BAB 2
TINJAUAN PUSTAKA

2.1. Profil Instansi

Dinas Peternakan Perikanan Dan Kelautan beralamat Jl. Slamet Riyadi No.8 Cianjur merupakan urusan pelaksana otonom daerah di bidang perternakan perikanan dan kelautan,dipimpin oleh seorang Kepala Dinas yang berada di bawah dan bertanggungjawab kepada Bupati melalui Sekretaris Daerah. Dinas Peternakan Perikanan Dan Kelautan . Dinas Peternakan Perikanan Dan Kelautan mempunyai tugas melaksanakan urusan pemerintah daerah di bidang peternakan perikanan dan kelautan berdasarkan asas otonomi dan tugas pembantuan.

Dalam melaksanakan tugas sebagaimana dimaksud diatas, Dinas Peternakan Perikanan Dan Kelautan, menyelenggarakan fungsi :

a. Perumusan kebijakan teknis dinas di bidang perencanaan, pelaksanaan, pembinaan, evaluasi dan lapoan penyelenggara sebagai urusan pemerintah di bidang perternakan perikanan dan kelautan sesuai dengan ketentuan dan/atau peraturan perundang-undangan yang berlaku;
b. Penyelenggaraan urusan pemerintah dan pelayanan umum di bidang perternakan perikanan dan kelautan sesuai dengan ketentuan dan/atau peraturan perundang-undangan yang berlaku;
c. Pembinaan dan Pelaksanaan tugas dinas dalam menyelenggarakan sebagai urusan pemerintah di bidang perternakan perikanan dan kelautan sesuai dengan ketentuan dan/ atau peraturan perundang-undangan yang berlaku;
d. Pelaksanaan tugas lain yang diberikan oleh Bupati sesuai dengan tugas dan fungsi Dinas.

2.1.1. Sejarah Instansi

Dinas Peternakan Perikanan dan Kelautan Cianjur pada awalnya bernama Jawatan Hewan dan Jawatan Perikanan darat (PERIDAR). Jawatan Hewan di Cianjur berdiri pada tahun 1950, dan diubah namanya menjadi Dinas Peternakan pada tahun 1962. Pada saat itu kedua Dinas ini terpisah. Barulah pada tahun 2002 Dinas Peternakan, dan Perikanan digabungkan, karena ditujukan untuk :

a. Mengefektifitaskanpelaksanaan program kerja.
b. Mengefisienkan biaya yang ada didaerah/ Kabupaten.
c. Memudahkan pengawasan.
d. Mempercepat (akselerasi) pekerjaan.

Kemudian pada tahun 2008 terjadi perubahan nama pada dinas tersebut menjadi Dinas Peternakan Perikanan dan Kelautan.

2.1.2. Visi dan Misi Instansi

2.1.2.1. Visi

Kinerja Dinas Peternakan Perikanan dan Kelauatan dalam pembangunan peternakan, perikanan dan kelautan didasarkan pada peningkatan pelayanan kepada masyarakat secara profesional seperti yang tertera dalam visi Dinas Peternakan Perikanan dan Kelauatan kabupaten cianjur, yang memiliki bunyi sebagai berikut:

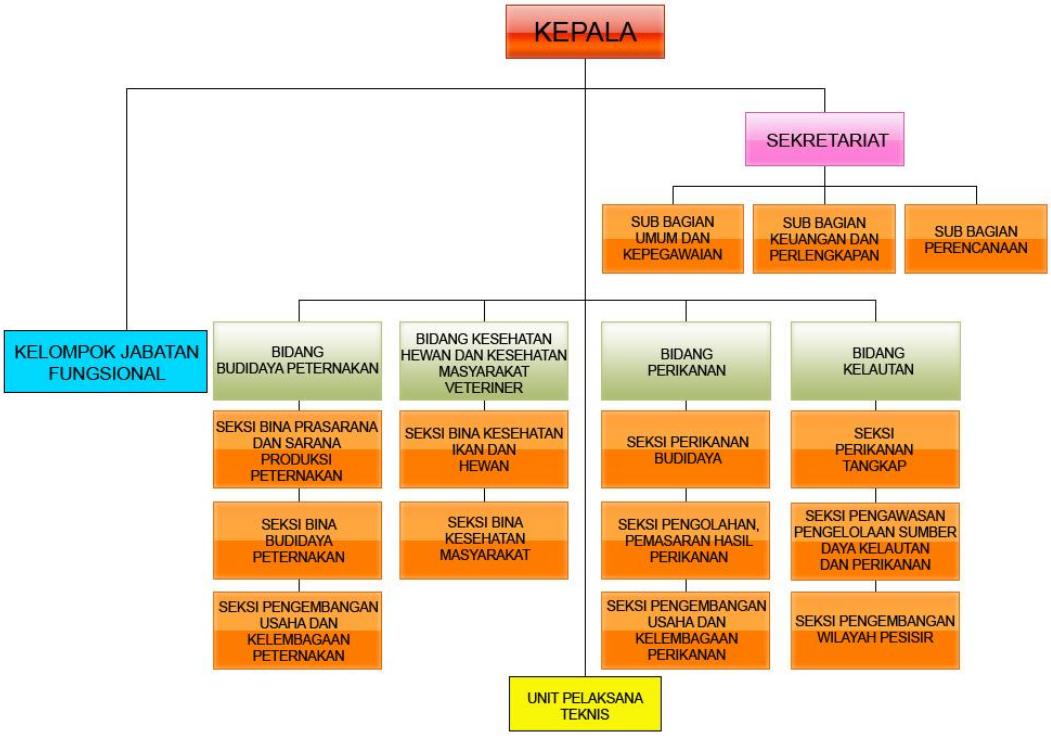
“ Melalui Akselerasi Pembangunan Peternakan, Perikanan Dan Kelautan Kita Wujudkan Masyarakat Cianjur Yang Cerdas, Sehat, sejahtera Dan Berakhlakul karimah”

2.1.2.2. Misi

Dinas Peternakan Perikanan dan Kelautan Cianjur nmenetapkan misi yang saling terkait yaitu:

a. Meningkatkan pelayanan kepada masyarakat secara profesional dalam memfasillitasi pembangunan perikanan dan peternakan berbasis agribisnis, marinebisnis dan pariwisata berwawasan lingkungan”
b. Memfasilitasi Penyediaan pangan protein asal ternak dan ikan yang cukup (kualitas, kuantitas).
c. Menciptakan peluang ekonomi untuk meningkatkan pendapatan petani ternak, pembudidayaan ikan dan nelayan.
d. Meningkatkan SDM (Sumber Daya Manusia) Aparatur, peternak, pembudidayaan ikan, nelayan dan pelaku pasca panen, agar dapat menghasilkan jasa dan produk yang berdaya saing tinggi.
e. Melestarikan dan memanfaatkan sumber daya alam pendukung yang dapat dijadikan komoditas *agro-marine* wisata.

2.1.3. Struktur Organisasi dan Deskripsi kerja



Gambar 2. 1. Struktur Organisasi Dinas Peternakan, Perikanan,dan Kelautan.

- **Tugas dan Wewenang**

a. Kepala Dinas

Kepala dinas mempunyai tugas membantu bupati dalam menyelenggarakan urusan pemerintahan daerah daerah dibidang peternakan,perikanan dan kelautan serta memimpin, mengkoordinasikan seluruh kegiatan dinas sesuai dengan ketentuan atau peraturan perundand-undangan yang berlaku.

b. Sekertariat

Sekertariat mempunyai tugas menyenggarakan sebagian tugas dan fungsi dinas dalam melaksanakan koordinasi pengelolaan urusan umum, kepegawaian, keuangan, perlengkapan, penyusunan rencana, evaluasi dan laporan program dan kegiatan sesuai dengan ketentuan atau peraturan perundang-undangan yang berlaku

