

BAB II

LANDASAN TEORI

Pada bab ini menjelaskan mengenai teori – teori yang digunakan untuk memperkaya proses penelitian. Teori yang digunakan tersebut diantaranya:

2.1.Pemberian Pakan Kucing

Memberi pakan kucing bukan hal yang rumit dilakukan oleh pemilik. Hal ini bergantung pada beberapa variabel seperti berat dan usia kucing, jenis makanan yang diberikan dan aktivitas kucing. Merek pakan yang diberikan ke kucing akan membawa perbedaan, contohnya *dry food* berkualitas tinggi akan mengandung lebih banyak nutrisi dibandingkan dengan *dry food* berkualitas rendah, sehingga untuk nutrisi yang sama dibutuhkan porsi yang lebih sedikit.

Banyak pemilik kucing yang memberikan akses bebas pada kucing dengan mengisi penuh wadah tempat pakannya. Cara ini dimaksudkan agar saat kucing lapar, pakan sudah tersedia di tempatnya. Namun apabila akan menggunakan metode tersebut, pemilik harus berhati-hati karena dapat menyebabkan kegemukan/obesitas pada kucing terlebih lagi jika tidak diberikan minum yang seimbang.

Pemberian makan kucing 2 kali sehari berfungsi untuk mengurangi rasa lapar diantara waktu makan dan meminimalisir masalah perilaku terkait makanan. Pemberian pakan sebaiknya pada waktu yang tepat yaitu, pagi hari pukul 08.00 dan sore hari pukul 17.00. Jadwal pemberian makan 1 kali per hari dianggap tidak baik karena tidak memenuhi kebutuhan energi kucing per hari sedangkan pemberian makan 3 kali per hari dianggap kurang baik. *American Society Prevention of Cruelty Animals* (ASPCA) menerangkan bahwa pemberian makan satu kali sehari diperbolehkan dengan metode pemberian *ad libitum*, tetapi hal demikian sangat tidak disarankan. Kucing yang dikandangan memiliki kebutuhan energi 240 kkal per hari setara dengan setengah cup. Dengan aktivitas yang sedikit kucing yang diberikan makan 3 kali sehari sangat berisiko obesitas. Oleh sebab itu, kucing yang mulai beranjak dewasa harus dibiasakan agar

memiliki pola makan rutin dengan kandungan nutrisi yang tercukupi untuk menghasilkan kondisi tubuh yang optimal (Hartuti, Mulyadi, & Murtina, 2013).

2.2.Kucing Sakit

Mengenali kesehatan pada kucing sangat penting dilakukan oleh pemilik. Oleh karena itu memahami tanda-tanda kucing yang tidak sehat merupakan tindakan penting. Berikut beberapa gejala kucing sakit: (Vadreas & Welly, 2018)

1. Kucing terlihat berdiam diri, menyendiri, lesu, malas, acuh tak acuh ketika diberi mainan atau boneka dan terus tiduran.
2. Makannya sedikit atau tidak mau makan, sulit mengunyah makanan dan perutnya buncit, tetapi tubuhnya kurus.
3. Kehausan, terus minum atau tidak minum sama sekali.
4. Pusing dan gelisah.
5. Sibuk menjilati tubuh dan bulunya seharian. Tekstur pada bulu tubuhnya berubah, kadang kering, kadang lengket, kusam, atau kasar. Bulunya berdiri kaku, rontok dan kulitnya berketombe. Ada bagian tubuh yang botak.
6. Muntah, terkadang muntahannya berisi gumpalan rambut atau darah.
7. Bersin-bersin,
8. Batuk-batuk, hidungnya kering.
9. Senang menggaruk-garuk tubuhnya, terutama dibagian kepala dan telinga.
10. Air liurnya keluar berlebihan (*drolling*).

2.3.Kecerdasan Buatan

Kecerdasan buatan berasal dari bahasa Inggris “*Artificial Intelligence*” atau singkatan AI, yaitu *intelligence* adalah kata sifat yang berarti cerdas, sedangkan *artificial* artinya buatan. Kecerdasan buatan yang dimaksud di sini merujuk pada mesin yang mampu berpikir, menimbang tindakan yang akan diambil dan mampu mengambil keputusan seperti yang dilakukan oleh manusia (Sutojo, dkk, 2011). Beberapa macam bidang yang menggunakan kecerdasan buatan antara lain sistem pakar, permainan komputer (*games*), logika *fuzzy*, jaringan syaraf tiruan dan robotika. Pada penelitian ini, kecerdasan buatan dimanfaatkan untuk sistem pemantauan dan pemberian pakan kucing secara tepat (sesuai kebutuhan pakan

kucing) dengan menggunakan metode *fuzzy logic sugeno* sehingga pemilik kucing tidak perlu khawatir apabila sedang tidak berada di rumah atau akan meninggalkan kucing dengan waktu yang cukup lama karena sistem akan memberikan pakan kucing secara terjadwal dan pemberian jumlah pakan kucing akan disesuaikan dengan kebutuhan serta dapat mengetahui kondisi kesehatan kucing.

