

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. Penelitian terkait**

Menurut penelitian dari Achmad Gildas Thauty, Suthami Ariessaputra, Cahyo Mustiko Okta Muvianto (2024) Yang berjudul Sistem Monitoring Pakan dan Air Minum Burung Peliharaan Pemakan Biji Berbasis *Internet Of Things*. Burung kicau pemakan biji-bijian sering dijadikan sebagai hobi untuk dipelihara. Burung ini memiliki warna bulu yang indah dan bersuara merdu. Salah satu kendala dalam memelihara burung kicau adalah ketika pemelihara memiliki waktu yang terbatas. Kesibukan pemilik burung yang bekerja dan pergi keluar kota dapat menyebabkan burung tidak memperoleh pakan dan air yang memadai, selain itu pemilik juga tidak dapat mengawasi kondisi kesehatan burung. Berdasarkan hal tersebut, maka dibuat sistem pemberi dan monitoring pakan burung kicau berbasis *IoT* dengan fitur berupa monitoring sisa makanan dan minuman dalam tangki menggunakan NodeMCU ESP32 sebagai kontrol sistem dan pengirim data ke internet. Sistem sudah dapat memberikan pakan dan air minum untuk burung peliharaan secara otomatis serta terhubung melalui koneksi internet dengan *Blynk* untuk mengirimkan hasil monitoring. Konsumsi pakan harian burung kicau pemakan biji berdasarkan hasil monitoring yaitu sebesar 4,74 gram per hari atau setara dengan 43,14 % dari berat badan burung tersebut [1].

Menurut penelitian dari Angelina Septiani Vania Putri, Gunawan Pria Utama (2023). dalam jurnal penelitiannya yang berjudul *Prototipe Internet Of Thing* Pakan Kucing Otomatis, Dengan Wemos D1r1, Ultrasonik Dan *Water Level*. Prototipe Sistem Pakan Otomatis, menyimpulkan bahwa semua sensor dan perangkat kontrol telah berhasil diimplementasikan dengan baik sesuai fungsinya. Namun, dalam pengujian Sistem Otomatis pada tanggal 7 Juli 2023, terdapat masalah delay selama 3 detik yang disebabkan oleh kendala koneksi internet dan pasokan listrik. Sebagai saran untuk meningkatkan kinerja dan fungsionalitas Sistem Pakan Otomatis ke depannya, kami merekomendasikan hal-hal berikut. Pertama, upayakan untuk meminimalisir delay dalam mengontrol alat dengan memastikan koneksi internet yang stabil dan handal. Kedua, tambahkan alat sensor tambahan yang dapat memantau apakah hewan sedang makan atau tidak, sehingga pemilik dapat lebih baik memahami pola makan hewan peliharaan mereka. Terakhir, kami menyarankan untuk mengimplementasikan fitur notifikasi pada aplikasi, sehingga pemilik dapat menerima pemberitahuan jika terjadi situasi khusus atau ketika pakan atau air hewan peliharaan sudah habis. Dengan memperhatikan saran-saran di atas, diharapkan Sistem Pakan Otomatis dapat berjalan dengan lebih baik lagi dan memberikan kenyamanan serta kemudahan bagi pemilik hewan peliharaan dalam merawat kucing mereka [2].

Menurut penelitian dari Wahyu Agung Nurcahyo, Arif Faizin (2023). Yang berjudul Rancang Bangun Sistem Pemberi Pakan Dan Minum Otomatis Dalam peternakan burung puyuh, *Internet of Things (IoT)* Peternak harus memastikan bahwa burung puyuh mereka dirawat dengan baik, termasuk memberi mereka pakan dan air minum tepat waktu. Jika mereka tidak melakukannya, hal itu akan menyebabkan hasil yang buruk akan berpengaruh terhadap kesehatan dan produksi telur burung puyuh. Penelitian ini mengembangkan *prototype* sistem pemberi pakan dan minum otomatis menggunakan NodeMCU ESP32 dengan metode studi kasus. Tujuan penelitian ini adalah untuk untuk mengembangkan sebuah sistem otomatis untuk memudahkan peternak dalam memberikan pakan dan minum puyuh. Setelah dilakukan pengujian kerja *system*. Hasil yang didapatkan selama 7 hari pengujian adalah memberikan berat puyuh dan hasil telur lebih baik, menunjukkan bahwa sistem dapat bekerja dengan memberikan pakan secara otomatis sesuai jadwal jam yang telah diberikan, sehingga bisa meningkatkan efisiensi waktu. Informasi saat penjadwalan terhubung ke *firebase*, saat pemberian pakan, katup pakan terbuka rata-rata 1,55 detik dan menghasilkan berat 47 gram pakan yang keluar. Peternak juga bisa memilih opsi untuk pengecekan volume pakan dan air pada wadah lewat bot telegram [3].