Dalam melaksanankan tugas sebagai mana dimaksud pada pasal 7, sekertariat menyenggarakan fungsi:

a. Pengkoordinasian dan penyusunan program, rencana kegiatan dan anggaran sekertariat
b. Pengkoordinasian dan penyusunan rencana strategis, program, dan kegiatan dan anggaran dinas
c. Pengkoordinasian dan penyiapan bahan penyusunan kebijakan umum pemerintah daerah dibidang peternakan, perikanan, dan kelautan
d. Pengelolaan administrasi umum, rumah tangga, hubungan masyarakat, dan keprotokolan
e. Pengelolaan administrasi kepegawaian
f. Pengkoordinasian dan penyusunan bahan pembinaan pegawai
g. Pengelolaan administrasi keuangan dan perlengkapan
h. Pengkoordinasian dan penyusunan evalaasi, dan laporan pelaksanaan program dan kegiatan dinas
i. Pengelolaan perpustakaan, data dan sistem in*form*asi peternakan , perikanan, dan kelautanPelaksanaan tugas lain yang diberikan oleh kepala dinas.

c. Bidang Budidaya Peternakan

Bidang budidaya peternakan, mempunyai tugas menyelenggarakan sebagian tugas dan fungsi dinas di bidang pembinaan, prasarana dan sarana, usaha dan budidaya peternakan sesuai dengan ketentuan dan atau peraturan perundang-undangan yang berlaku.

Dalam melaksanalan tugas pembinaan, bidang budidaya peternakan menyelenggarakan fungsi :

a. Penyusunan program dan kegiatan budidaya peternakan
b. Pengkoordinasian dan penyiapan bahan penyususnnan kebijakan umum pemerintah daerah di bidang pembinaan prasarana dan sarana produksi, budidaya dan usaha peternakan.
c. Penyiapan dan penyajian data dan in*form*asi mengenai potensi serta permasalahan di bidang pembinaan prasaran dan srana produksi budidaya dan usaha peternakan

d. Penyiapan bahan koordianasi penyusunan kebijakan teknis dians di bidang pembinaan di bidang prasarana dan sarana produksi, budidaya dan usaha peternakan sesuai dengan ketentuan atau peraturan perundang0-undangan yang berlaku.
e. Penyiapan bahan koordinasi penyusunan rencana dan pelaksanaan program, kegiatan dinas di bidang pembinaan prasarana, sarana produksi, budidaya dan usaha peternakan
f. Pelaksanaan pembinaan teknis operasional pembinaan prasarana , sarana produksi, budidaya dan usaha peternakan sesuai dengan ketentuan dan atau peratuaran perundang-undangan yang berlaku.
g. Pelaksnanaan koordinasi konsultasi dengan unit organisasi di lingkungan dinas atau lingkungan dinas lain yang terkait dengan tugas bidang-bidang budidaya peternakan
h. Pelaksanaan pelayanan umum di bidang pembinaan presaran dan sarana produsi, budidaya dan usaha peternakan sesuai denga ketentuan atau peraturan perundang-undangan yang berlaku
i. Penyiapan bahan koordinasi, penyusunan dvaluasi dan laporan pelaksanaan proaram dan kegiatan dinas di bidang pembinaan sarana dan prasarana produksi, budidaya dan usaha peternakan sesuai dengan ketentuan atau peraturan perundang-undangan yang berlaku.
j. Pelaksanaan tugas lain yang diberikan oleh kepala dinas berdasarkan ketentuan atau peraturan perundang-undangan yang berlaku

Untuk menyelenggarakan fungsi sebagaimana dimaksud pada ayat 1 bidang budidaya peternakan dibantu oleh:

a. Seksi bina prasarana dan sarana produksi peternakan
b. Seksi biana budidaya peternakan
c. Seksi bina usaha peternakan

2.2. Landasan Teori

Landasan teori sangat penting dalam sebuah penelitian, untuk mendukung pembuatan laporan tugas akhir ini, maka perlu dikemukakan hal-hal atau teoriteori

yang berkaitan dengan permasalahan dan ruang lingkup pembahasan sebagai landasan dalam pembuatan laporan tugas akhir ini.