2.4.Fuzzy Logic

Logika *fuzzy* (*fuzzy logic*) merupakan cabang dari sistem kecerdasan buatan (*Artificial Intelligent*) yang mengemulasi kemampuan manusia dalam berpikir ke dalam bentuk algoritma yang kemudian dijalankan oleh mesin. Logika *fuzzy* diperkenalkan pertama kali pada tahun 1965 oleh Prof Lutfi A. Zadeh seorang peneliti di Universitas California di Berkley dalam bidang ilmu komputer. Professor Zadeh beranggapan bahwa logika tegas (benar dan salah) tidak dapat mewakili setiap pemikiran manusia, oleh karena itu dikembangkanlah logika *fuzzy* yang dapat mempresentasikan setiap keadaan atau mewakili setiap pemikiran manusia. Perbedaan antara logika tegas dan logika *fuzzy* terletak pada keanggotaan elemen dalam suatu himpunan. Jika dalam logika tegas suatu elemen mempunyai dua pilihan yaitu terdapat dalam himpunan atau bernilai 1 yang berarti benar dan tidak pada himpunan atau bernilai 0 yang berarti salah. Sedangkan dalam logika *fuzzy*, keanggotaan elemen berada di interval $[0,1]$ (Trimartanti, 2016).

Beberapa hal yang perlu diketahui dalam memahami sistem *fuzzy* yaitu: (Mujab, 2018)

- Variabel *fuzzy*, merupakan variabel yang hendak dibahas dalam suatu sistem *fuzzy*, seperti: umur, temperatur, permintaan, dan sebagainya.
- Himpunan *fuzzy*, merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel *fuzzy*. Sebagai contoh, himpunan *fuzzy* adalah variabel laju kendaraan yang terbagi menjadi tiga himpunan *fuzzy*, yaitu: MUDA, PAROBAYA, TUA.
- Semesta pembicaraan, merupakan keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel *fuzzy*. Semesta pembicaraan merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai semesta pembicaraan dapat berupa bilangan positif maupun

negatif. Adakalanya nilai semesta pembicaraan ini tidak dibatasi batas atasnya. Sebagai contoh, semesta pembicaraan untuk variabel laju kendaraan adalah $[0,100]$.

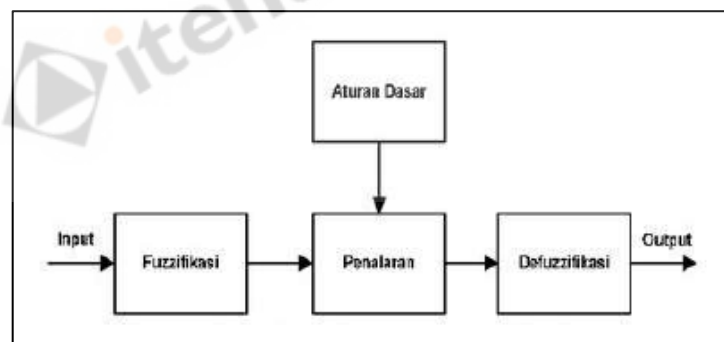
- Domain, merupakan keseluruhan nilai yang diizinkan dalam semesta pembicaraan (himpunan bilangan real yang bertambah secara monoton dari kiri ke kanan) dan boleh dioperasikan dalam suatu himpunan *fuzzy*. Sebagai contoh, domain dari himpunan *fuzzy* kecepatan adalah sebagai berikut:

MUDA = $[0, 40]$, artinya seseorang dikatakan muda dengan umur 0 hingga 40,

PAROBAYA = $[30,50]$, artinya seseorang dikatakan parobaya dengan umur 30 hingga 50,

CEPAT = $[40, +\infty]$, artinya seseorang dikatakan tua dengan umur 40 hingga $+\infty$.

Pada Gambar 3. merupakan blok diagram dari sistem logika *fuzzy*. Dalam sistem logika *fuzzy*, terdapat beberapa tahapan diantaranya: (Mahargiyak, Anggraeni, Wandiro, & Mahzar, 2013).



Gambar 3. Blok diagram sistem logika *fuzzy*

1. Fuzzifikasi

Fuzzifikasi yaitu suatu proses perubahan nilai tegas yang ada ke dalam fungsi keanggotaan.

2. Penalaran (*Inference Machine*)

Mesin penalaran adalah proses implikasi dalam menalar nilai masukan guna penentuan nilai keluaran sebagai bentuk pengambilan keputusan.

