# **BAB II**

# **TINJAUAN PUSTAKA**

## **2.1 Penelitian Terdahulu**

Referensi dalam mengerjakan penelitian ini, dapat dikaji beberapa penelitan yang telah berjalan yang bersifat terdahulu yang berhubungan dengan tema penelitian perancangan *prototype* otomasi wadah pakan berbasis mikrontroler esp32 wroom berguna untuk membantu penulis dalam penulisan ,perancangan dan pembuatan *prototype.*

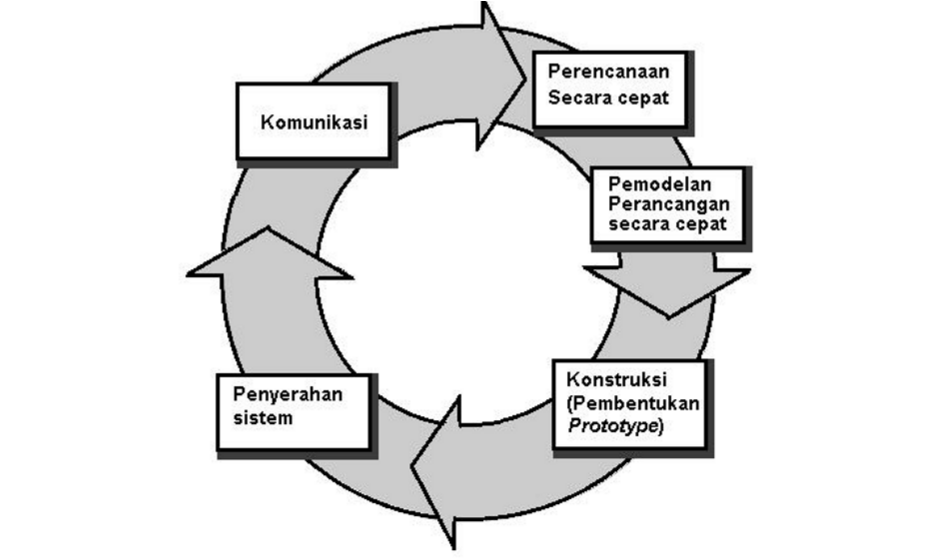
**Tabel 2.1** Penelitian Terdahulu

|  |  |  |
| --- | --- | --- |
| Nama Peneliti | Judul Penelitian | Hasil Penelitian |
| (Khair & Sabrina, 2019) | Alat Pemberi Makan Kucing Otomatis Berbasi Arduino Uno Pada Pet Shop | Alat ini dilengkapi dengan motor servo yang berjalan secara otomatis menggunakan *Real time clock* sebagai pengatur waktu, yang diperlukan untuk menjalankan servo setelah waktu yang telah ditentukan sebelumnya. Selain itu, terdapat sensor berat *(loadcell*) yang terhubung dengan modul HX711 dan memberikan informasi tentang berat benda pada sensor. Alat ini secara otomatis mengeluarkan makanan ketika waktu yang ditentukan dalam jam waktu nyata cocok dengan waktu sebenarnya dan tidak ada objek diatas sensor berat*(loadcell).* Media *timer* yang digunakan adalah jam real-time DS-1302 yang terhubung dengan Arduino Uno dan aplikasi Arduino yang terhubung dengan komputer/laptop. |
| (Slamet Efendy, 2021) | Rancang Bangun Alat Pemberi Pakan Ikan Berbasis Arduino Uno Dengan Menggunakan Komunikasi | Arduino dapat memberikan perintah yang benar ke Motor Servo, *loadcell* dan RTC sesuai dengan pengkodean yang telah dituliskan. Pada percobaan pertama, dengan reservoir penuh dan 2g makanan yang sisa, 53 g makanan dapat dikeluarkan pada jam pertama dan 52g makanan pada jam kedua. Pada percobaan kedua, 50g makanan dikonsumsi pada jam pertama dan kedua, menghabiskan seluruh isi tangki penyimpangan dan sisa makanan di tangki utama melebihi jumlah yang ditentukan, makanan yang keluar akan melebihi berat yang ditentukan. Namun, jika kandungan pakan ditangki utama rendah, pakan yang dikeluarkan akan lebih akurat. Kita dapat menyimpulkan bahwa alat yang dieksekusi menghasilkan presentase 99% dari tes yang dieksekusi. Lebih banyak makanan dilepaskan ketika tangka utama diisi dengan makanan. Namun, saat umpan sudah penuh, umpan yang keluar akan lebih akurat. |
| (Abdi et al., 2018)*.* | Gateway Pada Otomatisasi Kandang Kucing Dan Kelinci(*Hardware* Elektronik dan Pemrograman Sistem Komunikasi dengan Handphone) | Start kerja dari sistem adalah pemilihan mode kandang untuk kelinci atau kucing dan untuk memilih salah satu mode tersebut telah tersedia push button yang mewakili masing- masing perintah seperti mode kelinci, mode kucing, disamping menu utama. Dalam kerja sistem keseluruhan membutuhkan pewaktuan dikarenakan tiap alat bekerja berdasarkan jangka waktu yang telah ditetapkan. Untuk mendapatkan pewaktuan ini mikrokontroler AVR AT MEGA 32 mengambil data RTC atau *Real Time Clock.* |