Menurut penelitian dari Arkan Fadillah, Purwanto, Utomo Budiyanoto, Safrina Amini yang berjudul Sistem Monitoring dan Pemberian Pakan Ikan Hias Otomatis Berbasis *Website*. Salah satu faktor penting untuk meningkatkan persentase keberhasilan penggemar atau penghobi ikan hias

dalam memelihara ikan hias adalah dengan memperhatikan kualitas benih, kualitas pakan, dan sistem pengolahan air tetapi pada pelaksanaannya sistem pengelolaan kualitas air dan pemberian pakan masih dilakukan oleh manusia, tetapi masalah yang sering terjadi adalah pemberian pakan ikan hias tidak tepat waktu bahkan terlupakan karena kesibukan dari penghobi ikan hias tersebut dan monitoring suhu air tidak dilakukan secara rutin dan realtime. Untuk mengatasi masalah pada pemberian pakan dan monitoring suhu air dirancanglah sebuah sistem pemberian pakan hias otomatis dan monitoring suhu berbasis *internet of things* sehingga dapat membantu pekerjaan penggemar atau penghobi ikan hias, Tujuan penelitian adalah membuat sistem monitoring dan pemberian pakan ikan hias otomatis dengan sensor suhu DS18B20, ESP32Cam dan mikrokontroler NodeMCU ESP8266 berbasis *website* Berdasarkan hasil pengujian terhadap sistem pemberian pakan otomatis ini sistem berhasil memberikan pakan ikan hias sesuai jadwal dan mengirim foto dari habitat ikan hias ke *web server* dengan waktu jeda (*delay*) selama 1 sampai 3 detik, Monitoring suhu berjalan dengan baik dengan suhu normal sampai dengan 270C sehingga buzzer akan aktif jika suhu air di atas suhu 270C. Sistem monitoring berbasis *website* berjalan dengan baik dengan menampilkan pembacaan dari sensor dan laporan harian dari pembacaan sensor [4].

Menurut penelitian dari Agus Mulyono, Novi Lailiyul Wafiroh, Muthmainnah (2024). Yang berjudul Pelatihan Beternak Kenari Sistem Tanpa Jemur Dengan Teknologi Pencahayaan. Kenari merupakan salah satu jenis

burung yang cukup banyak digandrungi karena dikenal memiliki kicauan yang merdu dan menawan. Ada banyak alasan kenapa ternak burung kenari ini terbilang menguntungkan dan dapat memberikan penghasilan bulanan. Beberapa alasan tersebut antara lain: Harganya cenderung stabil, menjadi salah satu burung berkicau kelas atas yang banyak peminatnya, sangat sering diadakan kontes, Perkembangbiakannya tergolong cepat, modal yang dibutuhkan tidak terlalu besar atau bisa disesuaikan dengan budget, cara ternaknya tidak terlalu sulit hanya butuh ketelatenan (Kurnia et al., 2019). Telah dilakukan pengabdian masyarakat berjudul peningkatan keterampilan dan produktivitas dalam berternak burung kenari melalui teknologi pencahayaan (tanpa jemur) pada remaja di perumahan sawojajar kota malang Desa Sekarpuro Kecamatan Pakis Kabupaten Malang. Pengabdian ini dilakukan untuk beberapa tujuan diantaranya adalah meningkatkan pengetahuan masyarakat tentang berternak burung kenari dan meningkatkan produktivitas berternak burung dengan pencahayaan lampu. Metode pengabdian yang dilakukan dengan menggunakan *Participatory Action Research (PAR)*, dimana terdapat tiga rangkaian kegiatan dalam pengabdian ini, yaitu sosialisasi tentang berternak burung kenari, pelatihan teknik pencahayaan untuk meningkatkan produktivitas burung dan kunjungan ke peternak burung. Antusiasme masyarakat sangat tinggi saat kegiatan sosialisasi berternak burung. Terbukti dengan banyaknya pertanyaan yang masuk waktu sesi tanya jawab. Pelatihan ini melibatkan 12 orang peternak burung yang masih pemula. Setiap peserta mendapatkan 3 burung kenari

dengan rincian dua betina dan satu jantan serta 3 buah sangkar untuk masing-masing burung, serta stok pakan burung selama 1 bulan. Pemberian pakan pada burung kenari sangat penting untuk menunjang kehidupan, pertumbuhan serta untuk mengganti sel-sel yang rusak (Siregar, 2020). Jika burung kenari diberi pakan yang salah maka akan mengakibatkan pertumbuhan kurang baik dan tidak bersuara merdu. Pakan burung kenari harus memenuhi kebutuhan yang diperlukan oleh tubuh, seperti air, karbohidrat, lemak dan protein. Selain itu vitamin dan mineral juga diperlukan dalam memberi pakan burung kenari [5].

