Hias Yopi”. Proyek ini menggunakan papan Arduino Uno, yang terdiri dari Mikrokontroler ATmega 328, Soil Moisture Sensor, LCD, DHT22, Relay dan Pompa. Arduino Uno berguna untuk menghadapi permasalahan yang terjadi pada kehidupan saat ini. Sistem ini di program sedemikian rupa sehingga akan merasakan tingkat kelembaban tanaman dan menyediakan air jika diperlukan. Jenis sistem ini sering digunakan untuk perawatan tanaman umum, sebagai bagian dari merawat kebun kecil dan sedang. Sistem Penyiraman Otomatis Berbasis Mikrokontroler Arduino Uno mampu meningkatkan kinerja suatu organisasi ataupun instansi dalam pertanian atau perkebunan.

2.1.3 Literatur 3

(Nasrullah et al., 2018) melakukan penelitian “Rancang Bangun Sistem Penyiraman Tanaman Secara Otomatis Menggunakan Sensor Suhu Lm35 Berbasis Mikrokontroler Atmega8535”. Sistem penyiraman tanaman secara otomatis menggunakan mikrokontroler ATmega8535 sebagai pengendali utama dan sensor suhu LM35. Sistem ini juga menggunakan Real Time Clock (RTC) 1307 sebagai pewaktu, serta Liquid Crystal Display (LCD) sebagai penampil. Sistem penyiraman tanaman yang telah dibuat dapat menyiram tanaman secara otomatis. Apabila suhu yang terdeteksi termasuk ke dalam kategori melebihi batas yang telah ditentukan, maka sistem dapat langsung bekerja menyiram tanaman secara otomatis. LCD menampilkan waktu saat ini serta nilai suhu udara di sekitar tanaman yang akan disiram. Alat ini dilengkapi dengan baterai sebagai bahan pengganti sumber tenaganya, alat ini sangat membantu masyarakat untuk rajin mencuci tangan. Pada penelitian ini alat secara keseluruhan bisa bekerja dengan sangat baik dengan jumlah deteksi sensor mengeluarkan air pada kran sebanyak 1 kali dan mengeluarkan sabun sebanyak 1 kali. Desain yang portable ini setelah divalidasi menunjukkan kesesuaian dengan kebutuhan dan keinginan pengguna.

2.1.4 Literatur 4

(Pratama et al., 2018) melakukan penelitian tentang “Rancang Bangun Alat Pengontrol Penyiram Tanaman Otomatis Menggunakan Sensor Kelembaban Tanah Di Area Pertanian”. Cara kerja Alat ini yaitu dengan menggunakan Arduino sebagai otak pergerakan alat dan Sensor Kelembaban Tanah untuk mengambil data dari tanaman. Kemudian data yang telah didapat oleh alat tersebut akan dikirim ke sistem monitoring berbasis website menggunakan Ethernet Shield. Hasil percobaan

dan implementasi di lapangan menunjukkan bahwa prototype atau purwarupa ini dapat bekerja sesuai fungsi dan tujuannya.

2.1.5 Literatur 5

(Eriyadi & Nugroho, 2018) melakukan penelitian tentang “Prototipe Sistem Penyiram Tanaman Otomatis Berbasis Suhu Udara Dan Kelembaban Tanah” Dalam penelitian ini telah dikembangkan sistem penyiram tanaman otomatis berbasis mikrokontroler dengan platform arduino uno dengan sensor soil moisture dan DHT11 serta aktuator berupa solenoid valve. Hasil yang diperoleh dari penelitian yang dikembangkan menunjukkan bahwa prototipe sistem dapat menyiram air pada tanaman apabila kelembaban tanah yang dibaca oleh soil moisture di bawah 300 RH dan suhu udara yang dibaca DHT11 di atas 24 °C. Jika salah satu syarat tidak terpenuhi seperti kelembaban tanah di atas 300 RH atau suhu udara di bawah 24 °C, maka katup dari water solenoid valve tidak akan terbuka.