## **2.2 Teori Terkait**

### **2.2.1 Model Prototype**

Tahap pengembangan untuk membuat sistem baru dalam penelitian ini menggunakan pendekatan prototype sebagai metode pengembangannya. Prototype didefinisikan sebagai metode pengembangan perangkat lunak yang menggunakan pendekatan stakeholder dalam proses produksi, mengikuti lima langkah pendekatan prototyping (Saifulloh, 2021).



**Gambar 2.1** Metode *Prototype*

Berdasarkan skema pada gambar 2.2, model prototype memiliki lima tahap pengembangan sistem. Berikut tahapannya:

1. Komunikasi

Observasi yaitu berdiskusi dengan para *stakeholder* untuk menentukan kebutuhan sistem yang akan digunakan nanti pada perancang sistem.

1. Perencanaan Secara Cepat

Prototype system sementara kemudian melakukan pemodelan sistem sebagai skema sistem terstruktur.

1. Model Rancangan Cepat

Untuk menggambarkan model rancangan, penulis menggunakan *tools* UML khususnya *usecase* untuk nantinya mendefinisikan fungsionalitas sistem, class diagram untuk menggambarkan *database* pada sistem, dan *activity* diagram untuk menggambarkan alur proses dari sistem.

1. Pembuatan Prototype

Prototyping menyajikan hasil desain antarmuka pengguna atau tata letak sistem.

1. Deployment Delivery & Feedback

Menyerahkan prototipe kepada pemngku kepentigan untuk mengevaluasi sistem dan memberikan umpan balik untuk perbaikan meningkatkan spesifikasi sistem sehingga pengembang dapat melakukan perbaikan prototipe berdasarkan umpan balik pemangku kepentingan.

### **2.2.2 Metode Pengujian Black Box**

Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black box* memungkinkan perekayasa perangkat untuk lunak mendapatkan serangkaian kondisi input yang sepenuhnya memanfaatkan semua persyaratan fungsional untuk suatu program (Septilia & Styawati, 2020).Pengujian black box berusaha mencoba menemukan kesalahan dalam kategori sebagai berikut :

1. Fungsi – fungsi salah atau hilang,
2. Kesalahan interface
3. Kesalahan dalam struktur data atau akses eksternal
4. Kesalahan kinerja
5. Inisialisasi dan kesalahan terminasi

**Tabel 2.2** Kelebihan dan Kelemahan *Blackbox Testing*

|  |  |
| --- | --- |
| Kelebihan | Kelemahan |
| Perincian aplikasi dapat ditentukan di awal, dan pengujian dilakukan berdasarkan perincian spesifikasi tersebut. | Apabila keperluan perangkat lunak yang akan dikembangkan tidak begitu jelas, pembuatan dokumentasi yan tepat akan sedikit sulit. |
| Dapat dipakai untuk menilai konsistensi suatu aplikasi, dan tidak perlu melihat kode program secara detail. | Penggunaan akan kurang merasa yakin dengan perangkat lunak yang standar pengujian. |

