

## BAB II

### LANDASAN TEORI

#### 2.1 Hidroponik

Hidroponik adalah metode menanam yang tidak menggunakan tanah melainkan air yang berisi nutrisi. Tanaman hidroponik sangat bergantung pada larutan nutrisi dalam pertumbuhannya, dimana larutan tersebut akan diberikan langsung ke akar tanaman [6]. Dalam hal ini, sangatlah penting untuk memastikan bahwa tanaman mendapat nutrisi esensial dengan kondisi tumbuh yang optimal. Nutrisi pada sistem hidroponik telah diformulasikan secara efektif sesuai dengan apa yang dibutuhkan oleh tanaman, berisi unsur hara makro dan mikro.

Beberapa manfaat yang didapat dari hidroponik adalah seperti kemampuan untuk menggunakan kembali air dan nutrisi, kontrol pada lingkungan tanaman yang lebih mudah, dan mengurangi risiko penyakit pada tanaman oleh hama yang biasanya ditularkan melalui tanah [7]. Dengan adanya hidroponik, tanaman dapat tumbuh dalam lingkungan yang berbeda. Hidroponik tidak lagi memperlumahkan tempat dimana kondisi tanah tidak subur, lahan pertanian yang mahal, ataupun air yang sulit didapatkan [8]. Selain itu hidroponik tidak memerlukan pemakaian herbisida dan pestisida sehingga lebih ramah lingkungan dan sayuran yang dihasilkan akan lebih sehat.

Hidroponik memiliki beberapa jenis teknik yaitu *Wick* (sistem sumbu), *Drip*, *Ebb-Flow*, *Water Culture*, *Nutrient Film Technique (NFT)*, *Aeroponic* dan *Windowfarm* [9]. Dalam penelitian ini, sistem hidroponik yang dibuat menggunakan teknik NFT. Dengan teknik NFT, larutan nutrisi akan dialirkan terus menerus selama 24 jam pada *grow tray* (talang). *Grow tray* adalah tempat dimana diletakkannya tanaman dan tempat tanaman mendapat larutan nutrisi. Setelah dialirkan menuju *grow tray*, larutan nutrisi akan kembali ke *reservoir* (tangki tempat penampungan larutan nutrisi) dan disirkulasikan kembali menggunakan *water pump* (pompa).

Dalam NFT, kedalaman air selama disirkulasikan idealnya bersifat dangkal, hal ini memastikan agar perakaran selalu mendapat air, nutrisi dan oksigen yang cukup. Tingkat kemiringan dari *grow tray* dan daya *water pump* akan mempengaruhi tingkat kedalaman air, sehingga teknik NFT harus dirancang dengan kemiringan dan panjang *grow tray* serta laju aliran dari *water pump* yang tepat. Teknik NFT ini sangat baik untuk

memastikan akar tanaman mendapat oksigen yang cukup dan dialiri nutrisi dengan jumlah yang sesuai.

## 2.2 Nutrisi Hidroponik

Nutrisi hidroponik adalah pupuk hidroponik yang mengandung semua unsur hara makro dan mikro secara lengkap dengan komposisi yang diperlukan oleh tanaman. Nutrisi merupakan aspek yang sangat penting dalam pertumbuhan tanaman hidroponik karena menjadi sumber utama bagi tanaman untuk menumbuhkan sel-sel nya. Dalam nutrisi harus mengandung unsur hara makro seperti Nitrogen, Phospor, Kalium, Kalsium, Magnesium, dan Sulfur; serta unsur mikro seperti Klor, Zat Besi, Mangan, Tembaga, Zink, Boron, dan Molibdenum. Manfaat dari masing-masing unsur hara dapat dilihat pada Tabel 2.1 [10] [6].

Tabel 2.1 Manfaat dari Unsur Hara terhadap Tanaman

Unsur Hara	Manfaat terhadap tanaman
Nitrogen (N)	Merangsang pertumbuhan vegetatif; berperan dalam pembentukan klorofil dan protein.
Phospor (P)	Merangsang pertumbuhan akar; dibutuhkan dalam pembungaan dan pemasakan biji serta buah.
Kalium (K)	Membantu pembentukan protein dan karbohidrat; memperkuat tanaman; sumber daya tahan terhadap kekeringan dan penyakit.
Kalsium (Ca)	Merangsang pembentukan bulu-bulu akar dan pembentukan biji; memperkuat batang; menetralkan senyawa asam.
Magnesium (Mg)	Berperan membentuk klorofil, karbohidrat, lemak dan senyawa minyak; transportasi Phospat.
Sulfur (S)	Berperan dalam pembentukan bintil akar; membantu pertumbuhan anakan tanaman.
Klor (Cl)	Memperbaiki dan meningkatkan hasil kering tanaman.
Zat Besi (Fe)	Membantu proses pernapasan tanaman dan pembentukan klorofil.
Mangan (Mn)	Melancarkan proses asimilasi; membantu pembentukan dan melancarkan kerja enzim.
Tembaga (Cu)	Berperan dalam pembentukan klorofil; bahan pembentuk beberapa jenis enzim.
Zink (Fe)	Berfungsi mengaktifkan beberapa enzim; berperan dalam biosintesis auksin, pemanjangan sel dan ruas batang.
Boron (B)	Mengangkut kabohidrat ke tubuh tanaman; berperan dalam pembelahan sel pada tanaman biji.
Molibdenum (Mo)	Membantu mengikat nitrogen di udara bebas; mengaktifkan enzim Nitrogenase.

Tabel 2.1 Manfaat dari Unsur Hara terhadap Tanaman (lanjutan)

Petani hidroponik tidak perlu menformulasikan unsur-unsur tersebut secara manual karena nutrisi hidroponik sudah banyak dijual dan siap dipakai, yang disebut nutrisi hidroponik A & B MIX. Nutrisi ini tersedia untuk sayuran, buah dan juga bunga. Satu set dari nutrisi hidroponik tersebut terdiri dari 2 kantong yaitu kantong A dan kantong B, yang tersedia untuk sayuran, buah dan juga bunga. Isi dari masing-masing kantong berbentuk bubuk dan mudah larut dalam air.

Setiap tanaman memiliki tingkat kebutuhan nutrisi yang berbeda-beda. Kepekatan nutrisi yang dibutuhkan oleh masing-masing tanaman dapat diukur menggunakan TDS (*Total Dissolved Solid*) meter. Alat ini mengukur jumlah padatan terlarut pada suatu cairan. Satuan dari TDS meter adalah ppm (*Part per Million*) yang menyatakan rasio dari jumlah kontaminan atau konsentrasi yang terdapat dalam suatu zat [29]. Tabel 2.2 menyatakan besar ppm yang dibutuhkan oleh berbagai tanaman [11].