2.2.1. Data dan In*form*asi

Hubungan antara data dan in*form*asi sangatlah erat sebagaimana hubungan antara sebab dan akibat. Bahwa data merupakan bentuk dasar dari sebuah in*form*asi, sedangkan in*form*asi merupakan elemen yang dihasilkan dari suatu bentuk pengolahan data.

2.2.1.1. Data

Secara konseptual, data adalah deskripsi tentang benda, kejadian, aktivitas dan transaksi, yang tidak mempunyai makna atau tidak berpengaruh secara langsung kepada pemakai. Data sering kali disebut sebagai bahan mentah in*form*asi.

Berikut adalah kutipan pengertian data dari sudut pandang yang berbeda :

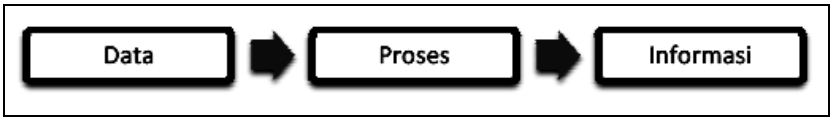
1. Menurut kamus bahasa Inggris-Indonesia, data diterjemahkan sebagai istilah yang berasal dari kata “datum” yang berarti fakta atau bahan-bahan keterangan.
2. Dari sudut pandang bisnis, terdapat pengertian data bisnis sebagai berikut: *“Bussines data is an organization’s description of things* (*resources*) *and events (transactions) that it faces”*. Jadi data, dalam hal ini disebut sebagai data bisnis, merupakan deskripsi organisasi tentang sesuatu (*resource*) dan kejadian (*transaction)* yang terjadi.
3. Menurut Jogiyanto, data merupakan bentuk jamak dari datum (kenyataan) yang berupa fakta-fakta, angka-angka, gambar-gambar yang dapat ditarik kesimpulannya.
4. Gordon B. Davis dalam bukunya *Management Informations System : Conceptual Foundations, Structure, and Development* menyebut data sebagai bahan mentah dari in*form*asi, yang dirumuskan sebagai sekelompok lambang-lambang tidak acak yang menunjukkan jumlah atau tindakan atau hal-hal lain.

Dari keempat pengertian diatas, dapat diambil kesimpulan bahwa data adalah bahan baku in*form*asi, didefinisikan sebagai kelompok teratur simbol-simbol yang

mewakili kuantitas, tindakan, benda,dan sebagainya. Data terbentuk dari karakter, dapat berupa alphabet, angka, maupun simbol khusus seperti *, $, dan /. Data disusun untuk diolah dalam bentuk struktur data, struktur file, dan basis data.

2.2.1.2. Informasi

In*form*asi adalah data yang telah diproses menjadi bentuk yang memiliki arti bagi penerima dan dapat berupa fakta, suatu nilai yang bermanfaat. Jadi ada suatu proses trans*form*asi data menjadi suatu in*form*asi *(input – proses – output*).



Gambar 2. 2. Pemrosesan Data Menjadi In*form*asi

Definisi umum untuk in*form*asi dalam sistem in*form*asi menurut Jogiyanto H.M (1990; 11) : "In*form*asi adalah bentuk data yang dapat diolah yang lebih berguna dan berarti bagi yang menerimanya". Menurut Robert G. Munik (1973 ; 12) : "In*form*asi adalah data yang telah diolah menjadi suatu bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau mendatang.Berikut pengertian in*form*asi dari berbagai sumber :