### **2.2.3 Internet of Things (IoT)**

*Internet of Things* mengacu pada penggunaan perangkat dan sistem yang terhubung secara cerdas untuk memanfaatkan data yang dikumpulkan dari sensor dan aktuator yang tertanam dalam mesin dan objek lainnya. *Internet of Things* diperkirakan akan menyebar dengan cepat dalam beberapa tahun mendatang dan integrasi ini akan mengeluarkan ruang layanan yang baru, sehingga meningkatkan kualitas hidup konsumen dan produktifitas seperti pada layanan dasar di bidang pendidikan, kesehatan, dan sektor lainnya serta menyediakan ekosistem baru untuk pengembangan aplikasi (GSMA *Association*, 2014). Menurut Burange dan Misalkar (Junaidi, 2015) “*Internet of Things* adalah Suatu struktur dimana objek atau orang diberikan identitas eksklusif dan kemampuan untuk memindahkan data dalam suatu jaringan tanpa memerlukan interaksi yaitu sumber ke tujuan atau manusia ke komputer”. Menurut Keoh dkk (Junaidi, 2015) “*Internet of Things* merupakan perkembangan ilmiah yang menjanjikan untuk optimalisasi kehidupan berdasarkan sensor dan perangkat cerdas yang terhubung dan berkomunikasi melalui Internet”.

### **2.2.4 Jaringan *Wireless***

Jaringan *wireless* ialah sebuah sistem komunikasi data yang fleksibel dengan menggunakan gelombang elektromagnetik tanpa mengandalkan koneksi fisik atau kabel (Sharma, 2015). Adapun beberapa penerapan dari jaringan wireless sebagai berikut:

1. Akses Internet

Jaringan *wireless* menjadi begitu luas sehingga dapat mengakses internet hampir dimana saja dan kapan saja tanpa menggunakan kabel.

1. Jaringan Publik

Meningkatnya *“hostpot”* publik membuat jaringan *wireless* sangat berguna.

1. *Location-based Service*

*Location-based Service* atau layanan berbasis lokasi ialah tingkat program komputer dengan layanan umum yang digunakan untuk mengontrol lokasi dan waktu khusus. LBS mencakup layanan untuk mengidentifikasi orang atau suatu objek seperti menemukan ATM terdekat.

### **2.2.5 Arduino IDE**

Arduino IDE merupakan kependekan dari *Integrated Development Environment software* untuk melakukan penulisan skrip code, dan mengeksekusi hasil skrip code, upload program ke *board* arduino. Program yang ditulis menggunakan Arduino IDE disebut *sketch.* Sketch ini ditulis di text editor dan disimpan dengan ekstensi file .ino. Konsol yang terdapat pada Arduino IDE memberikan *feedback* saat menyimpan, mengekspor, dan juga menampilkan pesan lengkap dengan informasi lainnya (arduino.cc).

**Gambar 2.2** Logo Arduino

### **2.2.6 BLYNK**

BLYNK adalah *platform* untuk aplikasi Sistem Operasi Seluler (iOS dan Android) yang ditujukan untuk mengendalikan *Arduino, Raspberry Pi, ESP8266, WEMOS D1*, dan modul serupa melalui Internet. Aplikasi ini adalah tempat kreatif untuk membuat GUI untuk proyek yang hanya akan dilakukan dengan *widget drag* dan *drop. S*angat mudah untuk digunakan untuk mengatur semuanya dan dapat dilakukan dalam waktu kurang dari 5 menit. Blynk tidak terikat pada *tag* atau modul tertentu. Dari *platform* aplikasi ini, Anda dapat mengontrol apapun dari jarak jauh, kapan saja, dimana saja. Dengan catatan terhubung ke internet dengan jalur stabil dan disebut sistem *Internet of Things* (Blynk.cc).