Perbedaan penelitian penulis dengan penelitian sebelumnya ialah penelitian sebelumnya penyiraman tanaman otomatis yang masih menggunakan sms maupun aplikasi BLYNK yang belum mampu menyiram tanaman secara otomatis melalui aplikasi jarak jauh. Sedangkan penelitian penulis menggunakan aplikasi telegram dan mampu menyiram tanaman secara otomatis melalui aplikasi tersebut tanpa harus menyiramnya secara manual.

2.2 NodeMCU

NodeMCU adalah sebuah *platform* IOT yang bersifat *open source*, yang biasanya dianalogikan sebagai board arduino ESP8266. Dalam seri tutorial ESP8266 pernah membahas bagaimana memprogram ESP8266 dalam pemograman ini sedikit merepotkan karena, dibutuhkan beberapa teknik

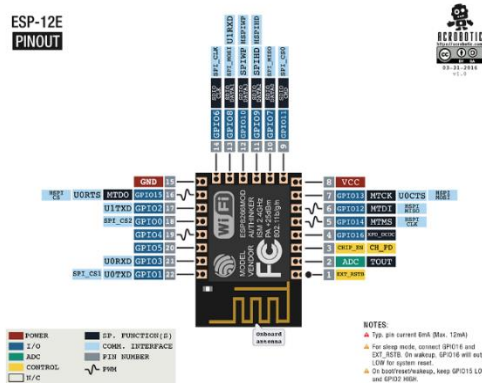
wiring serta tambahan modul USB to serial untuk mendownload program. Tetapi *NodeMCU* telah me-package ESP8266 ke dalam sebuah *board* yang sama dan dilengkapi berbagai fitur layaknya mikrokontroler dan kapabilitas akses terhadap *Wifi* juga chip komunikasi USB to serial. Sehingga untuk memprogramnya hanya diperlukan ekstensi kabel data USB persis yang digunakan sebagai kabel data dan kabel *charging* smartphone Android (Setyawan et al., 2018).

2.3 Sejarah NodeMCU

Sejarah dari *NodeMCU* berdekatan dengan rilis ESP8266 yaitu pada tanggal 30 Desember 2013, **Espressif Systems** selaku pembuat ESP8266 memulai produksi ESP8266 yang merupakan *SoC Wi-Fi* yang terintegrasi dengan prosesor Tensilica Xtensa LX106. Sedangkan *NodeMCU* dimulai pada 13 Oktober 2014 saat Hong me-commit file pertama *nodemcu-firmware* ke Github. Dua bulan kemudian project tersebut dikembangkan ke platform perangkat keras ketika Huang R meng-commit file dari *board* ESP8266, yang diberi nama *devkit v.0.9*. Berikutnya, di bulan yang sama. Tuan PM memporting pustaka *client* MQTT dari Contiki ke platform SOC ESP8266 dan di-commit ke project *NodeMCU* yang membuatnya mendukung protokol IOT MQTT (Setyawan et al., 2018).

Pemutakhiran penting berikutnya terjadi pada 30 Januari 2015 ketika *devsaurus* memporting *u8glib* ke project *NodeMCU* yang memungkinkan *NodeMCU* bisa *mendrive display* LCD, LED, hingga VGA. Demikianlah, project *NodeMCU* terus berkembang hingga kini berkat komunitas *open source* dibalikinya, pada musim panas 2016 *NodeMCU* sudah terdiri memiliki 40 modul fungsionalitas yang bisa digunakan sesuai kebutuhan *developer*. Karena jantung dari *NodeMCU* adalah ESP8266 (khususnya seri ESP-12, termasuk ESP-12E) maka fitur – fitur

yang dimiliki *NodeMCU* akan kurang lebih sama ESP-12 dibawah ini merupakan contoh *NodeMCU* yang akan digunakan dalam penelitian ini. Terdapat pada Gambar 2.3.



Gambar 2.3 Tampilan *NodeMCU*
Sumber : (Setyawan et al., 2018)

2.4 Kabel Jumper

Salah satu komponen yang cukup penting dalam membuat rangkaian ini adalah kabel *jumper* Arduino. Kabel jumper memiliki arti yaitu kabel elektrik yang memiliki pin konektor di setiap ujungnya dan memungkinkan untuk menghubungkan dua komponen yang melibatkan *Arduino* tanpa memerlukan *solder*. Konektor yang ada pada ujung kabel terdiri atas dua jenis yaitu konektor jantan (*male connector*) dan konektor betina (*female connector*). Dibawah ini contoh kabel *jumper* yang digunakan (Nusyirwan, 2019). Terdapat pada Gambar 2.4.