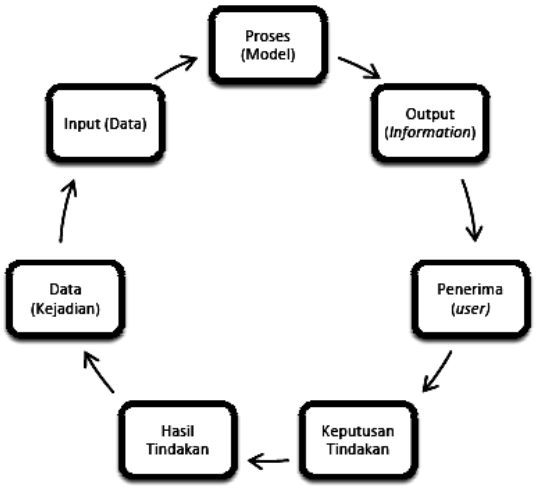
1. Menurut Gordon B. Davis dalam bukunya *"Management Informations System: Conceptual Foundations, Structures, and Development* menyebut in*form*asi sebagai data yang telah diolah menjadi bentuk yang berguna bagi penerimanya dan nyata, berupa nilai yang dapat dipahami di dalam keputusan sekarang maupun masa depan.
2. Menurut Berry E. Cushing dalam bukunya*"Accounting Information System and Business Organization"* dikatakan bahwa in*form*asi merupakan sesuatu yang menunjukkan hasil pengolahan data yang diorganisasi dan berguna kepada orang yang menerimanya.
3. Menurut Robert N. Anthony dan John Dearden dalam bukunya *"Managemet Control Systems"*, menyebut in*form*asi sebagai suatu kenyataan, data, item, yangmenambah pengetahuan bagi penggunanya.

4. Menurut Stephan A. Moscove dan Mark G. Simkin dalambukunya *"Accounting Information Systems: Concepts and Practise"* mengatakan in*form*asi sebagai kenyataan atau bentuk-bentuk yang berguna yang dapat digunakan untuk pengambilan keputusan bisnis.

Dari keempat pengertian seperti tersebut di atas dapat disimpulkan bahwa in*form*asi hasil dari pengolahan data menjadibentuk yang lebih berguna bagi yang menerimanya yang menggambarkan suatu kejadian-kejadian nyata dan dapat digunakan sebagai alat bantu untuk pengambilan suatu keputusan.

2.2.1.2.1. Siklus Informasi

Untuk memperoleh in*form*asi yang bermanfaat bagi penerimanya, perlu untuk dijelaskan bagaimana siklus yang terjadi atau dibutuhkan dalam menghasilkan in*form*asi. Pertama-tama data dimasukkan ke dalam model yang umumnya memiliki urutan proses tertentu dan pasti, setelah diproses akan dihasilkan in*form*asi tertentu yang membuat suatu keputusan atau melakukakn tindakan tertentu. Dari keputusan atau tindakan tersebut akan menghasilkan atau diperoleh kejadian-kejadian tertentu yang akan digunakan kembali sebagai data yang nantinya akan dimasukkan ke dalam model (proses), begitu seterusnya. Dengan demikian akan membentuk suatu siklus in*form*asi (*information cycle*) atau siklus pengolahan data (*data processing cycles*), seperti gambar 2.3 Siklus In*form*asi.



Gambar 2. 3. Siklus In*form*asi

2.2.2. Konsep Perancangan B*error*ientasi Objek

OOP (*Object Oriented Programming*) adalah suatu metode pemrograman yang b*error*ientasi kepada objek. Tujuan dari OOP diciptakan adalah untuk mempermudah pengembangan program dengan cara mengikuti model yang telah ada di kehidupan sehari-hari. Jadi setiap bagian dari suatu permasalahan adalah objek, objek itu sendiri merupakan gabungan dari beberapa objek yang lebih kecil lagi. Contoh Pesawat, Pesawat adalah sebuah objek. Pesawat itu sendiri terbentuk dari beberapa objek yang lebih kecil lagi seperti mesin, roda, baling-baling, kursi, dll. Pesawat sebagai objek yang terbentuk dari objek-objek yang lebih kecil saling berhubungan, berinteraksi, berkomunikasi dan saling mengirim pesan kepada objek-objek yang lainnya. Begitu juga dengan program, sebuah objek yang besar dibentuk dari beberapa objek yang lebih kecil, objek-objek itu saling berkomunikasi, dan saling berkirim pesan kepada objek yang lain.