**Gambar 2.3** Logo Blynk

### **2.2.7 ESP32 WROOM**

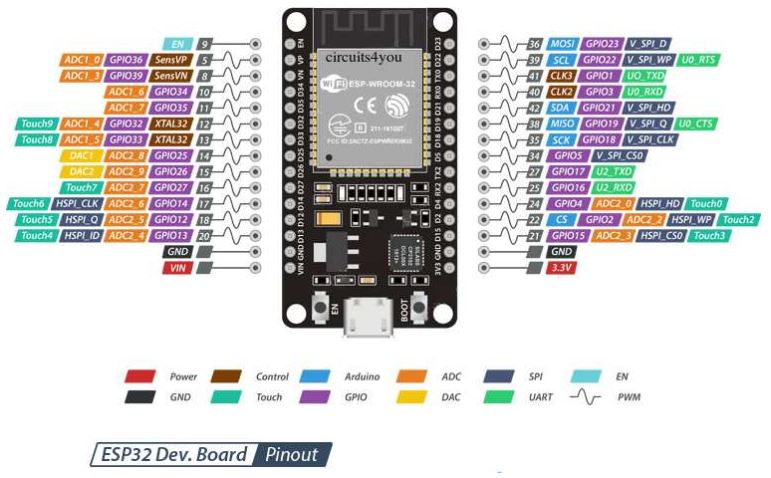
Esp32 Wroom 32D adalah rangkaian *microkontroller* berbiaya rendah dan berdaya rendah pada chip mikrontroler dengan Wi-Fi internal dan Bluetooth dual-mode. Seri Esp32 menggunakan mikroprosesor Tensilica Xtensa LX6 dalam varian dual-core dan single-core dan mencakup sakelar antena internal, balun RF, penguat daya, penguat penerima kebisingan rendah, filter, dan modul manajemen daya. ESP32 diproduksi dan dikembangkan oleh perusahaan China Espresif Systems yang berbasi di Shanghai dan diproduksi oleh TMSC pada proses 0nm. Mikrontoller ini adalah penerus mikrontorler ESP8266.

**Tabel 2.3** Spesifikasi Mikrontroller Esp32 Wroom

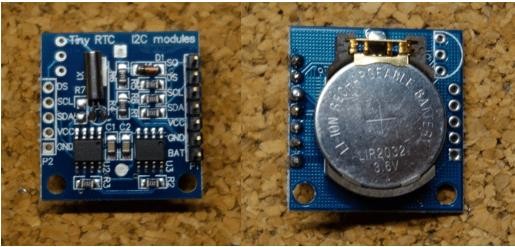
|  |  |
| --- | --- |
| Prosesor: | Konektivitas nirkabel: |
| CPU: Xtensa dual-core (atau single-core) 32-bit LX6 mikroprosesor, beroperasi pada 160 atau 240 MHz dan bekerja hingga 600 DMIPS | *Wi-Fi*: 802.11 b / g / n |
| Ultra low power (ULP) co-prosessor | *Bluetooth*: v4.2 BR / EDR dan BLE (berbagi radio dengan *Wi-Fi*) |
| Memori: 520 KiB SRAM |  |

|  |  |
| --- | --- |
| Antarmuka Periferal | Keamanan |
| * 12-bit SAR ADC hingga 18 *channel* * DAC 2 × 8-bit * 10 × sensor sentuh (GPIO penginderaan kapasitif) * 4 × SPI * Antarmuka 2 × I2S * Antarmuka 2 × I2C * 3 × UART * Pengontrol *host* SD / SDIO / CE-ATA / MMC / eMMC * Pengontrol *slave* SDIO / SPI * Antarmuka MAC *Ethernet* dengan dukungan khusus DMA dan IEEE 1588 *Precision Time Protocol* * CAN bus 2.0 * Pengontrol jarak jauh inframerah (TX / RX, hingga 8 saluran) * PWM motor * LED PWM (hingga 16 channel) * Sensor *hall effect* * *Pre-amplifier* analog daya sangat rendah | Semua fitur keamanan standar IEEE 802.11 didukung, termasuk WFA, WPA / WPA2 dan WAPI |
| Akselerasi perangkat keras kriptografi: AES, SHA-2, RSA, kriptografi kurva eliptik (ECC), generator bilangan acak (RNG) |
| OTP 1024-bit, hingga 768-bit untuk pelanggan |
| *Safe Booting* |
| Enkripsi *flash* |