Gambar 2.4 Kabel Jumper
Sumber : (Nusyirwan, 2019)

2.5 Sensor Soil Moisture

Moisture sensor adalah sensor yang dapat mendeteksi kadar air dalam tanah. Sensor ini sangat sederhana, tetapi ideal untuk memantau tanaman, atau tingkat air pada tanaman/tumbuhan pekarangan anda. Sensor ini terdiri dua probe untuk melewatkan arus melalui tanah, kemudian membaca resistansinya untuk mendapatkan nilai tingkat kadar air. Semakin banyak air membuat tanah lebih mudah menghantarkan listrik (resistansi kecil), sedangkan tanah yang kering sangat sulit menghantarkan listrik (resistansi besar). Sensor ini sangat membantu anda untuk mengingatkan tingkat kadar air pada tanaman anda atau memantau kadar air tanah di kebun (Galih Mardika, Kartadie, 2019). Terdapat pada Gambar 2.5.



Gambar 2.5 Sensor *Soil Moisture*
Sumber : (Galih Mardika, 2019)

2.6 Hardware

Dalam *hardware arduino* mempunyai beberapa jenis, ada kelebihan dan kekurangan dalam setiap papannya. *Arduino uno* yang digunakan harus sesuai dengan perancangan yang dibutuhkan. Penambahan fungsi dalam setiap *board* itulah yang membedakan setiap *arduino*. Dalam penelitian ini *NodeMCU ESP8266* yang akan digunakan (Putri et al., 2016).

2.7 Software

Driver IDE adalah *driver* dari *software* yang masih memiliki beberapa *software* lain yang sangat bermanfaat. *Integrated Development Enviroment (IDE)*,

suatu program khusus untuk suatu komputer agar dapat membuat suatu rancangan atau *sketsa* program untuk papan *arduino* (Putri, Marwan and Hariyono, 2016). IDE *arduino* terdiri dari :

1. **Editor Program**

Sebuah window yang digunakan untuk menulis dan mengedit program dalam bahasa *processing*.

2. **Compiler**

Berfungsi untuk kompilasi *sketch* tanpa unggah ke *board* bisa dipakai untuk pengecekan kesalahan *sintaks sketch*. Sebuah modul yang mengubah kode program menjadi kode biner bagaimanapun sebuah *mikrokontroler* tidak akan bisa memahami bahasa *processing*.

3. **Uploader**

Untuk mengunggah hasil kompilasi *sketch* ke *board* target. Pesan eror akan terlihat pada layar log.

4. **New Sketch**

Membuka window dan membuat *sketch* baru.

5. **Open Sketch**

Membuka *sketch* yang sudah pernah dibuat. *Sketch* yang dibuat dengan IDE *Arduino* akan disimpan pada folder yang kita inginkan .

6. **Save Sketch**

Menyimpan *sketch*, tapi tidak disertai dengan mengkompile.

7. **Serial Monitor**

Membuka interface untuk komunikasi serial, nanti akan kita diskusikan lebih lanjut pada bagian selanjutnya.

8. Keterangan Aplikasi

Pesan-pesan yang dilakukan aplikasi akan muncul di sini, misal **Compiling** dan **Done Uploading** ketika kita mengkompile dan mengupload *sketch* ke *board Arduino*.

9. Console log

Pesan-pesan yang dikerjakan aplikasi dan pesan-pesan tentang *sketch* akan muncul pada bagian ini. Misal ketika aplikasi mengkompile atau ketika ada kesalahan pada *sketch* yang kita buat, maka informasi error dan baris akan diinformasikan di bagian ini.

10. Baris Sketch

Bagian ini akan menunjukkan posisi baris kursor yang sedang aktif pada *sketch*.

11. Informasi Board dan Port

Bagian ini menginformasikan port yang dipakai oleh *board arduino*.