Menurut penelitian dari Himawan Sugiritno, Denny Irawan (2024). yang berjudul Sistem Monitoring Kelayakan Sangkar Dan Air Minum Otomatis Peternakan Kambing Berbasis *Website*. Sangkar merupakan habitat hewan yang membuatnya merasa aman dan bebas dari gangguan luar. Sangkar yang baik memiliki komponen internal seperti ukuran dan ventilasi yang tepat, Jika sangkar tidak dirancang untuk memenuhi kebutuhan ternak, stres dapat terjadi dan berdampak Negatif terhadap kinerja produksi susu. Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem pemantauan air minum otomatis pada peternakan kambing menggunakan sensor *ultrasonic* berbasis *website*. Sistem ini dirancang untuk mengukur ketinggian udara dalam ember secara *real-time* dan mengontrol pengisian udara secara otomatis menggunakan *solenoid valve* yang dikendalikan oleh mikrokontroler Wemos D1 R2. Data ketinggian udara yang dikirimkan ke *server web* melalui koneksi *Wi-Fi*, sehingga pengguna dapat menjaga kondisi

secara langsung melalui antarmuka *website* yang ramah pengguna. Selain itu, sistem ini juga dilengkapi dengan sensor MQ-135 untuk mendeteksi kualitas udara, khususnya kadar amonia, yang merupakan indikator penting dalam menjaga kesehatan kambing. Jika kadar amonia terdeteksi tinggi, sistem akan mengaktifkan kipas DC untuk meningkatkan sirkulasi udara, sehingga menciptakan lingkungan yang lebih sehat dan nyaman bagi hewan ternak. Hasil pengujian menunjukkan bahwa sistem ini tidak hanya dapat meningkatkan efisiensi operasional dalam pemeliharaan air minum, tetapi juga memastikan ketersediaan air minum yang konsisten bagi ternak, serta menjaga kualitas lingkungan di dalam sangkar. Dengan demikian, sistem ini diharapkan dapat menjadi solusi efektif untuk meningkatkan produktivitas dan kesejahteraannya [6].

Menurut penelitian dari Saharman, Fatma Agus Setyanigsih, Suhardi (2022) Yang berjudul *Monitoring Dan Kontrol Pemberian Pakan Dan Minum Pada Peternakan Burung Puyuh Berbasis Internet Of Things* Salah satu aspek penting yang harus diperhatikan dalam budidaya burung puyuh adalah pada pemberian pakan dan minum. Namun dalam implementasinya pemberian pakan dan minum burung puyuh umumnya masih dilakukan secara manual, pembudidaya harus menaburkan pakan setiap harinya dengan jumlah pakan dan waktu yang tidak teratur. Oleh karena itu, dibuatlah sistem yang dapat me-monitoring dan mengontrol pemberian pakan dan minum burung puyuh secara otomatis berbasis *Internet of Things*. Pada penelitian ini NodeMCU ESP32 digunakan sebagai kendali keseluruhan sistem. *Input* yang digunakan

sensor *ultrasonic*. Sedangkan untuk *output* adalah motor *servo*, konveyor dan pompa air. Hasil pengujian menunjukkan bahwa pemberian pakan burung puyuh berhasil yang terjadwal setiap pagi dan sore, pemberian minum juga berhasil dengan membaca nilai sensor *ultrasonic* [7].

Menurut penelitian dari Ega Mei Prianto, Rahmadani Pane, Rohani (2024). yang berjudul Rancang Bangun Alat *IoT* Memberi Pakan Ikan Lele Secara Otomatis Berbasis Telegram. Rancang bangun alat *IoT* pemberi pakan ikan secara otomatis berbasis telegram, untuk memudahkan peternak ikan memberi pakan ikan lele saat berpergian jauh. Penelitian ini menggunakan *metode Research And Development*. Metode ini merupakan langkah-langkah membuat dan mengembangkan suatu produk alat. Berdasarkan dari penelitian, alat ini dirancang untuk memudahkan pemberian pakan ikan lele secara otomatis atau perintah yang diterima melalui aplikasi telegram, dan pengguna dapat mengontrol dan memantau proses pemberian pakan dari jarak jauh. Penelitian ini mengimplementasikan konsep-konsep desain perangkat keras dan perangkat lunak yang diperlukan untuk operasi yang efisien dan dapat diandalkan dalam budidaya ikan lele. Hasil pengujian menunjukkan bahwa alat ini mampu meningkatkan efisiensi pengelolaan pakan ikan lele dan memberikan kontrol pakan ikan lele yang lebih baik. Dengan demikian, penggunaan alat *IoT* berbasis telegram ini diharapkan dapat memberikan kontribusi positif dalam meningkatkan produktivitas dan keberlanjutan dalam budidaya ikan lele secara modern [8].