|  |
| --- |
| Manajemen Daya |
| *Regulator break school* internal yang rendah |
| Domain daya individu untuk RTC |
| 5μA *sleep well* saat ini |
| Bangun dari interupsi GPIO, pengatur waktu, pengukuran ADC, interupsi sensor sentuh kapasitif |

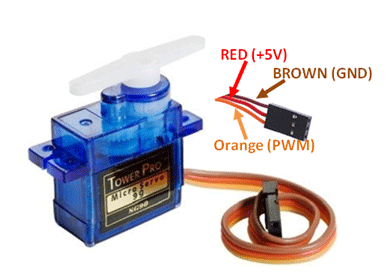
**Gambar 2.4** Mikrontroller ESP32

### **2.2.8 Real Time Clock (RTC) Modul DS3231**

Komponen jam real time adalah komponen IC counter yang dapat berperan sebagai sumber data waktu berupa data jam, hari, bulan atau tahun. Komponen DS1307 merupakan sirkuit terpadu yang harus dilengkapi dengan komponen pendukung lainnya seperti kristal sebagai sumber clock dan baterai eksternal 3,6 volt sebagai sumber daya cadangan agar fungsi counter tidak berhenti bekerja. Bentuk komunikasi data dari IC RTC adalah I2C, yang merupakan singkatan dari Inter Integrated Circuit. Jenis komunikasi ini hanya menggunakan dua jalur komunikasi, SCL dan SDA. Semua mikrokontroler dilengkapi dengan fungsi komunikasi dua arah ini, termasuk mikrokontroler Arduino.

**Gambar 2.5** Modul RTC DS3231

### **2.2.9 Servo Motor Tower Pro SG90**

Motor Servo adalah Akuator elektromagnetis yang tidak berputar secara kontinu seperti [*motor DC*](https://www.edukasielektronika.com/2018/05/motor-dc-direct-current.html) atau motor stepper. Motor servo digunakan untuk posisi dan memegang beberapa objek. Motor jenis ini digunakan dimana rotasi kontinu tidak diperlukan sehingga tidak digunakan untuk mengendalikan roda (kecuali servo ini dimodifikasi). Sebaliknya, motor servo digunakan dimana sesuatu yang dibutuhkan pindah ke posisi tertentu dan kemudian berhenti dan bertahan pada posisi itu.