3. Aturan Dasar (*Rule Based*)

Aturan dasar (*rule based*) pada kontrol logika *fuzzy* merupakan suatu bentuk aturan relasi “JIKA - MAKA” atau “IF - THEN” seperti berikut ini:

IF x is A THEN y is B, dimana A dan B adalah *linguistic values* yang didefinisikan dalam rentang variabel X dan Y. Pernyataan “ x is A” disebut anteseden atau premis, pernyataan “ y is B” disebut konsekuen atau kesimpulan.

4. Defuzzifikasi

Input dari proses defuzzifikasi adalah suatu himpunan *fuzzy* yang diperoleh dari komposisi aturan-aturan fuzzy, sedangkan *output* yang dihasilkan merupakan suatu bilangan pada domain himpunan *fuzzy* tersebut. Sehingga jika diberikan suatu himpunan *fuzzy* dalam *range* tertentu, maka harus dapat diambil suatu nilai crisp tertentu.

Dalam proses pemanfaatan logika *fuzzy*, ada beberapa hal yang perlu diperhatikan salah satunya yaitu cara mengolah *input* menjadi *output* melalui sistem inferensi *fuzzy*. Proses ini melibatkan fungsi keanggotaan, operasi logika, serta aturan *IF-THEN*. Hasil dari proses ini akan menghasilkan sebuah sistem yang disebut dengan FIS (*Fuzzy Inference System*). Dalam logika *fuzzy* tersedia beberapa jenis FIS diantaranya adalah *Mamdani*, *Sugeno* dan *Tsukamoto*.

FIS (*Fuzzy Inference System*) yang digunakan pada sistem yang dibangun yaitu *fuzzy logic sugeno*. *Fuzzy logic sugeno* merupakan metode inferensi *fuzzy* untuk aturan yang direpresentasikan dalam bentuk IF – THEN, dimana *output* (konsekuen) sistem tidak berupa himpunan *fuzzy*, melainkan berupa konstanta atau persamaan linear. Metode ini diperkenalkan oleh Takagi-Sugeno Kang pada tahun 1985. *Fuzzy logic Sugeno* menggunakan fungsi keanggotaan *Singleton* yaitu fungsi keanggotaan yang memiliki derajat keanggotaan 1 pada suatu nilai crisp tunggal dan 0 pada nilai crisp yang lain. Terdapat 2 model pada *fuzzy logic sugeno* yaitu:

- Model *Fuzzy Sugeno Orde-Nol*

Secara umum bentuk model *Fuzzy Sugeno Orde Nol* adalah IF (x_1 is A_1) \circ (x_2 is A_2) \circ (x_3 is A_3) \circ ... \circ (x_N is A_N) THEN $z = k$, dengan A_i adalah himpunan *fuzzy* ke- i sebagai anteseden (premis), \circ adalah operator *fuzzy* (AND atau OR) dan k merupakan konstanta tegas sebagai konsekuen (kesimpulan).

- Model *Fuzzy Sugeno* Orde-Satu

Secara umum bentuk model *Fuzzy Sugeno* Orde-Satu adalah IF $(x_1 \text{ is } A_1) \circ (x_2 \text{ is } A_2) \circ (x_3 \text{ is } A_3) \circ \dots \circ (x_N \text{ is } A_N)$ THEN $z = p_1 * x_1 + \dots + p_N * x_N + q$, dengan A_i adalah himpunan *fuzzy* ke- i sebagai anteseden (premis), \circ adalah operator *fuzzy* (AND atau OR), p_i adalah konstanta ke- i dan q juga merupakan konstanta dalam konsekuen (kesimpulan).

Pada sistem yang dibangun, penggunaan *fuzzy logic sugeno* menggunakan model *fuzzy sugeno* Orde-Nol pada bagian pengelompokan status kucing berdasarkan pakan yang dihabiskan kucing dan aktivitas gerak kucing.

Berikut merupakan tahapan-tahapan yang harus dilakukan dalam implementasi metode *fuzzy logic sugeno*:

a) Pembentukan Himpunan *Fuzzy*

Pada tahapan ini, variabel *input* dari sistem *fuzzy* ditransfer ke dalam himpunan *fuzzy* untuk dapat digunakan dalam perhitungan nilai kebenaran dari premis pada setiap aturan dalam basis pengetahuan. Dengan demikian tahap ini mengambil nilai-nilai tegas dan menentukan derajat dimana nilai-nilai tersebut menjadi anggota dari setiap himpunan *fuzzy* yang sesuai.