Gambar 2.7 *Arduino IDE*
Sumber : (Galih Mardika, 2019)

2.8 Sketch Arduino

(Nusyirwan, 2019) Pada arduino bahasa pemrograman yang digunakan adalah bahasa C++. Program pada Arduino terbagi menjadi tiga bagian utama yaitu:

1. Structure.

Struktur kode pada arduino yaitu berisi fungsi *setup()* dan *loop()*.

A. Setup()

Fungsi ini dipanggil pertama kali ketika menjalankan *sketch*. digunakan sebagai tempat inisialisasi *variabel*, *pin mode*, penggunaan *library* dan lainnya. fungsi ini dijalankan sekali ketika *board* dinyalakan atau di reset.

B. Loop()

Setelah membuat fungsi *setup()* sebagai tempat inisialisasi variabel dan menetapkan nilai maka selanjutnya fungsi *loop()* seperti namanya fungsi ini akan melakukan perulangan berturut-turut, memungkinkan program untuk mengubah dan menanggapi. digunakan untuk mengontrol *board Arduino*.

2.9 Internet Of Things (IOT)

Internet Of Things (IOT) merupakan perangkat elektronik yang mampu berinteraksi dengan pengguna untuk tujuan memantau atau mengendalikan pada perangkat tersebut melalui jaringan internet. Hal ini dapat diwujudkan dengan layanan kompleks melalui koneksi antara objek fisik dan virtual berdasarkan teknologi informasi terkini dan perkembangan teknologi komunikasi (Surahman et al., 2021). Dengan Internet Of Things (IOT) dapat membuat lingkungan internet yang dilengkapi dengan fasilitas untuk memudahkan masyarakat dalam mengakses

teknologi cerdas yang terintegrasi dengan otomatisasi yang dapat digunakan kapan saja dan dimana saja (Megawati, 2021).

Berdasarkan hal tersebut, penulis menggunakan IOT pada penelitian ini untuk mempermudah dan mempercepat pengguna dalam pengoperasian alat penyiraman tanaman bunga keladi barret merah berbasis internet of things berdasarkan kelembaban tanah menggunakan telegram.

2.10 Telegram

Telegram adalah aplikasi komunikasi yang fokus pada performa dan tingkat keamanan lebih dengan tampilan sederhana dan gratis dalam penggunaannya (Prabowo, Kusnadi and Subagio, 2020). Pada situs resminya di telegram.org, telegram mengklaim bahwa aplikasinya memungkinkan mengakses obrolan dari perangkat berada dalam satu akun (*sync*), selain itu telegram juga memiliki bot yang dapat digunakan dalam pengembangan IOT pada penelitian ini.

Oleh sebab itu penulis memilih aplikasi telegram sebagai perangkat lunak yang akan digunakan untuk mengirim perintah dan menerima data karena dinilai kemudahan dan keamanan dalam penggunaannya. Adapun logo dari aplikasi telegram terdapat pada Gambar 2.10.



Gambar 2.10 Logo Telegram
Sumber : (Prabowo, Kusnadi and Subagio, 2020)




2.11 Fritzing


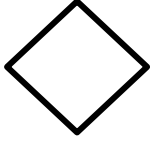



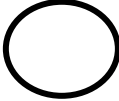
Fritzing merupakan salah satu software yang cukup bagus untuk belajar elektronika. *Software Fritzing* ini merupakan perangkat lunak yang dapat digunakan oleh para penghobi elektronika. *Software Fritzing* dapat dioperasikan pada sistem Windows maupun Linux. Pada penelitian ini *fritzing* digunakan untuk mendesain skematik alat (Ahmad et al., 2015).

2.12 Flowchart

Setelah penulis membuat blok diagram maka tahap selanjutnya adalah membuat flowchart. Flowchart tersebut memiliki fungsi sebagai penentu atau acuan untuk penulis melakukan urutan *step by step* dari proses yang akan dikerjakan oleh aplikasi dan mikrokontroler yang akan dibuat nantinya. Flowchart sangat berpengaruh terhadap layak atau tidak layak sistem tersebut dijalankan. Tahapan ini merupakan pondasi awal untuk sebelum terbentuknya suatu sistem atau alat. Jika pada pengerjaan atau pembuatan flowchart sudah tidak baik, maka bisa dipastikan bahwasannya sistem atau alat yang akan dibuat tidak baik atau sempurna. Maka sangatlah penting bagi kita untuk mengikuti prosedur dasar tersebut, agar sistem atau alat yang dihasilkan jauh lebih baik (Ilham Budiman, Sopyan Saori, Ramdan Nurul Anwar, Pangestu, 2021).