Konsep OOP, terbagi menjadi 4 kelas, yaitu sebagai berikut:

1. Kelas Abstrak (*Class Abstract*)
   a. Kelas merupakan deskripsi abstrak in*form*asi dan tingkah laku dari sekumpulan data.
   b. Kelas dapat diilustrasikan sebagai suatu cetak biru(blueprint) atau prototipe yang digunakan untuk menciptakan objek.
   c. Kelas merupakan tipe data bagi objek yang mengenkapsulasi data dan operasi pada data dalam suatu unit tunggal.
   d. Kelas mendefinisikan suatu struktur yang terdiri atas data kelas (*data field*), prosedur atau fungsi (*method*), dan sifat kelas (*property*).

2. Enkapsulasi (*encapsulation*)
   a. Istilah enkapsulasi sebenarnya adalah kombinasi data dan fungsionalitas dalam sebuah unit tunggal sebagai bentuk untuk menyembunyikan detail in*form*asi.
   b. Proses enkapsulasi memudahkan kita untuk menggunakan sebuah objek dari suatu kelas karena kita tidak perlu mengetahui segala hal secara rinci.

c. Enkapsulasi menekankan pada antarmuka suatu kelas, atau dengan kata lain bagaimana menggunakan objek kelas tertentu.
d. Contoh: kelas mobil menyediakan antarmuka fungsi untuk menjalankan mobil tersebut, tanpa kita perlu tahu komposisi bahan bakar, udara dan kalor yang diperlukan untuk proses tersebut.

3. Pewarisan (*Inheritance*)

a. Kita dapat mendefinisikan suatu kelas baru dengan mewarisi sifat dari kelas lain yang sudah ada.
b. Penurunan sifat ini bisa dilakukan secara bertingkattingkat, sehingga semakin ke bawah kelas tersebut menjadi semakin spesifik.
c. Sub kelas memungkinkan kita untuk melakukan spesifikasi detail dan perilaku khusus dari kelas supernya.
d. Dengan konsep pewarisan, seorang programmer dapat menggunakan kode yang telah ditulisnya pada kelas super berulang kali pada kelas-kelas turunannya tanpa harus menulis ulang semua kodekode itu.

4. Polimorfisme (*polymorphism*)

a. Polimorfisme merupakan kemampuan objekobjek yang berbeda kelas namun terkait dalam pewarisan untuk merespon secara berbeda terhadap suatu pesan yang sama.
b. Polimorfisme juga dapat dikatakan kemampuan sebuah objek untuk memutuskan *method* mana yang akan diterapkan padanya, tergantung letak objek tersebut pada jenjang pewarisan.
c. *Method overriding*.
d. *Method name overloading*.

2.2.3. *Unified Modeling Language* (UML)

UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk visualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis OO (*Object-Oriented*). UML tidak hanya merupakan sebuah bahasa pemograman visual saja, namun juga dapat secara langsung dihubungkan ke berbagai bahasa pemograman, seperti JAVA, C++, *Visual Basic*, atau bahkan

dihubungkan secara langsung ke dalam sebuah *object-oriented database*. Bagian - bagian utama dari UML adalah *View, diagram*, model element, dan *general mechanism* [13].

1. View

View digunakan untuk melihat sistem yang dimodelkan dari beberapa aspek yang berbeda. *View* bukan melihat grafik, tapi merupakan suatu abstraksi yang berisi sejumlah diagram. Beberapa jenis *View* dalam UML antara lain : *use case View, logical View, component View, concurrency View*, dan *deployment View*.

2. Use case View

Mendeskripsikan fungsionalitas sistem yang seharusnya dilakukan sesuai yang diinginkan external actors. *Actor* yang berinteraksi dengan sistem dapat berupa *User* atau sistem lainnya. *View* ini digambarkan dalam *use case diagrams* dan kadang-kadang dengan *activity diagrams*. *View* ini digunakan terutama untuk pelanggan, perancang (*designer*), pengembang (*developer*), dan penguji sistem (*tester*).