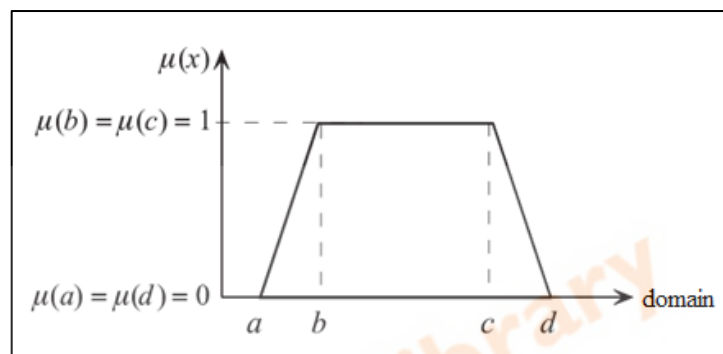
Pada himpunan tegas (crisp), nilai keanggotaan suatu item x dalam suatu himpunan A , yang sering ditulis dengan $\mu_A(x)$, memiliki dua kemungkinan yaitu:

- Satu (1), yang berarti bahwa suatu item menjadi anggota dalam suatu himpunan, atau
- Nol (0), yang berarti bahwa suatu item tidak menjadi anggota dalam suatu himpunan.

Pada himpunan *fuzzy* memiliki 2 atribut yaitu:

- *Linguistik*, yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami.
- *Numeris*, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel.

Fungsi keanggotaan (*membership function*) merupakan fungsi yang memetakan elemen suatu himpunan ke nilai keanggotaan (sering juga disebut dengan derajat keanggotaan) pada interval $[0,1]$. Fungsi keanggotaan membedakan antara himpunan *fuzzy* dengan himpunan tegas. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Ada beberapa fungsi yang dapat digunakan, fungsi yang digunakan pada sistem yang dibangun yaitu fungsi trapesium.



Gambar 4. Fungsi trapesium

Pada Gambar 4. merupakan fungsi trapesium dimana x adalah variabel yang akan dicari, a adalah batas bawah, b dan c adalah batas tengah dan d adalah batas atas dengan nilai $a < b < c < d$ (Princy & Dhenakaran, 2017). Pada fungsi trapesium ini terdapat nilai x yang memiliki derajat keanggotaan sama dengan 1, yaitu pada saat $b \leq x \leq c$. Fungsi keanggotaan trapesium ditunjukkan pada Persamaan 1.

$$\mu(x) = \left\{ \begin{array}{ll} 0 & ; \quad x \leq a \\ \frac{x-a}{b-a} & ; a < x < b \\ 1 & ; b \leq x \leq c \\ \frac{d-x}{d-c} & ; c < x < d \\ 0 & ; \quad x \geq d \end{array} \right\} \dots\dots\dots (1)$$

b) Aplikasi fungsi implikasi

Tiap-tiap aturan (proposisi) pada basis pengetahuan *fuzzy* akan berhubungan dengan suatu relasi *fuzzy*. Bentuk umum dari aturan yang digunakan dalam fungsi implikasi adalah IF x is A THEN y is B , dimana A dan B adalah *linguistic values* yang didefinisikan dalam rentang variabel X dan Y . Proposisi yang

mengikuti IF disebut sebagai anteseden (premis) sedangkan proposisi yang mengikuti THEN disebut konsekuen (kesimpulan).

Proposisi ini dapat diperluas dengan menggunakan operator *fuzzy* seperti, IF $(x_1 \text{ is } A_1) \circ (x_2 \text{ is } A_2) \circ (x_3 \text{ is } A_3) \circ \dots \circ (x_N \text{ is } A_N)$ THEN $y \text{ is } B$, dengan \circ adalah operator (misal: OR atau AND).

Secara umum fungsi implikasi yang dapat digunakan yaitu sebagai berikut:

- Min (*minimum*), pengambilan keputusan dengan fungsi min yaitu dengan mencari nilai minimum berdasarkan aturan ke-i dan dapat dinyatakan dengan Persamaan 2.

$$a_i = \mu_{A_i}(x) \cap \mu_{B_i}(x) = \min \{ \mu_{A_i}(x), \mu_{B_i}(x) \} \dots \dots \dots (2)$$

Keterangan:

a_i = nilai minimum dari himpunan *fuzzy* A dan B pada aturan ke-i,
 $\mu_{A_i}(x)$ = derajat keanggotaan x dari himpunan *fuzzy* A pada aturan ke-i,
 $\mu_{B_i}(x)$ = derajat keanggotaan x dari himpunan *fuzzy* A pada aturan ke-i.

- Dot (Hasil Kali), pengambilan keputusan dengan fungsi kali didasarkan pada aturan ke-i dinyatakan dengan Persamaan 3.

$$a_i \cdot \mu_{C_i}(Z) \dots \dots \dots (3)$$

Keterangan:

a_i = nilai minimum dari himpunan *fuzzy* A dan B pada aturan ke-i,
 $\mu_{C_i}(Z)$ = derajat keanggotaan konsekuen dari himpunan *fuzzy* C pada aturan ke-i.