Tabel 2.2 Nilai PPM Kebutuhan Tanaman

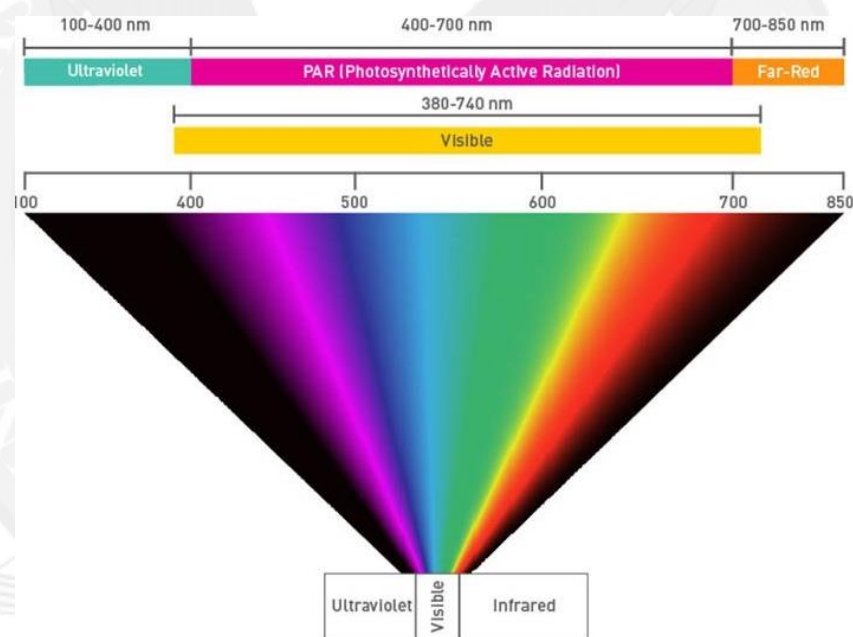
Nama Tanaman	Nilai ppm
Bayam	1260 – 1610
Kangkung	1050 – 1400
Kubis	1750 – 2100
Selada	560 – 840
Sawi Manis	1050 – 1400
Seledri	1260 – 1680
Tomat	1400 – 3500
Timun	1190 – 1750
Cabai	1260 – 1540
Strawberry	1260 – 1540
Mint	1400 – 1680
Parsley	560 – 1260

Rentang nilai ppm pada Tabel 2.2 menunjukkan batas bawah dan batas atas ppm. Perbedaan nilai ppm ini terjadi karena nutrisi yang diberikan kepada tanaman akan semakin pekat seiring dengan bertambahnya umur tanaman, sehingga nilai ppm pun semakin besar. Para petani hidroponik pada umumnya akan melakukan pengecekan nilai ppm dari larutan nutrisi pada *reservoir* (tangki) setiap hari. Apabila larutan nutrisi kurang pekat, maka mereka akan menambahkan kembali nutrisi A dan B kedalam tangki hingga mencapai nilai ppm yang dibutuhkan.

## 2.3 Cahaya pada Tanaman

Tanaman membutuhkan cahaya agar dapat hidup dan berkembang. Cahaya itu sendiri merupakan energi berbentuk gelombang elektromagnetik yang memiliki panjang gelombang berbeda-beda. Panjang gelombang cahaya yang dapat terlihat oleh manusia yaitu berkisar antara 380-740 nanometer (nm) [12] [13]. Selain itu masih ada sinar ultraviolet memiliki panjang gelombang 100-400 nm, sinar far-red 700-850 nm, infra-red 700-10<sup>6</sup> nm [12].

Pada tanaman, gelombang cahaya yang digunakan agar dapat melakukan fotosintesis dengan baik disebut sebagai PAR (*Photosynthetically Active Radiation*). Panjang gelombang PAR berkisar antara 400-700 nm, dan melingkupi warna-warna seperti merah, hijau dan biru. Gambar 2.1 menggambarkan panjang gelombang dari berbagai spektrum cahaya [12].

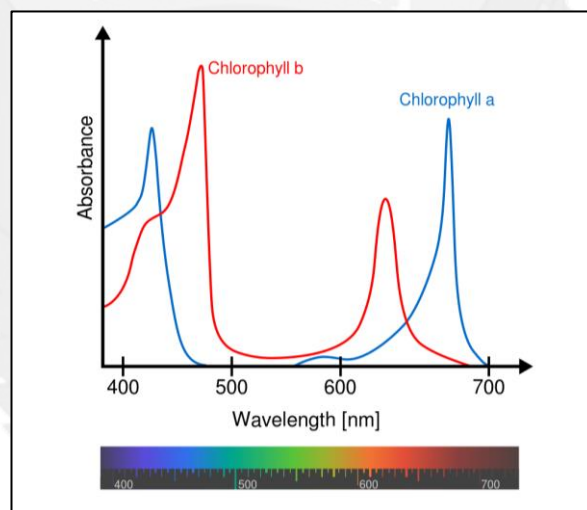


Gambar 2.1 Panjang Gelombang Spektrum Cahaya

Setiap spektrum cahaya yang ada dalam lingkup PAR memiliki perbedaan pengaruh terhadap pertumbuhan tanaman. Tanaman memiliki fotoreseptor yang dapat memicu karakteristik pertumbuhan yang berbeda ketika diaktifkan oleh foton dengan panjang gelombang tertentu. Dengan mengendalikan spektrum cahaya apa yang diberikan pada tanaman dan seberapa besar intensitasnya akan menghasilkan pertumbuhan tanaman yang baik. Karakteristik pertumbuhan yang dapat dipengaruhi oleh spektrum cahaya

adalah seperti pembuahan, pembungaan, berat, perkembangan akar, warna, rasa, dan nutrisi [12].

Diantara warna-warna yang termasuk dalam cahaya PAR, warna biru dan merah adalah warna yang paling efektif terhadap fotosintesis tanaman [12][14]. Pada daun terdapat klorofil a dan b yang merupakan pigmen fotosintesis utama. Klorofil ini paling kuat menyerap cahaya biru dan merah. Gambar 2.2 mengilustrasikan grafik penyerapan cahaya pada klorofil a dan b. Cahaya hijau sedikit diserap oleh daun sehingga membuat daun tampak hijau karena memantulkan sekitar 10-50% foton gelombang hijau yang diterima [12][14].



Gambar 2.2 Grafik Perbandingan Tingkat Penyerapan Setiap Gelombang Cahaya pada Daun

Memberikan cahaya biru dan merah dengan porsi atau rasio yang sesuai juga akan sangat membantu proses fotosintesis dan penyerapan mineral. Cahaya hijau juga tetap memiliki manfaat yang baik bagi tanaman walaupun sedikit terserap oleh daun [15]. Pengaruh dari masing-masing spektrum cahaya dalam lingkup PAR terhadap perkembangan tanaman dijelaskan pada Tabel 2.3 [12] [14].