Tabel 2.2 Simbol *Flowchart*

NAMA	SIMBOL	KETERANGAN
Terminal		Simbol yang berfungsi untuk menunjukkan proses awal atau akhir suatu proses
Proses		Simbol yang berfungsi untuk menunjukkan proses suatu sistem
Proses		Simbol proses yang dilakukan secara manual

Proses		Simbol yang digunakan oleh manusia dan komputer seperti memasukan data ke komputer
Decision		Simbol pengambilan keputusan bagaimana alur dalam flowchart berjalan selanjutnya berdasarkan pernyataan
Stored data		Simbol informasi yang disimpan ke dalam media penyimpanan umum.
Data Base		Untuk basis data atau databases
Predefined Process		Untuk proses yang telah kita jelaskan lebih rinci di dalam flowchart tersendiri
Koneksi		Pengganti garis penghubung

2.13 LCD 16 x 2

Liquid Crystal Display atau sering disebut dengan nama LCD adalah alat yang sering digunakan untuk menampilkan informasi dari sistem yang berupa tulisan atau teks (Samsugi et al., 2020) LCD bisa dikatakan alat yang penting bagi pengguna dikarenakan LCD berfungsi untuk menampilkan informasi tampilan interface mengenai yang sedang berjalan. Berdasarkan panjang data antarmuka LCD yang dibedakan menjadi dua jenis yaitu : 4 bit dan 8 bit. Namun, dalam perancangan LCD akan memerlukan banyak pin dari mikrokontroler. Itu karena, mempunyai fungsinya masing-masing yang sangat dibutuhkan untuk mensupport kinerja dari LCD. Untuk menjalankan LCD mikrokontroler biasanya akan sangat membutuhkan perangkat dari sebuah variabel. Pada penelitian ini penulis menggunakan LCD 16 x 2 yang berarti memiliki 16 kolom dan 2 baris karakter. Terdapat pada Gambar 2.13.



Gambar 2.13 *Liquid Crystal Display (LCD)*
Sumber : (Syahrin, 2017)

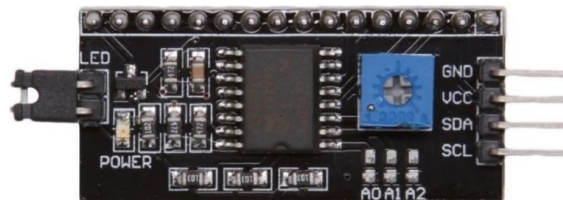
Tabel 2.3 Keterangan LCD 16x2

1	VSS	Power	Catu daya, ground
2	VDD	Power	Catu daya Positif
3	V0	Power	Pengaturan kontras, perlu dihubungkan dengan pin VSS melalui resistor 2,2 – 0 Ohm
4	RS	Input	Register Select High : untuk mengirim data Low : untuk mengirim instruksi
5	R/W	Input	Read/Write control bus High = membaca data lcd Low = menulis ke lcd
6	E	Input	Data enable untuk mengatur lcd. Ketika bernilai LOW, lcd tidak dapat diakses.
7-14	DB0	I/O	Data
15	BLA	Power	Catu daya layar, positif
P16	BLK	Power	Catu daya layar, negatif

2.14 I2C

I2C (*Inter Integrated Circuit*) merupakan standar serial komunikasi dua arah yang didesain dengan dua saluran untuk mengirim atau menerima data. I2C tersusun atas saluran SCL (*Serial Clock*) dan SDA (*Serial Data*) dimana saluran ini membawa informasi data antara pengontrol dengan I2C. Piranti yang terhubung dengan I2C Bus dioperasikan sebagai *Master* dan *Slave*. *Master* akan memulai transfer data pada I2C Bus dengan cara membentuk sinyal *Start*, mengakhirinya

dengan sinyal *Stop* dan memicu sinyal *Clock*. *Master* akan memberi alamat pada piranti *Slave* (Firman, 2016). Berdasarkan hal tersebut, I2C dapat digunakan untuk menampilkan data dari mikrokontroler ke LCD. Tampilan fisik dari I2C terdapat pada Gambar 2.14.