Pada sistem yang dibangun, fungsi implikasi metode *fuzzy logic sugeno* yang digunakan adalah fungsi *min*.

c) Defuzzifikasi (*Defuzzification*)

Input dari proses defuzzifikasi adalah himpunan *fuzzy* yang dihasilkan dari proses komposisi dan *output* adalah sebuah nilai. Untuk aturan IF-THEN *fuzzy* dalam persamaan $RU(k) = \text{IF } x_1 \text{ is } A_{1k} \text{ and } \dots \text{ and } x_n \text{ is } A_{nk} \text{ THEN } y \text{ is } B_k$, dimana A_{1k} dan B_k berturut-turut adalah himpunan *fuzzy* dalam $U_i R$ (U dan V adalah domain fisik), $i = 1, 2, \dots, n$ dan $x = (x_1, x_2, \dots, x_n)$ U dan y V berturut-turut adalah variabel *input* dan *output* (linguistik) dari sistem *fuzzy*. *Defuzzifier* pada persamaan

di atas didefinisikan sebagai suatu pemetaan dari himpunan *fuzzy B* ke dalam $V R$ (yang merupakan output dari inferensi *Fuzzy*) ke titik tegas $y * V$. Pada metode Sugeno, *defuzzification* dilakukan dengan perhitungan *Weight Average* (WA) yang ditunjukkan pada Persamaan 4.

$$WA = \frac{\alpha_1 z_1 + \alpha_2 z_2 + \alpha_3 z_3 + \dots + \alpha_n z_n}{\alpha_1 + \alpha_2 + \alpha_3 + \dots + \alpha_n} \dots\dots\dots (4)$$

Keterangan:

WA = nilai rata-rata,

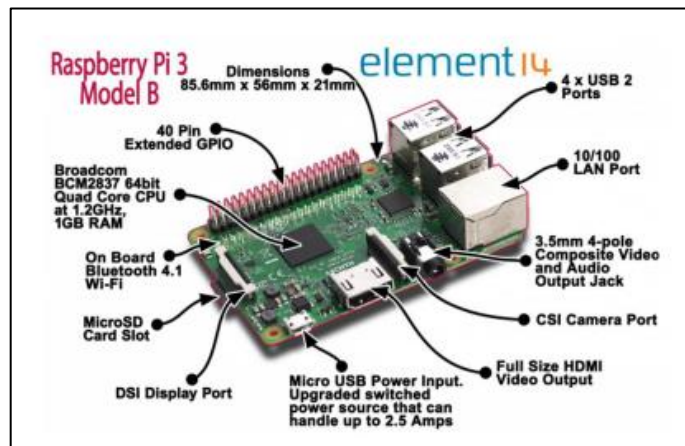
α_n = nilai predikat aturan ke-n,

z_n = indeks nilai output (konstanta) ke-n.

2.5. *Raspberry Pi*

Raspberry Pi adalah sebuah komputer papan tunggal (*single-board computer*) atau SBC berukuran kartu kredit. *Raspberry Pi* telah dilengkapi dengan semua fungsi layaknya sebuah komputer lengkap, menggunakan SoC (*System-on-a-chip*) ARM yang dikemas dan diintegrasikan diatas PCB (*Printed Circuit Board*). Perangkat ini menggunakan kartu SD untuk *booting* dan penyimpanan jangka panjang. *Raspberry Pi* bersifat *open source* (berbasis Linux) dan dapat dimodifikasi sesuai kebutuhan penggunanya. Sistem operasi utama *Raspberry Pi* menggunakan Debian GNU/Linux dan bahasa pemrograman *Python*. Salah satu pengembang OS untuk *Raspberry Pi* telah meluncurkan sistem operasi yang dinamai Raspbian, Raspbian diklaim mampu memaksimalkan perangkat *Raspberry Pi*. Sistem operasi tersebut dibuat berbasis Debian yang merupakan salah satu distribusi Linux OS (Yuwono, Nugroho, & Heriyanto, 2015).

Pada Gambar 5. merupakan tampilan dari *Raspberry Pi 3*. *Raspberry Pi 3* adalah mini komputer generasi ketiga yang dikeluarkan *Raspberry Pi* sebagai pengembangan dari generasi *Raspberry Pi* sebelumnya. Pada sistem yang dibangun, *Raspberry Pi 3* dimanfaatkan sebagai pusat pemrosesan dan pengendalian.