Tabel 2.3 Pengaruh Spektrum Cahaya PAR terhadap Perkembangan Tanaman

Gelombang Cahaya	Pengaruh
Biru (400-500 nm)	<ul style="list-style-type: none"> <li>• Meningkatkan kerapatan tanaman, perkembangan akar, dan produksi pada metabolisme sekunder (rasa dan aroma);</li> <li>• Meningkatkan akumulasi klorofil</li> <li>• Memfasilitasi pertukaran gas (pembukaan stomatal)</li> </ul>
Hijau (500-600 nm)	<ul style="list-style-type: none"> <li>• Memaksimalkan penyerapan cahaya</li> <li>• Mengatur pembukaan dan penutupan pori-pori tanaman</li> <li>• Meningkatkan efisiensi fotosintesis</li> </ul>
Merah (600-700 nm)	<ul style="list-style-type: none"> <li>• Merangsang terjadinya fotosintesis</li> <li>• Mendorong pertumbuhan biomassa dan vegetatif tanaman</li> <li>• Meningkatkan kandungan klorofil total dalam daun</li> </ul>

## 2.4 WeMos D1 R1 ESP8266 Wi-Fi Board

Wemos D1 R1 adalah *board* mikrokontroler yang berbasis pada ESP8266 Wi-Fi *board* dan menggunakan *design layout* dari Arduino. Espressif 8266 *Systems* adalah sebuah perusahaan *fabless semiconductor* yang menyediakan *wireless communication* dan Wi-Fi *chips* untuk penggunaan pada *mobile devices* dan *Internet of Things* (IoT) [16]. Perusahaan *fabless semiconductor* ini melakukan *outsourcing* fabrikasi mereka ke produsen khusus yang disebut *semiconductor foundry*, yang salah satunya adalah Wemos [17].

*Board* WeMos D1 R1 memiliki *flash memory* yang lebih besar dibandingkan dengan Arduino Uno yaitu sebesar 4mb dan total RAM sebesar 80kb. *Board* ini terdiri dari 11 pin *digital input/output* dan 1 pin *analog (input)* yang berjalan pada tegangan 3.3V. Terdapat 2 IDE yang dapat digunakan untuk memprogram WeMos D1 R1 yaitu Arduino IDE dan NodeMCU IDE [18]. Penulis menggunakan Arduino IDE dalam perancangan sistem usulan.

Dengan adanya fitur Wi-Fi *build-in*, penerapan teknologi *Internet of Things* (IoT) dapat dilakukan. Pada penelitian ini, sistem yang dibuat akan menggunakan WeMos D1 R1 sehingga data-data yang didapat dari berbagai sensor dapat ditampilkan pada *website* dan alat-alat dalam sistem hidroponik dapat dikontrol secara *remote*.

## 2.5 HyperText Markup Language (HTML)

HTML merupakan suatu *markup language* yang digunakan untuk membuat halaman *web*. HTML telah digunakan secara global sebagai *web scripting* untuk



melakukan format terhadap halaman *web*. Dikatakan sebagai *script* karena HTML berisi kode-kode program yang dapat diterjemahkan oleh komputer untuk menampilkan teks, gambar, grafik dan multimedia, namun tidak dapat melakukan *execute* pada kode-kode tersebut [19]. Dalam HTML terdapat berbagai *tag* yang digunakan untuk membuat komponen-komponen dalam *website* seperti *header*, paragraf, *link*, dan lain-lain. Dengan menggunakan *tag* inilah kerangka *website* dapat dibuat.

*Web scripting* seperti HTML bersifat *client-side*, artinya akan menghasilkan *website* yang statis, lebih menekankan pada format *design* dan informasi pada, serta tidak memiliki fungsionalitas yang dinamis [19]. Hal ini menyebabkan tidak adanya interaksi yang dinamis antara pengguna dan *website*. Namun, *tag-tag* di dalam file HTML akan membentuk struktur kode yang dapat digabungkan dengan bahasa lain seperti CSS, JavaScript dan PHP. Dengan demikian, HTML dapat dikombinasikan dengan *web scripting* lain yang bersifat *server-side* seperti PHP yang akan menjadikan *website* dinamis, dimana pengguna dapat mengirimkan *request* dan *website* akan mengeluarkan *output* yang sesuai.

## 2.6 Hypertext Preprocessor (PHP)

PHP adalah *web scripting* yang bersifat *server-side* dimana kode-kode program dalam PHP akan dijalankan atau diproses oleh server. PHP digunakan untuk membuat tampilan *website* yang dinamis, memproses sebuah *form*, dan dapat melakukan penyimpanan data-data ke dalam *database*. PHP bersifat *open-source*, dapat digunakan dengan gratis, dan sudah menjadi bahasa standar dalam pemrograman *web* di seluruh dunia. PHP dapat dijalankan pada berbagai sistem operasi seperti Windows, MacOS, dan Linux, serta mudah diinstalasikan ke dalam *web server* yang mendukung PHP seperti APACHE.

*Script* atau kode-kode PHP dapat disisipkan kedalam *script* HTML untuk membangun suatu *website* dinamis dengan kerangka design yang baik. *Script* PHP yang disisipkan pada *script* HTML akan diproses dan diinterpretasikan oleh server, kemudian server akan mengeluarkan hasilnya sebagai kode HTML pada *web browser* pengguna. Penulisan *syntax* PHP yang tepat adalah diawali dengan “<?php” dan diakhiri dengan “?>”. Dengan menggunakan *syntax* tersebut, *script* PHP dapat disisipkan di semua bagian *script* HTML, sehingga kombinasi kedua *script* tersebut akan menghasilkan *website* yang interaktif dan dinamis.

## 2.7 MySQL

*My Structured Query Language* (MySQL) adalah sebuah *Database Management System* (DBMS) yang menggunakan perintah dasar SQL. MySQL bekerja dengan arsitektur *client-server* dan dapat dijalankan pada berbagai sistem operasi seperti Windows, Linux, Unix, dan MacOS. Tipe data yang didukung oleh MySQL seperti *Character*, *String*, *Numeric*, *Bit-string*, *Date & Time*, *Boolean* dan *Timestamp* [26]. MySQL termasuk ke dalam jenis RDBMS (Relational Database Management System) dan bersifat *open source* dengan lisensi GNU *General Public License* (GPL) [27].