Menurut penelitian dari Muhtarudin Khoiri, Bambang Hari Purwoto (2025). Yang berjudul Rancang Bangun Alat Pemberi Pakan Dan Minum Pada Burung Kicau Menggunakan Esp32 Berbasis *IoT*. Burung kicau merupakan hewan peliharaan yang banyak disukai oleh masyarakat Indoneisa. Karena burung kicau memiliki suara yang merdu sehingga dapat memberikan kesenangan tersendiri untuk pemelihara. Dalam memelihara burung kicau, pemberian pakan dan minum sangat mempengaruhi kesehatan dan tumbuh kembang burung. Pemberian pakan dan minum burung kebanyakan masih banyak dengan cara manual. Sering kali para pemelihara burung lupa mengasih pakan dan minum pada burungnya sehingga mengakibatkan burung sakit bahkan sampai mati. Maka dari itu penulis membuat alat yaitu pemberi pakan dan minum otomatis berbasis *internet of things* sehingga memudahkan pemelihara burung untuk memantau pakan dan minum pada burung melalui *smartphone*. Alat ini dibuat menggunakan mikrokontroler ESP32, sensor *LDR* digunakan untuk mengukur keadaan pakan yang ada didalam sangkar, *motor servo* untuk membuka/menutup saluran makan yang ada di sangkar, sensor *water level* untuk mengukur kedalaman air yang ada dalam wadah air dalam sangkar, *solenoid valve* digunakan untuk membuka dan menutup saluran air dari penampungan air diluar sangkar kedalam wadah air dalam sangkar, *Relay* sebagai skalar *on/off solenoid valve*. Pengguna nantinya juga dapat memonitoring ketersediaan pakan dan minum burungnya melalui *smartphone* melalui aplikasi *blynk*. Hasil dari pengujian alat ini dapat disimpulkan bahwa sensor *LDR* dengan

akuator *motor servo* yang dimanfaatkan sebagai gerbang pengisian pakan memiliki tingkat akurasi 100% dalam pengisian pakan dalam wadah pakan yang ada didalam sangkar. Sensor *water level* dengan akuator *solenoid valve* yang dimanfaatkan sebagai gerbang pengisian air minum burung ketika wadah air dalam kendang kondisi kurang air sudah sesuai berjalan sesuai dengan keinginan dari penulis. Data pembacaan dari sensor *LDR* dan sensor *water level* yang dikirimkan pada aplikasi *blynk* dapat memudahkan dalam monitoring jarak jauh dan tombol *button* pakan minum bisa mengontrol pakan minum secara jauh [9].

Menurut penelitian dari Agustinus Yasvin Nahak, Yoseph P.K Kelen, Budiman Baso, Willy Sucipto (2024). Dalam jurnal penelitiannya yang berjudul Otomatisasi Pakan Minum dan Monitoring Kelembaban Udara pada Sangkar Ayam berbasis *IoT*. Dengan menggunakan *Internet of Things (IoT)*, *prototipe* sangkar ayam broiler dapat bekerja secara otomatis. Hal ini memudahkan bagi pengguna jika sedang tidak secara terus menerus memantau sangkarnya. Program yang dirancang dengan konsep ini juga dapat melakukan hal-hal yang diharapkan, seperti mengupload data ke *server*, menampilkan data kelembaban, dan menampilkan waktu secara *realtime*. Node MCU sudah dilengkapi juga dengan *module wifi*, yang memudahkan pengiriman data monitoring *server* dan mengontrol keadaan sangkar ayam dari mana saja dan kapan saja selama pengguna terkoneksi dengan jaringan internet. Dengan merujuk pada hasil yang telah diuraikan pada kesimpulan di atas, sistem pemberi pakan minum dan monitoring kelembaban udara ini

masih jauh dari sempurna. Ada beberapa saran yang dapat dimanfaatkan untuk mengembangkan penelitian lebih lanjut, seperti menambahkan sensor *ultrasonic* HC-SR04 untuk pengukur ketinggian air minum dan pakan di wadah, sehingga jika air dan pakan masih banyak maka tidak perlu dilakukan pengisian lagi. Kemudian menambahkan sensor *water level* yang berfungsi untuk melakukan pengecekan ketinggian air minum yang ditempatkan dalam wadah. Dan terakhir menambahkan penghangat sangkar ayam berupa lampu. Jika kelembaban udara meningkat, maka lampu akan menyala sehingga suhu ruangan dapat hangat kembali agar ayam tidak kedinginan [10].