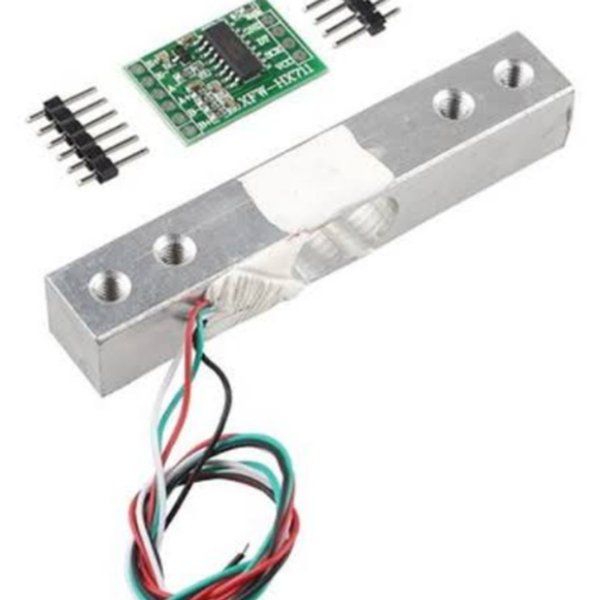
**Gambar 2.6** Servo Motor

### **2.2.10 Modul ESP32 Cam**

ESP32-CAM adalah papan pengembangan berbiaya rendah berdasarkan ESP32 dengan kamera onboard, ukuran kecil. Ini adalah solusi ideal untuk aplikasi IoT, membangun prototipe, dan proyek DIY. Papan ini mengintegrasikan WiFi, Bluetooth tradisional, dan BLE berdaya rendah, dengan 2 prosesor LX6 32-bit berkinerja tinggi. Ini mengadopsi arsitektur pipa 7-tahap, sensor pada chip, sensor Hall, sensor suhu dan sebagainya. dan penyetelan frekuensi utamanya berkisar dari 80MHz hingga 2 0MHz. Sepenuhnya sesuai dengan standar WiFi 802.11b/g/n/e/i dan Bluetooth .2, ini dapat digunakan sebagai mode master untuk membangun pengontrol jaringan mandiri atau sebagai penyedia layanan untuk MCU host lain untuk menambahkan kemampuan jaringan ke perangkat yang ada.

**Gambar 2.7** ESP32 Cam Modul

### **2.2.11 Sensor Berat dan Modul Amplifier HX711**

 Sensor berat atau sensor gaya adalah transduser yang digunakan untuk mengubah tekanan menjadi sinyal listrik. Sensor berat biasanya menyertakan pengukur regangan dalam konfigurasi jembatan Wheatstone, tetapi beberapa masih menyertakan 2 pengukur regangan. Keluarannya berupa sinyal listrik dalam satuan milivolt dan memerlukan penguat berupa penguat sebelum dapat digunakan (Handajadi dan Soleh 2009), modul HX711 merupakan modul yang membantu kita untuk membaca sensor.variable force lebih mudah dalam mengukur bobot. Modul ini digunakan untuk memperkuat sinyal keluaran sensor dan mengubah data analog menjadi data digital dan dihubungkan dengan mikrokontroler sehingga kita dapat membaca perubahan tahanan gaya sensor. Setelah proses kalibrasi, kami mendapatkan hasil penimbangan dengan akurasi tinggi (Limantara, Chandra dan Mudjanarko 2017). Modul HX711 dan sensor berat dapat dilihat pada gambar 2.7.

**Gambar 2.8** Modul *Amplifier* HX711 dan Sensor Berat (Loadcell)

### **2.2.12 Pompa air Celup Mikro DC 2.5 – 6V USB Model**

Pompa air submersible adalah jenis pompa yang menggunakan motor DC dan tegangan DC sebagai sumber tenaganya. Dengan memberikan perbedaan tegangan antara kedua ujungnya maka motor akan berputar satu arah, dan jika polaritas tegangan dibalik maka arah putaran motor juga akan terbalik. Kutub tegangan yang diterapkan pada kedua ujung menentukan arah putaran motor, dan besarnya tegangan pada kedua ujung menentukan kecepatan motor. Pompa submersible memiliki 3 bagian dasar:

* + 1. Bagian tetap/tetap disebut stator. Stator ini menghasilkan medan magnet, yang dihasilkan oleh kumparan (elektromagnetik) atau oleh magnet permanen.
    2. Bagian yang berputar disebut rotor. Rotor ini berbentuk kumparan, dengan arus yang mengalir melaluinya.
    3. Gearbox dipasang pada pompa. Peredam ini di mana ada roda gigi yang dipasang di ujung rotor untuk menarik air. Gaya elektromagnetik pada motor DC terjadi ketika arus mengalir melalui konduktor yang ditempatkan dalam medan magnet. Medan magnet itu sendiri dihasilkan oleh magnet permanen. Garis-garis medan magnet berjalan di antara dua kutub magnet dengan arah dari kutub utara ke kutub selatan.