MySQL adalah salah satu *database* yang paling banyak digunakan dan memiliki performa yang baik. Keunggulan utama MySQL adalah ketersediaan dari MySQL *cluster servers* yang memfasilitasi *auto-sharding*, dimana data akan didistribusikan ke beberapa *server* yang berbeda. Dengan cara ini, data akan seimbang dan dapat selalu diakses walaupun dalam situasi *high-traffic*. MySQL menyediakan mekanisme keamanan yang kuat seperti otorisasi *user account* dan *password encryptions*. Selain itu, MySQL juga memanfaatkan fungsi dari komponen SQL seperti *views*, *stored procedures*, *triggers* dan *functions* yang menjadi fitur-fitur *powerful* dalam MySQL [26].

## 2.8 Ultrasonic Sensor (PING Sensor)

*Ultrasonic sensor* adalah alat yang digunakan untuk mengukur jarak antara sensor dan benda/objek. Sensor akan mengirimkan gelombang suara, dimana apabila terdapat objek dengan jarak tertentu di depan sensor, maka gelombang suara akan dipantulkan kembali menuju sensor. Sensor akan mencatat waktu saat gelombang suara dikirim dan diterima kembali, sehingga jarak objek dari sensor akan diketahui dengan rumus berikut [28]:

$$d = v \times t$$

Keterangan:

$d$  = *distance* (jarak)

$v$  = *velocity* (kecepatan)

$t$  = *time* (waktu)



Gambar 2.3 menunjukkan *ultrasonic sensor* yang digunakan dalam penelitian ini yaitu HC-SR04. Sensor ini mengeluarkan gelombang suara *ultrasonic* dengan frekuensi sekitar 40KHz yang tidak dapat didengar oleh manusia. HC-SR04 *ultrasonic sensor* bekerja dengan *power supply* 5V DC, dengan sudut efektif 15 derajat, dan rentang jarak yang dapat diukur sekitar 2-2400cm [28]. Sensor ini dipakai untuk mengukur tinggi air di dalam tangki larutan nutrisi. Nilai dari tinggi air ini diperlukan untuk menghitung total volume yang tersisa pada tangki tersebut.



Gambar 2.3 HC-SR04 *Ultrasonic Sensor*

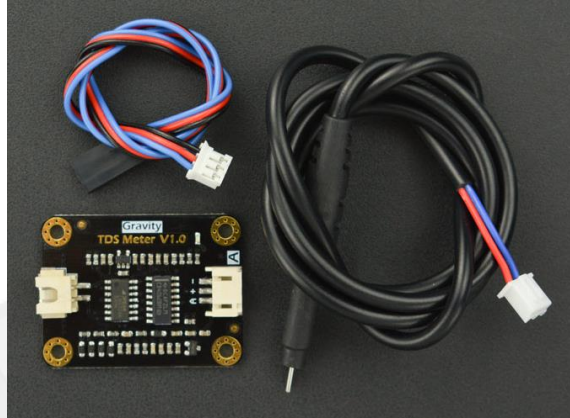
HC-SR04 ini mempunyai 4 pin yaitu VCC, GND, TRIG, dan ECHO dengan fungsi yang berbeda-beda. VCC terhubung pada 5V dan GND pada *ground* dari *power supply*. TRIG pin berfungsi untuk mengirimkan gelombang *ultrasonic*. Pin ini harus diperintah menjadi HIGH selama 10  $\mu$ s. Setelah gelombang *ultrasonic* dikirimkan, pin ECHO menjadi HIGH dan berubah menjadi LOW jika gelombang ultrasonic terdeteksi kembali. Pin ECHO mencatat waktu perubahan kondisi HIGH menjadi LOW yang dibutuhkan untuk menghitung jarak suatu objek [28].

## 2.9 TDS (*Total Dissolved Solid*) Meter

TDS (*Total Dissolved Solid*) meter adalah alat yang berfungsi untuk mengukur jumlah padatan terlarut pada suatu cairan. Satuan dari TDS meter adalah ppm (*Part per Million*, “bagian per sejuta”) yang menyatakan rasio dari jumlah kontaminan atau konsentrasi yang terdapat dalam suatu zat, dimana satu ppm adalah jumlah unit dibagi per sejuta unit [29]. Sebagai contoh pada larutan dengan konsentrasi 30ppm, menunjukkan bahwa setiap 1.000.000 bagian larutan hanya ada 30 bagian zat terlarut. Nilai ppm dapat digunakan untuk mengetahui kepekatan larutan nutrisi yang akan diberikan pada tanaman hidroponik.

Dalam penelitian ini, TDS meter yang digunakan Gravity: Analog TDS Meter seperti pada Gambar 2.4. Alat ini terdiri dari *signal transmitter board* dengan *input*

voltage 3.3 ~ 5.5V dan dapat mengukur dari 0 ~ 1000ppm, serta bagian TDS *probe* yang memiliki 2 jarum kecil dengan panjang kabel 83cm [30]. TDS meter ini *compatible* dengan Arduino atau mikrokontroler lain untuk menghasilkan nilai TDS dalam suatu cairan.



Gambar 2.4 Gravity: Analog TDS Meter

Nilai TDS dipengaruhi oleh nilai suhu suatu larutan, maka untuk mendapat hasil pengukuran yang lebih akurat, ditambahkan sensor suhu DS18B20. Nilai suhu ini akan digunakan untuk mencari nilai koefisien temperatur dengan rumus [32]:

$$\text{tempCoefficient} = 1.0 + 0.02 \times (\text{temp} - 25.0)$$

Keterangan:

tempCoefficient = *temperature Coefficient* (koefisien temperatur)

temp = hasil pengukuran temperatur

Kemudian dari hasil perhitungan koefisien temperatur, akan dihitung nilai EC (*Electrical Conductivity*) yaitu ukuran jumlah arus listrik yang dapat dihantarkan oleh suatu material [31]. Nilai EC ini juga dapat digunakan untuk mengetahui kepekatan suatu larutan namun memiliki satuan yang berbeda dari nilai ppm. Berikut adalah rumus untuk mencari nilai EC [32]:

$$\text{EC} = (\text{tdsValue} \times 5 / 1024.0) / \text{tempCoefficient} * \text{ecCalibration}$$

Keterangan:

tdsValue = hasil pengukuran TDS meter

tempCoefficient = *temperature Coefficient* (koefisien temperatur)

ecCalibration = nilai kalibrasi EC (nilai standar: 1)

Hasil dari nilai EC yang didapat akan digunakan untuk mencari nilai TDS dengan rumus sebagai berikut [31]:

$$\text{ppm} = (133.42 \times \text{EC}^3 - (255.86 \times \text{EC}^2) + (857.39 \times \text{EC})) \times 0.5$$