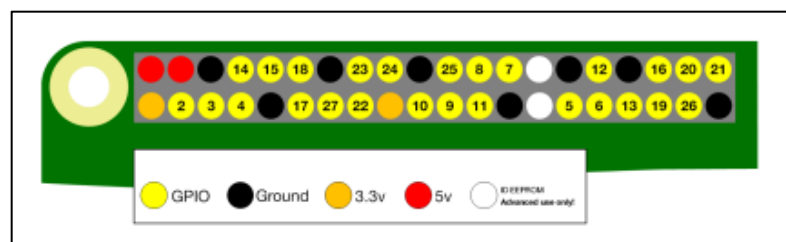


Gambar 5. Tampilan Raspberry Pi 3

Beberapa kelebihan dan peningkatan *Raspberry Pi* 3 di banding sebelumnya sebagai berikut:

1. *Raspberry Pi* 3 menggunakan *chipset* baru yaitu *Broadcom BCM2873B0* Cortex A53 64-bit 1,4GHz, mempunyai kecepatan *processor* jauh lebih cepat.
2. Memiliki kemampuan jaringan lebih baik dengan koneksi *wireless dual band* yang sudah mendukung 802.11ac dan *Bluetooth* 4.2.
3. *Chipset* pada *Raspberry Pi* 3 memiliki manajemen suhu yang lebih baik.
4. *Faster Ethernet* (Gigabit Ethernet over USB 2.0).
5. *Power-over-Ethernet support* (with separate PoE HAT) .

Salah satu fitur yang terdapat pada *Raspberry Pi* yaitu fitur *GPIO (General Purpose Input Output)* pin yang terletak di salah satu sudut papan didekat *port* USB. *GPIO* merupakan penghubung antara *Raspberry Pi* dengan komponen lain yang berada di luar *Raspberry Pi*, dimana *Raspberry Pi* dapat berfungsi sebagai *input* atau *output* secara bersamaan. Terdapat 40 pin *GPIO* yang dapat digunakan pada *Raspberry Pi* seperti yang ditunjukkan pada Gambar 6.



Gambar 6. *Raspberry Pi* GPIO pin

Dari 26 pin GPIO yang dimiliki *Raspberry Pi*, terdapat 2 pin sebagai sumber tegangan 5 V, 2 pin sumber tegangan 3.3 V, 5 pin *ground*, 17 pin *input / output*. GPIO pada *Raspberry Pi* dapat dikendalikan dengan berbagai cara salah satunya dengan terminal menggunakan *bash script* atau dengan bahasa program yang lain. Pada Gambar 7. merupakan penjelasan lebih lanjut mengenai fungsi masing-masing pin GPIO yang terdapat pada *Raspberry Pi 3*.

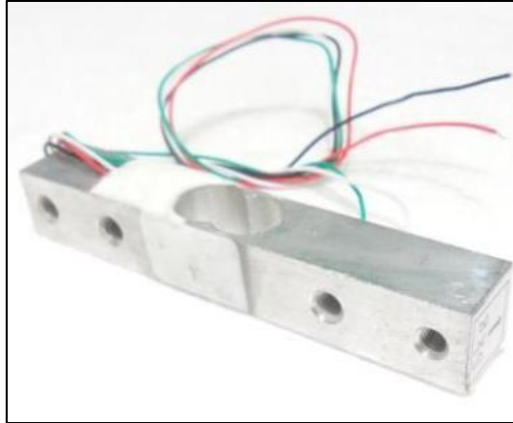
Pin#	NAME	NAME	Pin#
01	3.3v DC Power	DC Power 5v	02
03	GPIO02 (SDA1 , I ² C)	DC Power 5v	04
05	GPIO03 (SCL1 , I ² C)	Ground	06
07	GPIO04 (GPIO_GCLK)	(TXD0) GPIO14	08
09	Ground	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	Ground	14
15	GPIO22 (GPIO_GEN3)	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	Ground	20
21	GPIO09 (SPI_MISO)	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	(SPI_CE0_N) GPIO08	24
25	Ground	(SPI_CE1_N) GPIO07	26
27	ID_SD (I ² C ID EEPROM)	(I ² C ID EEPROM) ID_SC	28
29	GPIO05	Ground	30
31	GPIO06	GPIO12	32
33	GPIO13	Ground	34
35	GPIO19	GPIO16	36
37	GPIO26	GPIO20	38
39	Ground	GPIO21	40

Gambar 7. *Raspberry Pi 3* GPIO pin

2.6.Sensor Berat (*Load cell*)

Sensor *load cell* merupakan sensor yang dirancang untuk mendeteksi tekanan atau berat sebuah beban, sensor *load cell* umumnya digunakan sebagai komponen utama pada sistem timbangan digital dan dapat diaplikasikan pada jembatan timbangan yang berfungsi untuk menimbang berat dari truk pengangkut bahan baku, pengukuran yang dilakukan oleh *load cell* menggunakan prinsip tekanan. Pada Gambar 8. merupakan bentuk fisik dari *load cell* (AmericaModule, 2010).

Pada sistem yang dibangun, sensor berat digunakan sebagai pendeteksian berat badan pada kucing dan berat pada wadah pakan kucing.



Gambar 8. *Load cell*

2.7.Sensor Gerak (PIR)

Pada Gambar 9. merupakan Sensor PIR (*Passive Infra Red*) sensor yang digunakan untuk mendeteksi adanya pancaran sinar infra merah.