## **2.2. Landasan Teori**

### **2.2.1. Otomatisasi**

Otomasi adalah penciptaan dan penerapan teknologi untuk memantau dan mengendalikan produk, serta memberikan produk dan layanan (*International Society of Automation*). Otomasi juga dapat diartikan sebagai proses yang secara otomatis mengelola operasi dan perlengkapan mekanik atau elektronika yang berfungsi sebagai pengganti manusia untuk melihat dan membuat keputusan. Dari penjelasan yang disampaikan dapat dikatakan bahwa otomatisasi merupakan implementasi teknologi dengan meminimalkan keterlibatan manusia sehingga mewujudkan pemanfaatan berbagai sistem kendali terhadap pengoperasian peralatan [10].

### 2.2.2. Monitoring

Monitoring berarti pengamatan, pemeriksaan, pengendalian, dan pengoreksian aktivitas yang hendak diketahui. Sistem monitoring juga disebut sebagai upaya sistematis untuk menetapkan kinerja standar untuk perencanaan untuk merancang sistem umpan balik informasi; tujuan dari sistem ini adalah untuk membandingkan kinerja aktual dengan standar yang telah ditentukan [10].

### 2.2.3. Notifikasi

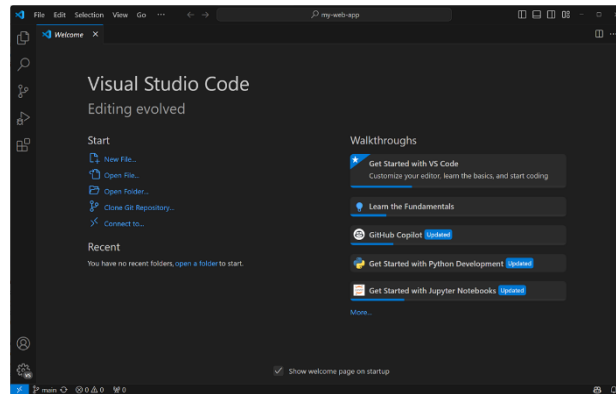
Notifikasi adalah pemberitahuan mengenai tindakan yang memengaruhi mereka. Pemberitahuan ini dapat membantu pengguna lebih menyadari peristiwa-peristiwa yang berhubungan dengan mereka dan, jika mereka mau, dapat mengambil tindakan dengan cepat [11].

### 2.2.4. Visual Code

Visual Studio Code adalah *source code editor* yang dikembangkan Microsoft yang bisa digunakan pada berbagai platform (Linux, Mac OS dan Microsoft). Visual Studio Code adalah *Software* yang sangat ringan, namun kuat editor kode sumbernya yang berjalan dari desktop. Muncul dengan *built-in* dukungan untuk JavaScript, naskah dan Node.js dan memiliki *array* beragam ekstensi yang tersedia untuk bahasa lain, termasuk C ++, C # , Python, dan PHP. Hal ini didasarkan sekitar Github ini Elektron, yang merupakan versi

*cross-platform* dari Atom komponen kode-editing, berdasarkan JavaScript dan *HTML5*. Visual Studio Code telah dirancang untuk bekerja dengan alat-alat yang ada, dan Microsoft menyediakan dokumentasi untuk membantu pengembang bersama, dengan bantuan untuk bekerja dengan ASP.NET 5, Node.js, dan Microsoft naskah, serta alat-alat yang dapat digunakan untuk membantu membangun dan mengelola aplikasi Node.js. Visual Studio Code benar-benar sedang ditargetkan pada pengembang JavaScript yang ingin alat pengembangannya lengkap untuk *scripting server-side* mereka dan yang mungkin ingin usaha dari Node.js untuk kerangka berbasis NET. Visual Studio Code, adalah belum solid, lintas *platform* kode Editor ringan, yang dapat digunakan oleh siapa saja untuk membangun aplikasi untuk *website*. Komponen pada Visual Studio Code berjumlah delapan. Komponen tersebut meliputi: *Customize*, *Command Palette*, *Integrated Terminal*, *Extention: Extention*, *Search*, *Grid Editor Layout*, *Color Themes*, dan *Cloud Enviroment*. Ada sembilan *extention* pada Visual studio Code meliputi: *Live server*, *GitLens*, *Prettier (Code Formatter)*, *Auto Rename Tag*, *Beautify*, *Open In Browser*, *CSS Peak*, *Indent Rainow*, dan *SQL Server*. Dengan keberadaan sejumlah komponen tersebut, maka Visual Studio Code mempunyai sejumlah kelebihan antara-lain: mudah untuk mengelola *extention*, memiliki *extention* yang banyak, kontribusi tampilan,

dukungan bahasa, *text* editor gratis, dan dapat membuat Snippet sendiri [12].