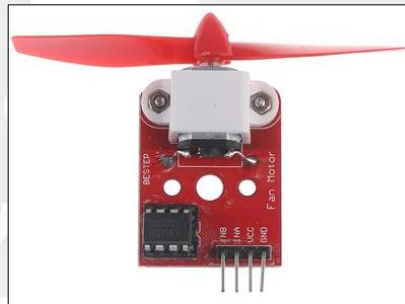
Keterangan:

ppm = nilai TDS dari suatu larutan

EC = *Electrical Conductivity*

## 2.10 Fan Motor L9110 Keyes Board

*Fan motor L9110 Keyes Board* merupakan modul motor DC kecil dengan *propellor* (kipas) yang terintegrasi dengan *L9110 motor driver* dan dapat mengatur arah rotasi dari motor dan kipasnya. *Working voltage* dari *fan motor* ini adalah 5V dan dapat dengan mudah meniup api pada korek api dengan jarak 20cm [33]. Gambar 2.5 adalah contoh dari *fan motor L9110 Keyes Board*:



Gambar 2.5 *Fan Motor L9110 Keyes Board*

*Fan motor* ini memiliki dua pin *input* yaitu INA dan INB. Kedua pin ini diperintah dengan *signal* HIGH dan LOW. Jika kedua pin diberi *signal* LOW maka *motor* akan diam. Jika pin INA HIGH dan INB LOW, maka *motor* akan berputar ke satu arah, namun akan berputar ke arah berlawanan saat INA LOW dan INB HIGH. Dua pin lain yaitu pin VCC dihubungkan pada 5V dan GND pada *ground* dari *power supply*. Pada penelitian ini, *fan motor* akan digunakan untuk mengaduk larutan nutrisi apabila terjadi penambahan air atau nutrisi ke dalam tangki.

## 2.11 Application Programming Interface (API)

*Application programming interface* atau API merupakan suatu *software intermediary* (perantara perangkat lunak) yang dapat mengintegrasikan antar

*software/aplikasi* agar saling berkomunikasi dengan mudah. Seorang *developer* dapat membuat suatu fungsi aplikasi dengan menggunakan API dari aplikasi lain. Salah satu jenis dari arsitektur API adalah REST atau *Representational State Transfer*. REST API sering digunakan karena pemakaiannya yang mudah dan bersifat lebih ringan.

Cara kerja REST API adalah dengan mengirimkan HTTP *request* untuk mengakses dan menggunakan data. Metode yang tersedia pada saat melakukan HTTP *request* adalah GET (membaca data), POST (membuat data baru), PUT (memperbarui data) dan DELETE (menghapus data). Setelah mengirimkan HTTP *request*, maka *client* akan menerima HTTP *response* yang berisi data hasil permintaan. Format data dari HTTP *response* dapat berupa JSON dan XML. Hasil data inilah yang kemudian dapat digunakan dalam aplikasi berbeda sesuai dengan kebutuhannya masing-masing.

## **2.12 JavaScript Object Notation (JSON)**

*JavaScript Object Notation* atau JSON adalah format pertukaran data yang bertujuan untuk menyimpan serta mentransfer data. Data JSON mudah dibaca dan ditulis oleh manusia, serta mudah dibuat dan diterjemahkan oleh mesin. JSON sering digunakan untuk pengiriman data dari server menuju suatu *web page*. Data dalam JSON dituliskan dengan *syntax* berupa *key* dan *value* yang berpasang-pasangan. *Code* program untuk membaca dan menghasilkan data JSON dapat ditulis dalam bahasa pemrograman apapun.

## **2.13 System Development Life Cycle (SDLC)**

SDLC merupakan siklus hidup dalam pengembangan sistem. Siklus hidup ini berupa tahapan-tahapan kerja yang digunakan untuk membangun suatu sistem informasi agar sesuai dengan hasil akhir yang ingin dicapai. Pada dasarnya terdapat 4 tahap dalam SDLC yaitu *planning*, *analysis*, *design* dan *implementation*. Namun seiring dengan berjalannya waktu, ada tahapan baru yang ditambahkan seperti *testing* dan *maintenance*. Semua tahapan tersebut memiliki karakteristik yang jelas dan tujuan yang berbeda antar satu dengan yang lainnya.

Pada tahap *planning*, dilakukan studi kelayakan pengembangan sistem (*feasibility study*). Beberapa kegiatan dalam tahap ini adalah pembentukan tim pengembang, menentukan tujuan dan ruang lingkup sistem, serta mengidentifikasi masalah apa

yang dapat diselesaikan melalui pengembangan sistem. Tahap selanjutnya adalah *analysis*, dimana para pengembang sistem akan melakukan analisis masalah, peluang dan solusi yang dapat diterapkan pada sistem. Selain itu juga mendefinisikan berbagai kebutuhan sistem yang akan dibuat. Tahap ketiga adalah *design*. Dalam tahap ini fitur yang ada pada sistem akan dideskripsikan secara detail dan menyeluruh. Para pengembang sistem juga membuat skema *database* dan merancang user *interface*. Selanjutnya pada tahap *implementation* akan dilakukan implementasi dari rancangan sistem yang telah dibuat dari tahap sebelumnya.

Setelah tahap implementasi selesai, maka dapat dikatakan sistem telah berhasil dibuat, namun tetap memerlukan tahapan tambahan yaitu *testing* dan *maintenance*. Pada tahap *testing* akan dilakukan uji coba sistem sebelum digunakan secara luas oleh pengguna. Uji coba ini memastikan sistem berjalan dengan baik tanpa adanya *bug*, dan jika ditemukan *bug* tertentu maka akan diselesaikan dalam tahap *testing* ini. Apabila tahap *testing* selesai, maka sistem akan siap digunakan. Tahap *maintenance* dilakukan untuk menjaga sistem agar tetap beroperasi dengan baik setelah digunakan secara keseluruhan oleh pengguna.

Salah satu model dari SDLC yaitu model *prototyping*, akan digunakan dalam penelitian ini. Dalam model *prototyping*, setelah tahap *planning* selesai, maka pengumpulan informasi mengenai kebutuhan sistem (tahap *analysis*) akan dilakukan dengan cepat. Hasil dari tahap *analysis* akan langsung menuju tahap *design* dan *implementation* yang kemudian menghasilkan sebuah *prototype*. Tahap *implementation* disini bersifat sementara, artinya hasil pengembangan sistem masih dapat berubah.

*Prototype* merupakan unit atau bagian sistem yang bersifat sementara dan dapat berubah yang akan diuji oleh pengguna sistem untuk dievaluasi. Setelah dievaluasi maka tahapan sistem akan kembali menuju tahap *analysis* dan terus berlanjut untuk memperbaiki dan mengembangkan *prototype* berdasarkan *feedback* dari pengguna sistem. Siklus ini akan berakhir apabila *prototype* sudah memenuhi seluruh kebutuhan sistem dan layak untuk digunakan. *Prototype* inilah yang akan diinstalasi dan menjadi sebuah sistem baru.