**Gambar 2.9** Pompa air celup mikro DC

### **2.2.13 Wadah Pakan dan Minum Kucing**

Wadah Pakan Kucing atau *Food Bowl* disediakan sebanyak 3 buah untuk 3 ekor kucing yang nantinya akan menampung jatuh nya dari cereal dispenser makanan kucing berjenis *dryfood*, Wadah minum kucing berfungsi sebagai penampung atau tempat tadahan air yang mengalir dari pompa air agar tidak mengenang dan membasahi lantai.

**Gambar 2.10** Wadah Pakan dan Minum

### **2.2.14 Relay Single Channel**

5V 1-Channel Relay *interface board*. Dapat digunakan untuk mengendalikan berbagai perangkat listrik dengan arus yang besar. Dikendalikan langsung oleh mikrokontroler pilihan Anda (Arduino , 8051, AVR, PIC, DSP, ARM, ARM, MSP430, TTL logic.

**Gambar 2.11** Relay 1 Channel

### **2.2.15 Modul External Power Supply MB102**

Modul eksternal tambahan untuk meningkatkankan performa daya yang berkerja sebagai PSU*(Power Supply Unit)* mendukung mikrontroller yang memiliki daya dasar dibawah 5v, komponen ini banyak ditemukan di *marketplace*. Modul ini dapat mengkonversi daya listrik dari 7 – 12v agar tidak terjadi *overheat* pada mikrontroler dilengkapi dengan 4 buah GND pin out, 2 buah pin out 3.3v dan 2 buah pin out 5v, untuk menghubungkan kelistrik dilengkapi dengan 1 port USB, 1 port Adapater Female, dan tombol daya dengan indicator.

**Gambar 2.12** External Power Supply

### **2.2.16 Flowchart**

Merupakan sebuah bagan alir yang berbentuk diagram atau grafik dan digunakan untuk menggambarkan urutan-urutan kegiatan dari sistem. Flowchart ini menunjukkan alur proses dalam sistem, yang memiliki beberapa alat media *input, process, output,* serta jenis media penyimpanan dalam proses pengolahan data (Sutabri, 2012).

**Tabel 2.4** Simbol *Flowchart*

|  |  |
| --- | --- |
| Simbol | Keterangan |
|  | Termintator, fungsi sebagai awal(mulai) atau akhir program |
|  | Flow, fungsi sebagai menunjukan arus dari proses atau arah aliran program |
|  | Input/Ouput, fungsi sebagai input atau output data |
|  | Proses, fungsi sebagai proses pengolahan data |
|  | Decision, fungsi sebagai kondisi untuk keputusan dengan seleksi yaitu tidak atau ya |
|  | On Page atau connector, fungsi sebagai penghubung atau penyambung bagian *flowchart* dalam lembar halaman yang sama |

### **2.2.17 Diagram Blok**

Diagram blok adalah suatu pernyataan gambar yang ringkas, dari gabungan sebab dan akibat antara masukkan dan keluaran dari suatu system.

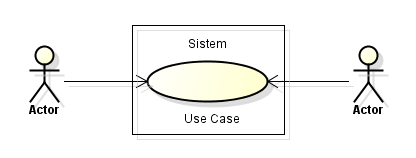
**Tabel 2.5** Simbol Diagram Blok

|  |  |
| --- | --- |
| Gambar | Keterangan |
|  | Blok/Kotak: Biasanya berisikan uraian dan nama elemennya, atau simbul untuk operasi matematis yang harus dilakukan pada masukkan untuk menghasilkan Keluaran. |
|  | Tanda anak panah : Menyatakan arah informasi aliran isyarat atau unilateral. Sebagai contoh sederhana diperlihatkan sbb: |

### **2.2.19 Use Case Diagram**

Use Case merupakan suatu gambaran fungsional dari sebuah sistem dari sudut pandang pengguna. Cara kerja use case dengam menggambarkan interaksi atara user / atau pengguna dengan sistem secara global melalui sebuah skenario bagaimana sistem tersebut berjalan (Grizelda & Septiani, 2020). Use case diagram menjelaskan bagaiman interaksi antara admin, alumni, dan mitra dengan pemberian sebuah gambaran bagaimana sistem tersebut berjalan sesuai dengan kebutuhan aktor. Use case diagram memberi gambaran singkat bagaimana relasi antara aktor, dan sistem.