Gambar 2. 1 Tampilan Visual Code

### 2.2.5. Bahasa Pemrograman PHP

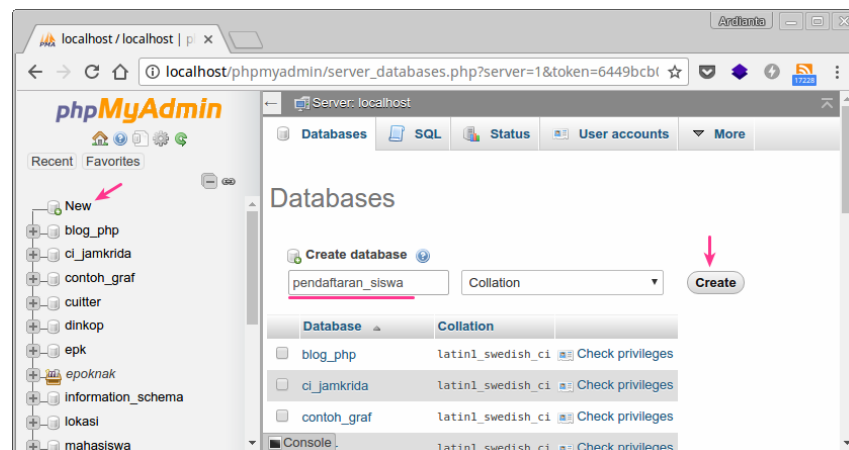
Bahasa pemrograman PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman yang bekerja dalam sebuah *web server*. *Hypertext Preprocessor* merupakan bahasa skrip yang tertanam dalam *HTML* untuk dieksekusi bersifat *server-side*. PHP termasuk dalam *open source product*, sehingga *source code* PHP dapat diubah dan didistribusikan secara bebas [13].



Gambar 2. 2 Logo Bahasa pemrograman PHP

### 2.2.6. MySQL

MySQL merupakan database yang pertama kali didukung oleh bahasa pemrograman *script* untuk internet (PHP dan Perl). MySQL dan PHP dianggap sebagai pasangan *software* pembangun aplikasi *website* yang ideal. MySQL lebih sering digunakan untuk membangun aplikasi berbasis *website*, umumnya pengembangan aplikasinya menggunakan bahasa pemrograman *script* PHP [13].



Gambar 2. 3 Tampilan Halaman Database Mysql

### 2.2.7. HTML

*HTML (HyperText Markup Language)* adalah bahasa pemformatan teks untuk dokumen-dokumen pada jaringan komputer yang sering disebut sebagai *world wide web*. *HTML* itu adalah bahasa

yang digunakan untuk menulis halaman *website*, biasanya menggunakan ekstensi *.htm*, *.HTML* atau *.SHTML* [13].

```
<!DOCTYPE html>
<html>

  <head>
    <title>My First Webpage</title>
  </head>

  <body>
    <h1>
      My First Webpage
    </h1>
    <p>This is a paragraph...</p>
  </body>

</html>
```

Gambar 2. 4 Tampilan HTML

### 2.2.8. *Bootstrap*

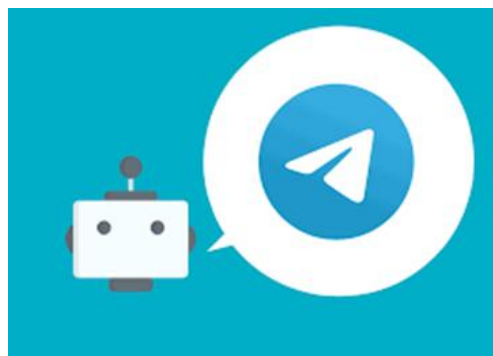
*Bootstrap* adalah paket aplikasi yang dapat digunakan saat membuat *front-end website*. Bisa dikatakan bahwa *bootstrap* adalah template desain *website* yang memiliki banyak fitur. *Bootstrap* dirancang untuk membuat desain *website* lebih mudah bagi pengguna dari tingkat pemula hingga yang lebih berpengalaman. Anda cukup memahami dasar *HTML* dan *CSS* dan siap menggunakan *bootstrap*. [14].



Gambar 2. 5 Tampilan Logo *Bootstrap*

### 2.2.9. Bot Telegram

Bot pada aplikasi *chatting* telegram merupakan *fitur third party* yang berjalan dalam telegram. Pengguna dapat berinteraksi dengan bot dengan mengirim pesan, juga bisa membalas sesuai *request* otomatis. Telegram bot juga dapat mencari *username* pengguna yang ditarget serta menemukan kontak yang saling berkaitan [15].