## 2.14 Unified Modeling Language (UML)

UML adalah suatu metode pemodelan visual untuk menggambarkan sistem yang berorientasi pada objek. Penelitian ini menggunakan UML 2.5 yang berisi sekumpulan teknik dan notasi standar yang bertujuan untuk memodelkan pengembangan sistem mulai dari analisis hingga implementasi. Teknik dan notasi standar tersebut dapat dibagi menjadi 2 kelompok utama yaitu pemodelan fungsional dan pemodelan struktural.


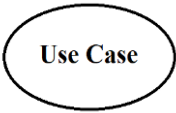
### 2.14.1 Pemodelan Fungsional

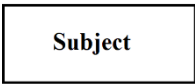
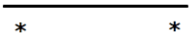
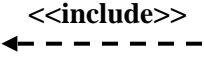
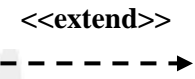

Dalam pemodelan fungsional, diagram dan notasi akan menggambarkan bagaimana proses bisnis yang terjadi, serta bagaimana interaksi antara sistem dan objek disekitar sistem. Diagram yang menggambarkan fungsionalitas dari sebuah sistem yaitu *use case diagram* dan *activity diagram*.

#### 2.14.1.1 Use Case Diagram

*Use case diagram* digunakan untuk menggambarkan secara sederhana aktor atau *user* yang berinteraksi dengan sistem dan bagaimana hubungan aktor tersebut dengan fungsi-fungsi sistem yang dapat mereka akses. *Use case diagram* sangat penting untuk memvisualisasikan, menjelaskan, dan mendokumentasikan berbagai kebutuhan sistem. Elemen-elemen yang terdapat dalam *use case diagram* sudah terstandarisasi agar dapat dipahami oleh semua orang. Elemen-elemen ini berisi *actor*, *use cases*, *subject boundaries*, dan kumpulan dari hubungan antar *actor*, *actor* dengan *use cases*, dan hubungan antar *use cases* [20]. Berikut adalah penejelasan dari masing-masing elemen:

Tabel 2.4 Elemen *Use Case Diagram*

Elemen	Simbol
<b>Actor</b> <ul style="list-style-type: none"><li>• Adalah seseorang atau sistem yang mendapat keuntungan dari dan bersifat eksternal terhadap subjek.</li><li>• Digambarkan dengan <i>stick figure (default)</i> atau persegi yang bertuliskan &lt;&lt;actor&gt;&gt; didalamnya jika <i>actor</i> bukan manusia.</li><li>• Diberi label sesuai dengan perannya.</li><li>• Dapat diasosiasikan dengan <i>actor</i> lain menggunakan asosiasi <i>specialization/superclass</i>.</li><li>• Terletak diluar dari <i>subject boundary</i>.</li></ul>	 Actor/Role
<b>Use Case</b> <ul style="list-style-type: none"><li>• Menggambarkan sebagian besar fungsionalitas sistem.</li><li>• Dapat <i>extend</i> menuju <i>usecase</i> lain.</li><li>• Dapat <i>include</i> terhadap <i>usecase</i> lain.</li><li>• Terletak didalam <i>subject boundary</i>.</li><li>• Diberi label dengan kata kerja - kata benda yang deskriptif.</li></ul>	 Use Case

Elemen	Simbol
<b>Subject Boundary</b> <ul style="list-style-type: none"> <li>Berisi nama subjek yang diletakan diaatas atau didalamnya.</li> <li>Merepresentasikan lingkup (<i>scope</i>) dari subjek, contohnya sistem atau proses bisnis.</li> </ul>	
<b>Association Relationship</b> <ul style="list-style-type: none"> <li>Menghubungkan <i>actor</i> dengan <i>use cases</i> yang berinteraksi dengannya.</li> </ul>	
<b>Include Relationship</b> <ul style="list-style-type: none"> <li>Merepresentasikan penyertaan fungsionalitas antar satu <i>use case</i> dengan <i>use case</i> lain.</li> <li>Memiliki tanda panah yang digambarkan dari <i>use case</i> dasar ke <i>use case</i> yang disertakan.</li> </ul>	
<b>Extend Relationship</b> <ul style="list-style-type: none"> <li>Merepresentasikan ekstensi dari fungsionalitas <i>use case</i> untuk menyertakan sesuatu yang opsional.</li> <li>Memiliki tanda panah yang digambarkan dari <i>use case</i> ekstensi ke <i>use case</i> dasar.</li> </ul>	
<b>Generalization Relationship</b> <ul style="list-style-type: none"> <li>Merepresentasikan <i>use case</i> yang lebih khusus ke <i>use case</i> yang lebih umum.</li> <li>Memiliki tanda panah yang digambarkan dari <i>use case</i> khusus ke <i>use case</i> dasar (<i>parent</i>)</li> </ul>	



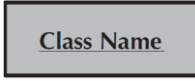

Tabel 2.4 Elemen *Use Case Diagram* (lanjutan)





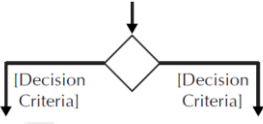
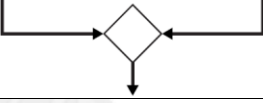


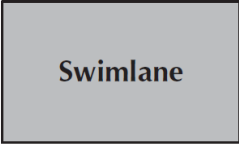
Sumber: Dennis et al., 2015, p. 122

#### 2.14.1.2 Activity Diagram

*Activity diagram* menggambarkan alur aktivitas yang terjadi di dalam sistem. Diagram ini menjelaskan berbagai jenis proses yang ada secara general. Activity diagram menunjukkan bagaimana suatu aktivitas dimulai, dan tahapan alur yang terjadi hingga aktivitas itu selesai. Berikut adalah elemen-elemen yang digunakan untuk membuat suatu activity diagram:

Tabel 2.5 Elemen *Activity Diagram*

Elemen	Simbol
<b>Action</b> <ul style="list-style-type: none"> <li>Adalah suatu bentuk perilaku yang sederhana dan tidak dapat didekomposisi.</li> <li>Diberi label sesuai namanya.</li> </ul>	
<b>Activity</b> <ul style="list-style-type: none"> <li>Merepresentasikan kumpulan dari <i>actions</i>.</li> <li>Diberi label sesuai namanya.</li> </ul>	
<b>Object Node</b> <ul style="list-style-type: none"> <li>Merepresentasikan sebuah objek yang terhubung pada suatu kumpulan <i>object flows</i>.</li> <li>Diberi label sesuai dengan nama kelasnya.</li> </ul>	
<b>Control Flow</b> <ul style="list-style-type: none"> <li>Menunjukkan urutan eksekusi.</li> </ul>	