Gambar 2.14 I2C
Sumber : (Firman, 2016)

2.15 Modul Relay

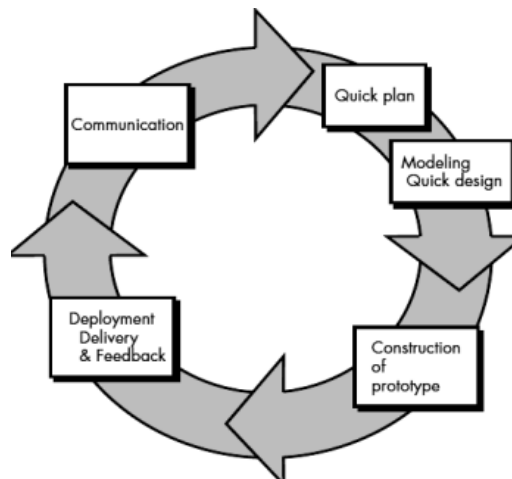
Relay adalah salah satu piranti yang beroperasi berdasarkan prinsip elektromagnetik untuk menggerakkan kontaktor guna memindahkan posisi hidup ke mati atau sebaliknya dengan memanfaatkan tenaga listrik. Peristiwa tertutup dan terbukanya kontaktor ini terjadi akibat adanya efek induksi magnet yang timbul dari kumparan induksi listrik (Turang, 2015). Dibawah ini adalah contoh relay yang akan penulis gunakan dalam penelitian ini. Modul Relay 5V 1 Channel. Tampilan fisik dari relay terdapat pada Gambar 2.15.



Gambar 2.15 Modul Relay
Sumber : (Turang, 2015).

2.16 Metode *Prototype*

Metode *prototype* adalah metode yang dapat digunakan untuk mengembangkan sebuah perangkat yang akan dikembangkan kembali. Metode ini dimulai dengan pengumpulan kebutuhan pengguna dalam hal ini pengguna dari perangkat yang dikembangkan adalah tanaman bunga keladi barret merah. Kemudian membuat sebuah rancangan kilat yang selanjutnya akan dievaluasi kembali sebelumnya diproduksi secara benar (Purnomo, 2017). Berikut ini adalah gambar *prototype* yang digunakan oleh penulis. Tampilan fisik dari metode *prototype* terdapat pada Gambar 2.16.



Gambar 2.16 Metode *Prototype*
Sumber : (Pressman, 2010)

Keterangan Metode *Prototype* :

1. *Communication* / Komunikasi pengembangan perangkat lunak melakukan pertemuan dengan pengguna untuk menentukan kebutuhan perangkat lunak yang saat itu diketahui dan untuk menggambarkan area-area dimana definisi lebih lanjut untuk iterasi selanjutnya.
2. *Quin Plan* / Perencanaan secara cepat dalam pembuatan *prototype*. Setelah itu dilakukan pemodelan dalam bentuk “rancangan cepat”.

3. *Modeling Quick Design* / Model rancangan cepat pada tahap ini memodelkan perancangan tadi menggunakan tools yed graph editor yaitu *flowchart* untuk mendefinisikan fungsi dari sistem dan alat.
4. *Construction Of Prototype* / Pembuatan prototype dalam pembuatan rancangan cepat berdasarkan pada representasi aspek-aspek perangkat lunak yang akan terlihat oleh para pengguna.
5. *Deployment Delivery & Feedback* / Penyerahan dan memberikan umpan balik terhadap pengembangan prototype kemudian diserahkan kepada pengguna untuk evaluasi *prototype* yang telah dibuat sebelumnya dan memberikan umpan balik yang akan digunakan untuk memperbaiki spesifikasi kebutuhan. Iterasi terjadi saat pengembangan melakukan perbaikan terhadap *prototype* tersebut.