Dalam diagram use case terdapat 3 aspek yaitu : *actor*, *use case*, dan *system boundary*. Bagan dari diagram use case dapat dilihat pada gambar di bawah ini :



**Gambar 2.13** Bagan Usecase

Simbol-simbol yang digunakan pada *usecase diagram* dapat dilihat pada tabel 2.8.1 dibawah ini :

**Tabel 2.6** Simbol pada *Usecase Diagram*

| Gambar | Keterangan |
| --- | --- |
|  | Use case menggambarkan fungsionalitas yang disediakan oleh sistem sebagai unit pertukaran pesan antara unit dan aktor, biasanya diekspresikan dengan kata kerja di awal nama use case. |
|  | Aktor adalah abstraksi dari orang atau sistem lain yang mengaktifkan fungsi sistem target. Untuk mengidentifikasi aktor, perlu untuk mendefinisikan pembagian kerja dan tugas yang terkait dengan peran dalam konteks sistem target. Seseorang atau sistem dapat muncul dalam berbagai peran. Perhatikan bahwa aktorberinteraksi dengan use case, tetapi tidak memiliki kendali atas use case.. |
|  | Asosiasi antara aktor dan *usecase*, diwakili oleh garis tanpa panah yang menunjukkan siapa atau apa yang meminta interaksi secara langsung daripada menunjukkan aliran data. |
|  | Asosiasi antara aktor dan use case menggunakan panah terbuka untuk menunjukkan kapan agen berinteraksi secara pasif dengan sistem. |
|  | Include, merupakan di dalam *usecase* lain *(required)* atau *usecase* yang dipanggil oleh *usecase* lain, sebuah *instance* yang meanggil fungsi program. |
|  | *Extend,* merupakan perpanjangan dari use case lain jika kondisi atau kondisi terpenuhi. |
|  | *Package,* merupakan bungusan satu atau lebih node. |

### **2.2.20 Sequence Diagram**

*Sequence diagram* adalah bagian dari UML yang mendeskripsikan hubungan antar objek yang ada di dalam sistem, termasuk pengguna, yang disampaikan melaui pesan yang akan diterima antar objek-objek yang ada dalam sistem yang akan menghasilkan output tertentu (Wahyudi, 2020). Simbol-simbol yang akan digunakan untuk membuat Squence Diagram dapat dilihat pada tabel dibawah ini :

**Tabel 2.7** Simbol-Simbol *Sequence Diagram*

| No | Simbol | Keterangan |
| --- | --- | --- |
| 1. | *Actor* | *Actor dapat berkomunikasi dengan object, maka actor juga dapat diurutkan sebagai kolom.* |
| 2. | *Lifeline* | Lifeline mengindikasikan keberadaan sebuah object dalam basis waktu. Notasi untuk Lifeline adalah garis putus-putus vertikal yang ditarik dari sebuah objek. |
| 3. | *Activation* | Activation dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah lifeline. Activation mengindikasikan sebuah objek yang akan melakukan sebuah aksi. |
| 4. | *Boundary* | Boundary terletak di antara sistem dengan dunia sekelilingnya. Semua form, laporan-laporan, antar muka ke perangkat keras seperti printer atau scanner dan antar muka ke sistem lainnya adalah termasuk dalam kategori. |
| 5. | *Control* | Control berhubungan dengan fungsionalitas seperti pemanfaatan sumber daya, pemrosesan terdistribusi, atau penanganan kesalahan. |
| 6. | *Entity* | Elemen fisik yang eksis saat Entity digunakan menangani informasi yang mungkin akan disimpan secara permanen. Entity bisa juga merupakan sebuah tabel pada struktur basis data. |
| 7. | *Message* | Message, di gambarkan dengan anak panah horizontal antara Activation. Message mengindikasikan komunikasi antara object-object. |