Elemen	Simbol
<b>Object Flow</b> <ul style="list-style-type: none"> <li>Menunjukkan arus sebuah objek dari satu <i>activity</i> (atau <i>action</i>) ke <i>activity</i> (atau <i>action</i>) lain.</li> </ul>	
<b>Initial Node</b> <ul style="list-style-type: none"> <li>Menggambarkan awal dari sekumpulan <i>action</i> atau <i>activity</i>.</li> </ul>	
<b>Final-activity Node</b> <ul style="list-style-type: none"> <li>Digunakan untuk menghentikan setiap <i>control flow</i> dan <i>object flow</i>.</li> </ul>	
<b>Final-flow Node</b> <ul style="list-style-type: none"> <li>Digunakan untuk menghentikan <i>control flow</i> atau <i>object flow</i> yang spesifik.</li> </ul>	
<b>Decision Node</b> <ul style="list-style-type: none"> <li>Merepresentasikan suatu kondisi pengujian untuk memastikan <i>control flow</i> atau <i>object flow</i> hanya berada di satu jalur.</li> <li>Diberi label sesuai dengan kriteria keputusan (<i>decision</i>) untuk melanjutkan ke jalur yang spesifik.</li> </ul>	
<b>Merge Node</b> <ul style="list-style-type: none"> <li>Digunakan untuk menyatukan kembali jalur keputusan yang berbeda yang dihasilkan dari <i>decision node</i>.</li> </ul>	
<b>Fork Node</b> <ul style="list-style-type: none"> <li>Digunakan untuk memisahkan perilaku (<i>behavior</i>) menjadi sekumpulan <i>flow activity/action</i> yang paralel atau bersamaan.</li> </ul>	
<b>Join Node</b> <ul style="list-style-type: none"> <li>Digunakan untuk menyatukan kembali sekumpulan <i>flow activity/action</i> yang paralel atau bersamaan.</li> </ul>	
<b>Swimlane</b> <ul style="list-style-type: none"> <li>Digunakan untuk memisahkan <i>activity diagram</i> menjadi beberapa baris dan kolom untuk menetapkan setiap <i>activity</i> atau <i>action</i> kepada individu atau objek yang akan mengeksekusi <i>activity</i> atau <i>action</i> tersebut.</li> <li>Diberi label dengan nama individu atau objek yang bertanggung jawab atasnya.</li> </ul>	

Tabel 2.5 Elemen *Activity Diagram* (lanjutan)

Sumber: Dennis et al., 2015, p. 132

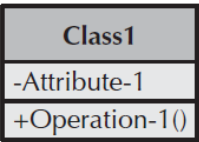
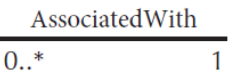

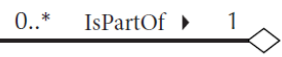

### 2.14.2 Pemodelan Struktural

Pemodelan struktural merupakan model yang menggambarkan objek yang dibentuk dan digunakan dalam sebuah sistem informasi seperti orang, benda, tempat, atau informasi yang disimpan. Selain itu juga menggambarkan bagaimana hubungan antar objek-objek tersebut. Dalam pemodelan struktural, diagram yang digunakan adalah *class diagram*.

*Class diagram* adalah diagram yang menunjukkan hubungan antar class yang bersifat statis, konstan dari waktu ke waktu dalam sistem. Dikatakan bersifat statis, karena *class diagram* tidak menggambarkan apa yang terjadi jika antar *class* saling berhubungan, melainkan hubungan apa yang terjadi. Diagram ini berisi kelas-kelas, perilaku, keadaan

(*state*), dan hubungan antar kelas. Berbagai elemen yang ada dalam class diagram mampu memberikan gambaran yang lebih luas mengenai suatu sistem. Berikut adalah elemen-elemen yang digunakan untuk membuat suatu *class diagram*:

Tabel 2.6 Elemen *Class Diagram*

Elemen	Simbol
<b>Class</b> <ul style="list-style-type: none"> <li>Merepresentasikan orang, tempat, atau objek lain yang informasinya perlu diambil dan disimpan oleh sistem.</li> <li>Diberi label dengan nama yang dicetak tebal dan diletakan dibagian tengah atas.</li> <li>Memiliki daftar atribut di bagian tengah.</li> <li>Memiliki daftar operasi (<i>operation</i>) di bagian bawah.</li> <li>Tidak secara eksplisit menunjukkan operasi yang tersedia untuk semua <i>class</i>.</li> </ul>	
<b>Attribute</b> <ul style="list-style-type: none"> <li>Merepresentasikan properti yang mendeskripsikan kondisi dari suatu objek.</li> <li>Dapat diturunkan atau diambil dari atribut lain, dianotasikan dengan meletakkan garis miring sebelum nama atribut.</li> </ul>	<i>attribute name</i> <i>/derived attribute name</i>
<b>Operation</b> <ul style="list-style-type: none"> <li>Merepresentasikan aksi atau fungsi yang dapat dilakukan oleh sebuah <i>class</i>.</li> <li>Dapat diklasifikasikan sebagai sebuah <i>constructor</i>, <i>query</i>, atau operasi <i>update</i>.</li> <li>Terdapat tanda kurung yang mungkin berisi informasi yang dibutuhkan untuk melakukan operasi.</li> </ul>	<i>operation name ()</i>
<b>Association</b> <ul style="list-style-type: none"> <li>Merepresentasikan hubungan antara beberapa <i>class</i> atau <i>class</i> dengan dirinya sendiri.</li> <li>Diberi label dengan kata kerja atau nama peran.</li> <li>Dapat muncul diantara satu atau lebih <i>class</i>.</li> <li>Berisi simbol <i>multiplicity</i>, yang merepresentasikan jumlah minimal dan maksimal suatu instansi <i>class</i> dapat diasosiasikan dengan instansi <i>class</i> yang berhubungan</li> </ul>	
<b>Generalization</b> <ul style="list-style-type: none"> <li>Merepresentasikan hubungan <i>a-kind-of</i> antara beberapa <i>class</i></li> </ul>	
<b>Aggregation</b> <ul style="list-style-type: none"> <li>Merepresentasikan hubungan <i>a-part-of</i> yang logis antara beberapa <i>class</i>, atau antara suatu <i>class</i> dengan dirinya sendiri.</li> <li>Merupakan bentuk khusus dari <i>association</i>.</li> </ul>	
<b>Composition</b> <ul style="list-style-type: none"> <li>Merepresentasikan hubungan <i>a-part-of</i> yang fisik antara beberapa <i>class</i>, atau antara suatu <i>class</i> dengan dirinya sendiri.</li> <li>Merupakan bentuk khusus dari <i>association</i>.</li> </ul>	

Sumber: Dennis et al., 2015, p. 178

## 2.15 Testing

*Testing* adalah suatu proses atau aktivitas yang dilakukan untuk menguji dan mengidentifikasi adanya perbedaan antara sistem yang telah selesai dibuat dengan apa yang diharapkan pengguna. Dengan melakukan testing maka para pengembang sistem

dapat menemukan *error* atau *bugs* yang ada, dan selanjutnya dapat melakukan perbaikan. Saat proses *testing* terjadi, perlu adanya dokumentasi terhadap bagian-bagian sistem yang sedang diuji.