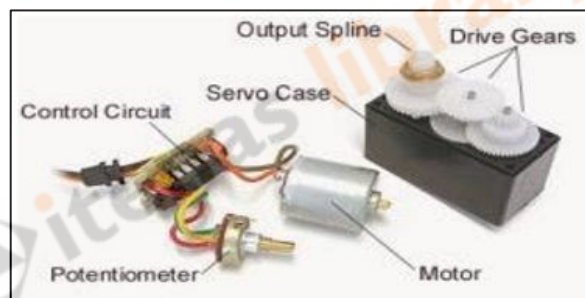


Gambar 9. *Sensor PIR*

Sensor PIR bersifat pasif, artinya sensor ini tidak memancarkan sinar infra merah tetapi menerima radiasi sinar infra merah dari luar. Sensor PIR ini digunakan dalam perancangan detektor gerak. Karena semua benda memancarkan energi radiasi, sebuah gerakan akan terdeteksi ketika sumber infra merah dengan suhu tertentu seperti manusia melewati sumber infra merah yang lain dengan suhu yang berbeda seperti dinding maka sensor akan membandingkan pancaran infra merah yang diterima setiap waktu, sehingga jika ada pergerakan maka akan terjadi perubahan pembacaan pada sensor (Dakhi, 2017). Pada sistem yang dibangun, sensor gerak digunakan untuk pendeteksian gerak yang dilakukan oleh kucing.

2.8. Motor Servo

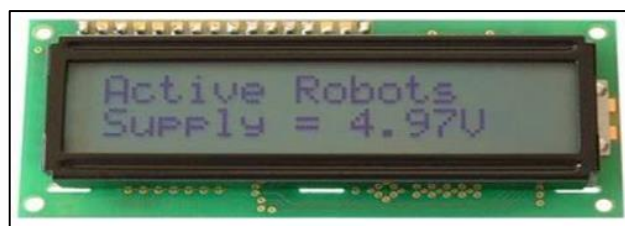
Pada Gambar 10. merupakan motor servo. Motor Servo adalah sebuah perangkat atau aktuator putar (motor) yang dirancang dengan sistem kontrol umpan balik *loop* tertutup (servo), sehingga dapat diatur untuk menentukan dan memastikan posisi sudut dari poros *output* motor. motor servo merupakan perangkat yang terdiri dari motor DC, serangkaian *gear*, rangkaian kontrol dan potensiometer. Serangkaian *gear* yang melekat pada poros motor DC akan memperlambat putaran poros dan meningkatkan torsi motor servo, sedangkan potensiometer dengan perubahan resistansinya saat motor berputar berfungsi sebagai penentu batas posisi putaran poros motor servo (Mardiansyah, 2016). Pada sistem yang dibangun, motor servo digunakan sebagai gerbang untuk membuka atau menutup pada saat pengisian pakan kucing dari penampungan ke wadah tempat pakan kucing.



Gambar 10. Motor servo

2.9. LCD

LCD adalah *display* aplikasi yang mampu menampilkan lebih banyak karakter, LCD digunakan antara lain untuk menampilkan pesan atau informasi kepada pengguna, LCD dibedakan berdasarkan jumlah baris dan kolom karakter yang ditampilkan. Misalnya LCD 2 x 16 menunjukkan 2 baris dan 16 kolom karakter. Seperti pada Gambar 11. berikut ini:



Gambar 11. LCD

Pada sistem yang dibangun, LCD dimanfaatkan untuk menampilkan informasi kepada pengguna mengenai berat badan kucing dan berat pada wadah tempat pakan kucing.

2.10. VNC Viewer

VNC Viewer (*Virtual Network Computing Viewer*) merupakan sebuah *software remote control* yang digunakan untuk mengakses atau mengontrol suatu komputer dengan komputer lainnya yang terhubung dengan jaringan LAN (*Local Area Network*) maupun jaringan Internet. Gambar 12. merupakan logo dari *software VNC Viewer*. Pada sistem yang dibangun, *VNC Viewer* digunakan sebagai *remote control Raspberry Pi* dengan PC atau Laptop.



Gambar 12. Logo VNC Viewer

Beberapa kelebihan dari *software VNC* dibandingkan dengan *software-software* sejenisnya yaitu: (Bakhrun, 2013)

1. *Multi Platform*

Software VNC dapat digunakan di lingkungan Windows, Linux, Beos, Macintosh, Unix dan lain-lain, bahkan penggunaannya dapat dilakukan secara lintas platform. *VNC client & VCN server* dapat saling diakses misalnya dari sistem operasi Windows ke sistem operasi Linux, maupun dari sistem operasi Linux ke sistem operasi Windows.

2. *Client-Server*

Software VNC terdiri dari aplikasi *server* dan *client* yang harus diinstal di kedua sisi. Bagi orang-orang tertentu hal ini mungkin merepotkan, tetapi hal ini dimaksudkan untuk melindungi *privacy* komputer yang menggunakan VNC.