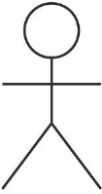
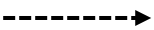

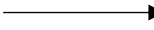
Gambar 2. 6 Bot Telegram


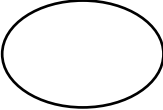
### 2.2.10. UML

*Unified Modeling Language (UML)* adalah sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk mendokumentasikan dan menyesifikasikan sistem. Pada tahun 1994, Grady Booch dan James Rumbaugh menciptakan *UML* untuk menggabungkan dua metodologi terkenal, Booch dan OMT. Ivar Jacobson, pencipta OOSE, kemudian bergabung. OMG membuat standar *UML* (Sri Mulyani NS, 2017). *UML* yang sering digunakan dalam pengembangan sebuah sistem yaitu: [16].

1. *Use Case*: Diagram dalam *UML (Unified Modeling Language)* yang menggambarkan interaksi antara aktor (pengguna) dan sistem dengan menggunakan kasus (fungsi atau fitur yang dapat digunakan oleh aktor).


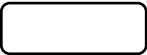
Tabel 2. 1 *Use Case Diagram*



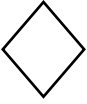
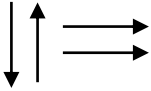
No.	Simbol	Keterangan
1.		<b>Actor</b> Menspesifikan himpauan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2.		<b>Include</b> Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
3.		<b>Extend</b> Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
4.		<b>Association</b> Apa yang menghubungkan antara objek satu dengan objek lainnya.

No.	Simbol	Keterangan
5.		<b>System</b> Menspesifikasikan paket yang menampilkan sistem secara terbatas.
6.		<b>Use Case</b> Abstraksi dan interaksi antara sistem dan <i>actor</i> .

2. *Activity Diagram*: Merupakan diagram yang bersifat statis, yang menggambarkan aktivitas dari suatu sistem bisnis. Untuk simbol dari diagram aktivitas bisa dilihat pada tabel 2.2.


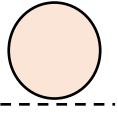
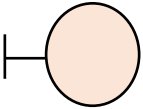
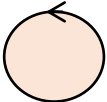
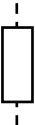

Tabel 2. 2 *Activity Diagram*

No.	Simbol	Keterangan
1.		<b>Activity</b> Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2.		<b>Action</b> <i>State</i> dari sistem yang mencerminkan eksekusi dari suatu aksi.

No.	Simbol	Keterangan
3.		<b><i>Initial Node</i></b> Bagaimana objek dibentuk atau diawali.
4.		<b><i>Activity Final Node</i></b> Bagaimana objek dibentuk dan diakhiri.
5.		<b><i>Decision</i></b> Digunakan untuk menggambarkan suatu keputusan/tindakan yang harus diambil pada kondisi tertentu.
6.		<b><i>Line Connector</i></b> Digunakan untuk menghubungkan satu simbol.

3. *Sequence Diagram*: Adalah diagram dalam *UML (Unified Modeling Language)* yang digunakan untuk menggambarkan interaksi antar objek dalam suatu sistem berdasarkan urutan waktu. Diagram ini menunjukkan bagaimana pesan dikirim antar objek untuk menyelesaikan suatu proses atau fungsi. Diagramnya secara visual menampilkan aktor di sebelah kiri, sistem di tengah, dan database jika diperlukan.

Tabel 2. 3 *Sequence Diagram*

No.	Simbol	Keterangan
1.		<b>Actor</b> Menggambar orang yang sedang berinteraksi dengan sistem.
2.		<b>Entity Class</b> Menggambarkan hubungan yang akan dilakukan.
3.		<b>Boundary Class</b> Menggambarkan sebuah gambaran dari <i>form</i>
4.		<b>Control Class</b> Menggambarkan penghubung antara <i>boundry</i> dengan tabel.
5.		<b>A focus of Control &amp; A Life Line</b> Menggambarkan tempat mulai dan berakhirnya <i>message</i> .
6.		<b>A message</b> Menggambarkan pengiriman pesan.

### 2.2.11. *Flowchart*



*Flowchart* sistem merupakan suatu urutan proses dalam *system* dengan menunjukkan alat dari media input, output serta jenis media yang digunakan untuk penyimpanan dalam proses pengolahan data sedangkan *flowchart* program merupakan suatu bagan dengan simbol-simbol tertentu yang menggambarkan suatu urutan dari proses secara detail dan berhubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program. Jika seseorang analis dan programmer yang akan membuat *flowchart*, terdapat beberapa petunjuk yang harus diperhatikan, seperti berikut:

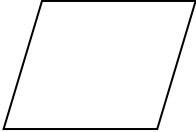

1. *Flowchart* dibuat mengikuti proses nya dari halaman atas ke bawah dan dari kiri ke kanan.
2. Aktivitas yang tergambarakan harus didefinisikan secara hati hati dan didefinisikan harus dapat dipahami oleh pembacanya.
3. Setiap aktivitas yang dimulai dan diakhiri harus ditentukan secara jelas.
4. Setiap langkah-langkah dari aktivitas harus dijabarkan dengan menggunakan deskripsi kata kerja.
5. Langkah-langkah dari setiap aktivitas harus berada pada urutan yang benar.
6. Lingkup dan range dari aktifitas yang sedang digambarkan harus ditelusuri dengan hati-hati. Percabangan-percabangan yang memotong aktivitas yang sedang digambarkan tidak perlu


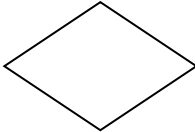
digambarkan pada *flowchart* yang sama. Simbol konektor harus digunakan dan percabangannya diletakan pada halaman yang terpisah atau hilangkan seluruhnya bila percabangannya tidak berkaitan dengan sistem.


7. Menggunakan simbol simbol *flowchart* yang standar. Tabel dibawah ini simbol *flowchart* simbol nama fungsi terminator untuk mengawali [17].

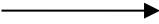
Tabel 2. 4 *FlowChart*

No.	Simbol	Keterangan
1.		<p><i>Terminator</i> / Terminal</p> <p>Merupakan simbol yang digunakan untuk menentukan <i>state</i> awal dan <i>state</i> akhir suatu <i>flowchart</i> program.</p>
2.		<p><i>Preparation</i> / Persiapan</p> <p>merupakan simbol yang digunakan untuk menunjukkan variabel yang akan digunakan oleh program. Bisa digunakan untuk memberikan harga awal variabel, yang ditunjukkan dengan nama variabel yang serupa dengan tanda "" untuk tipe string, (0) untuk tipe <i>numeric</i>, (.F./T.)</p>

No.	Simbol	Keterangan
		<p>untuk tipe <i>Boolean</i>, dan (<code>{/}</code>)</p> <p>untuk tipe tanggal.</p>
3.		<p><i>Input output</i> / Masukan keluaran</p> <p>Ini adalah simbol yang tidak memiliki operator, baik aritmatika maupun perbandingan, dan dapat digunakan untuk memasukkan nilai dan menampilkan nilai variabel.</p> <p>Jika ciri masukan adalah variabel yang belum menerima operasi dari operator tertentu, seperti pemberian nilai atau penambahan nilai, maka ciri keluaran biasanya adalah variabel yang telah menerima pemberian nilai atau operasi dengan operator tertentu. Ini membedakan masukan dan keluaran.</p>
4.		<p><i>Process</i> / Proses</p> <p>merupakan simbol yang dapat memberikan nilai tertentu. Ini dapat berupa rumus, perhitungan <i>counter</i>,</p>

No.	Simbol	Keterangan
		atau hanya memberi nilai tertentu terhadap suatu variabel.
5.		<p><i>Predefined Process</i> / Proses Terdefinisi</p> <p>Merupakan simbol yang penggunaannya seperti link atau menu. Jadi proses yang ada di dalam simbol ini harus di buatkan penjelasan <i>flowchart</i> programnya secara tersendiri yang terdiri dari terminator dan diakhiri dengan terminator.</p>
6.		<p><i>Decision</i> / simbol Keputusan digunakan untuk menentukan apakah suatu kondisi menerima atau menolak. Dibandingkan dengan simbol-simbol lain dalam diagram <i>flowchart</i> program, simbol keputusan ini harus memiliki minimal dua keluaran baris.</p> <p>Dengan kata lain, jika hanya ada satu keluaran baris, penulisan</p>

No.	Simbol	Keterangan
		<p>simbol ini adalah salah. Oleh karena itu, pengguna dapat memilih antara benar atau salah untuk kondisi. sehingga dapat ditulis jika keluaran simbol ini lebih dari dua. Ada kebutuhan untuk memberikan keterangan Ya dan Tidak pada arus keluaran yang memiliki dua <i>output</i></p>
7.		<p><i>Connector</i></p> <p>Simbol-simbol di setiap halaman dihubungkan oleh konektor. Arus yang panjang harus ditulis di Tampa. Untuk membuat gambar aliran program lebih sederhana, simbol konektornya adalah lingkaran. Di sisi lain, untuk menghubungkan antara simbol-simbol yang berbeda halaman, simbol konektor segi lima dapat digunakan, dengan identitas huruf alfabet A–Z, a–z, atau angka 1–9.</p>

No.	Simbol	Keterangan
8.		<p><i>Arrow / Arus</i></p> <p>Ini adalah simbol yang digunakan untuk menunjukkan aliran dalam <i>flowchart</i> program. Karena itu, simbol panah harus digunakan untuk menunjukkan arus data.</p>