Ada beberapa jenis *testing* yang umum dilakukan yaitu seperti *system testing*, *unit testing*, *integration testing*, *usability testing*, dan *acceptance testing*. Pada penelitian ini, *testing* yang akan digunakan adalah *unit testing*. Pada *unit testing*, pengujian dilakukan terhadap setiap bagian kecil *code* atau program yang dibuat, memastikan setiap modul atau bagian tersebut sudah berjalan dengan baik sebelum diintegrasikan menjadi suatu sistem utuh.

## 2.16 Penelitian Terdahulu

Tabel 2.7 Penelitian Terdahulu

No	Judul	Penulis	Publikasi	Deskripsi	Aplikasi Usulan
1	<i>SmartGreenHouse: Prototipe Sistem Monitoring Budidaya Tanaman Hidroponik Tipe Nutrient Film Technique (NFT) Berbasis Web</i> [22]	Helmy, Mita Savira Anggraeni, Abu Hasan, Arif Nursyahdid dan Thomas Agung Setyawan	Prosiding Sentrinov (2017)	Penelitian ini bertujuan untuk membangun dan menguji prototipe sistem monitoring pH, EC, suhu larutan nutrisi hidroponik NFT dan suhu <i>greenhouse</i> berbasis web.	Sistem yang ingin dibuat adalah sistem yang dapat memonitor nilai dari pH, EC, suhu larutan nutrisi dan suhu di dalam <i>greenhouse</i> dari web, sehingga hasil <i>monitoring</i> tersebut dapat dilihat dimana saja baik <i>offline</i> maupun <i>online</i> . Fitur utama dalam penelitian ini adalah menampilkan informasi <i>monitoring</i> ke dalam <i>website</i> dengan <i>user interface</i> yang sederhana dan mudah dipahami.
2	<i>Smart Hydro System Sebagai Solusi Otomasi Pemeliharaan Pertanian Hidroponik</i> [5]	Michelle Kartosugondo, Felicia Leliana dan Agnes Yolanda	Prosiding SNST ke-9 (2018)	Penelitian ini membahas pentingnya sistem yang dapat memonitor kuantitas air secara otomatis dan terintegrasi pada hidroponik NFT, sehingga	Penulis menjelaskan masalah yang dihadapi petani hidroponik adalah dalam memonitor suhu dan kuantitas air yang diperlukan, dan tidak adanya <i>warning system</i>



No	Judul	Penulis	Publikasi	Deskripsi	Aplikasi Usulan
				dapat mengurangi kegagalan panen akibat kesalahan penanganan dan kuantitas air.	dalam memonitor suhu dan kuantitas air tersebut. Maka dalam penelitian ini sistem yang akan dibangun adalah sistem yang dapat memonitor suhu dan kuantitas air, serta melakukan pengairan pada pertanian hidroponik secara otomatis.
3	Otomasi Pencampur Nutrisi Hidroponik Sistem NFT ( <i>Nutrien Film Technique</i> ) Berbasis Arduino Mega 2560 [23]	Nuris Dwi Setiawan	Jurnal Teknik Informatika Unika St. Thomas (JTIUST), Vol. 3 No. 2 (2018)	Penelitian ini membahas sistem hidroponik NFT dimana pemberian nutrisi harus dengan takaran yang tepat, terdiri dari pencampuran pupuk dan air. Namun hal ini sistem NFT ini membutuhkan waktu yang banyak untuk mengamati stok nutrisi tersebut.	Sistem usulan dalam penelitian ini adalah sistem otomasi yang menggunakan Arduino Mega 2560 dan terintegrasi dengan berbagai sensor. Sistem otomasi untuk hidroponik ini akan mengatur pompa, LED <i>grow light</i> , sistem pemupukan dan <i>water flow</i> . Hasil yang ingin dicapai adalah sistem hidroponik dengan teknik NTF yang mempunyai otomasi dalam pencampuran nutrisi.
4	Sistem Monitoring Kelembapan Tanah, Suhu, pH, dan Penyiraman Otomatis Pada Tanaman Tomat Berbasis <i>Internet of Things</i> [24]	Rudy Gunawan, Tegas Andhika, Sandi dan Fadil Hibatulloh	Jurnal Telekomtran Vol. 7, No. 1 (2019)	Penelitian ini bertujuan untuk membuat sistem yang dapat mengetahui informasi parameter ukur yakni kelembapan tanah, suhu, pH serta penyiraman dan pemupukan cair yang dilakukan secara otomatis.	Sistem usulan berupa sistem rekayasa teknologi yang dapat membantu aktivitas petani dengan memberikan informasi melalui aplikasi <i>smartphone</i> ( <i>Blynk</i> ) yang sudah tersambung dengan sistem. Informasi yang di dapat berupa data parameter ukur, kondisi lahan

No	Judul	Penulis	Publikasi	Deskripsi	Aplikasi Usulan
					seperti kelembapan tanah, suhu, pH. Selain itu terdapat fitur penyiraman dan pemupukan secara otomatis pada tanaman tomat di atas lahan yang telah di tentukan ukurannya sehingga dapat merata saat penyiraman dan pemupukannya.
5	<i>Automated Hydroponics Nutrition Plants Systems Using Arduino Uno Microcontroller based on Android</i> [25]	Amy Lizbeth J. Rico	<i>Journal of Engineering and Applied Sciences</i> (2020)	Penelitian ini bertujuan untuk membangun alat pengontrolan pengaliran nutrisi hidroponik secara otomatis menggunakan mikrokontroler Arduino dan diakses melalui smartphone.	Sistem usulan yang dibuat akan memanfaatkan Arduino Uno sebagai ‘otak’ dari alat-alat yang akan memonitor tanaman hidroponik, menggunakan sensor untuk mendeteksi ketinggian larutan nutrisi, sehingga apabila volume air kurang dari nilai batas, maka air dan nutrisi akan dicampurkan kemudian baru dialirkan menuju tanaman.

Tabel 2.7 Penelitian Terdahulu (lanjutan)