3. *HTTP Support*

VNC dapat diakses menggunakan *default port* 5900 atau 5901 untuk TCP (*Transmission Control Protocol*) maupun *port* 5800 atau 5801 untuk HTTP

(*Hypertext Transfer Protocol*). Sehingga VNC *server* dapat diakses oleh VNC *client* menggunakan *browser* seperti Mozilla Firefox, Opera, dan Internet Explorer dengan menggunakan Java Applet.

4. Transparan

VNC adalah sebuah program yang transparan, tidak seperti beberapa *software remote desktop* yang lain yang menyembunyikan keberadaannya dari *user* awam sehingga dapat dikategorikan sebagai sebuah *software* trojan yang akan dideteksi oleh *software* antivirus. Apabila sebuah komputer Windows dipasang VNC *server*, akan muncul sebuah *icon* kecil logo VNC di sebelah kanan *taskbar* yang akan berubah warna dari abu-abu menjadi biru apabila komputer tersebut sedang diakses. VNC juga mengharuskan pengguna memasang *password* untuk bisa diaktifkan. Sebelum *password* dipasang, program ini tidak akan bisa digunakan.

5. *Open Source*

VNC bersifat *open source* dengan lisensi GPL (*General Public License*). Dengan begitu pengguna dapat leluasa menggunakannya, meski demikian tentu harus mengikuti sifat lisensi *open source*-nya.

2.11. *Advanced IP Scanner*

Advanced IP Scanner merupakan perangkat lunak (*software*) yang memiliki fungsi untuk memindai ratusan alamat IP secara simultan (bersamaan) pada kecepatan tinggi, melakukan *scan port* komputer jaringan, menemukan HTTP, HTTPS, FTP dan folder *sharing*. Selain itu, *Advanced IP Scanner* dapat mendeteksi perangkat *wireless* (termasuk jaringan Wi-Fi *router*) dan *network camera* yang terhubung ke jaringan serta memberikan informasi lebih lanjut mengenai nama komputer dan MAC *address* perangkat tersebut. Gambar 13. merupakan logo dari *software Advanced IP Scanner*. Pada sistem yang dibangun, *Advanced IP Scanner* digunakan untuk memindai jaringan dan mendapatkan alamat IP pada *Raspberry Pi*.



Gambar 13. Logo VNC Viewer

Advanced IP Scanner memiliki beberapa fitur yang disediakan, diantaranya:
(Wirandanu, 2013)

1. *Fast Network Scanning*

Advanced IP Scanner dapat memindai alamat IP secara cepat, sehingga pengguna tidak perlu menunggu dengan waktu yang lama untuk mendapatkan alamat IP yang diinginkan.

2. *Integration with Radmin remote access software*

Advanced IP Scanner dan Radmin (*Remote Administrator*) sangat terintegrasi. *Advanced IP Scanner* memungkinkan pengguna untuk memindai jaringan, menemukan semua komputer yang menjalankan Radmin, *server*, dan dapat terhubung ke salah satu dari mereka. Dengan Radmin, pengguna dapat mengakses PC dengan kontrol penuh, transfer file, dan mode Telnet.

3. *Remote PC Shutdown*

Fitur ini mampu untuk mematikan PC sistem operasi *Windows* yang terhubung di jaringan. Pengguna dapat menggunakan hak akses *default* atau menentukan *login* dan *password* untuk *shutdown*.

4. *Wake-On-LAN*

Pengguna dapat menghidupkan komputer secara jarak jauh dengan menggunakan *Advanced IP Scanner* jika komputer tersebut mendukung fitur “Wake-On-LAN”.

5. *Simple dan User Friendly*

Advanced IP Scanner merupakan *software* yang *simple* serta *user friendly interface*. Untuk operasi *batch* sederhana, memungkinkan pengguna untuk menyimpan daftar komputer yang akan dipindai dalam daftar *favorites*. *Advanced IP Scanner* akan secara otomatis memanggil *favorit* pengguna pada

saat *startup*. Pengguna kemudian dapat memilih apakah akan memindai seluruh jaringan atau hanya komputer yang berada pada daftar *favorites*.

2.12. MariaDB

MariaDB dirancang sebagai pengganti untuk MySQL yang dirilis pada bulan Januari tahun 2009. Para pengembang MariaDB sebagian besar terdiri dari mantan pengembang MySQL. MariaDB adalah sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan MariaDB, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. Kemampuan utama MariaDB sama dengan kemampuan pada MySQL. MariaDB juga mencakup sistem informasi geografis (GIS) dan JavaScript Object Notation (JSON) (Gomes, 2019). Pada sistem yang dibangun, MariaDB dimanfaatkan sebagai *server* yang didalamnya terdapat basis data untuk menampung data-data yang akan digunakan oleh sistem.