3. Logical View

Mendeskripsikan bagaimana fungsionalitas dari sistem, struktur statis (*class, object,* dan *relationship*) dan kolaborasi dinamis yang terjadi ketika object mengirim pesan ke object lain dalam suatu fungsi tertentu. *View* ini digambarkan dalam *class* diagrams untuk struktur statis dan dalam *state, sequence, collaboration*, dan *activity* diagram untuk model dinamisnya. *View* ini digunakan untuk perancang (*designer*) dan pengembang (*developer*).

4. Component View

Mendeskripsikan implementasi dan ketergantungan modul. Komponen yang merupakan tipe lainnya dari *code module* diperlihatkan dengan struktur dan ketergantungannya juga alokasi sumber daya komponen dan in*form*asi *administrative* lainnya. *View* ini digambarkan dalam component *View* dan digunakan untuk pengembang (*developer*).

5. Concurrency View

Membagi sistem ke dalam proses dan prosesor. *View* ini digambarkan dalam diagram dinamis (*state, sequence, collaboration,* dan *activity diagrams*) dan

diagram implementasi (*component* dan *deployment diagrams*) serta digunakan untuk pengembang , pengintegrasi, dan penguji.

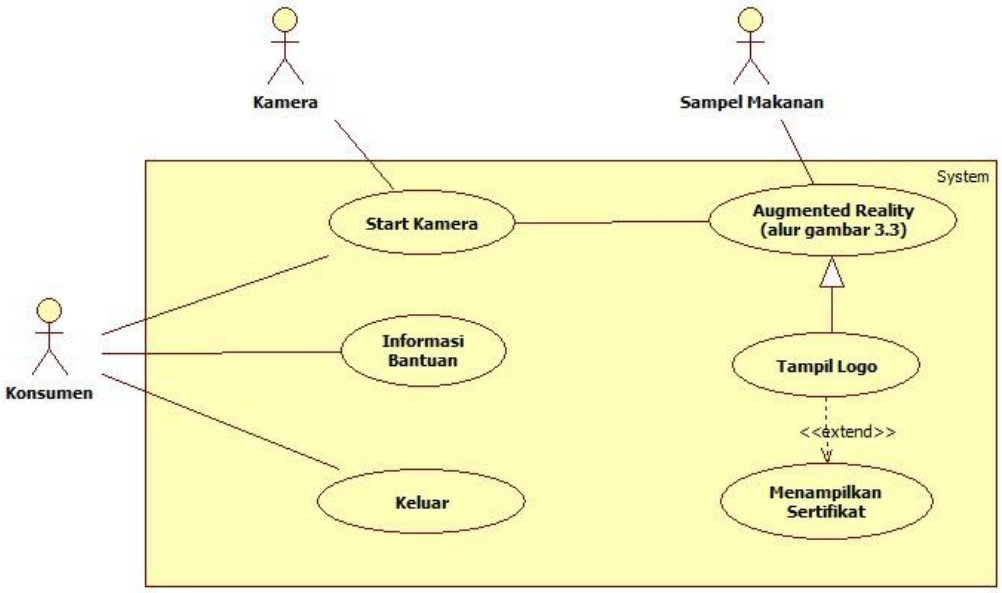
6. Deployment View

Mendeskripsikan fisik dari sistem seperti komputer dan perangkat (*nodes*) dan bagaimana hubungannya dengan yang lain. *View* ini digambarkan dalam deployment diagrams dan digunakan untuk pengembang (*developer*), pengintegrasi (*integrator*), dan penguji (*tester*).

Berikut ini menjelaskan tentang diagram-diagram yang ada di UML:

1. Use Case Diagram

Use case adalah abstraksi dari interaksi antara sistem dan *actor*. *Use case* bekerja dengan cara mendeskripsikan tipe interaksi antara *User* sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. *Use case* merupakan konstruksi untuk mendeskripsikan bagaimana sistem akan terlihat di mata *User*. Sedangkan *use case diagram* memfasilitasi komunikasi diantara analis dan pengguna serta antara analis dan *client*.



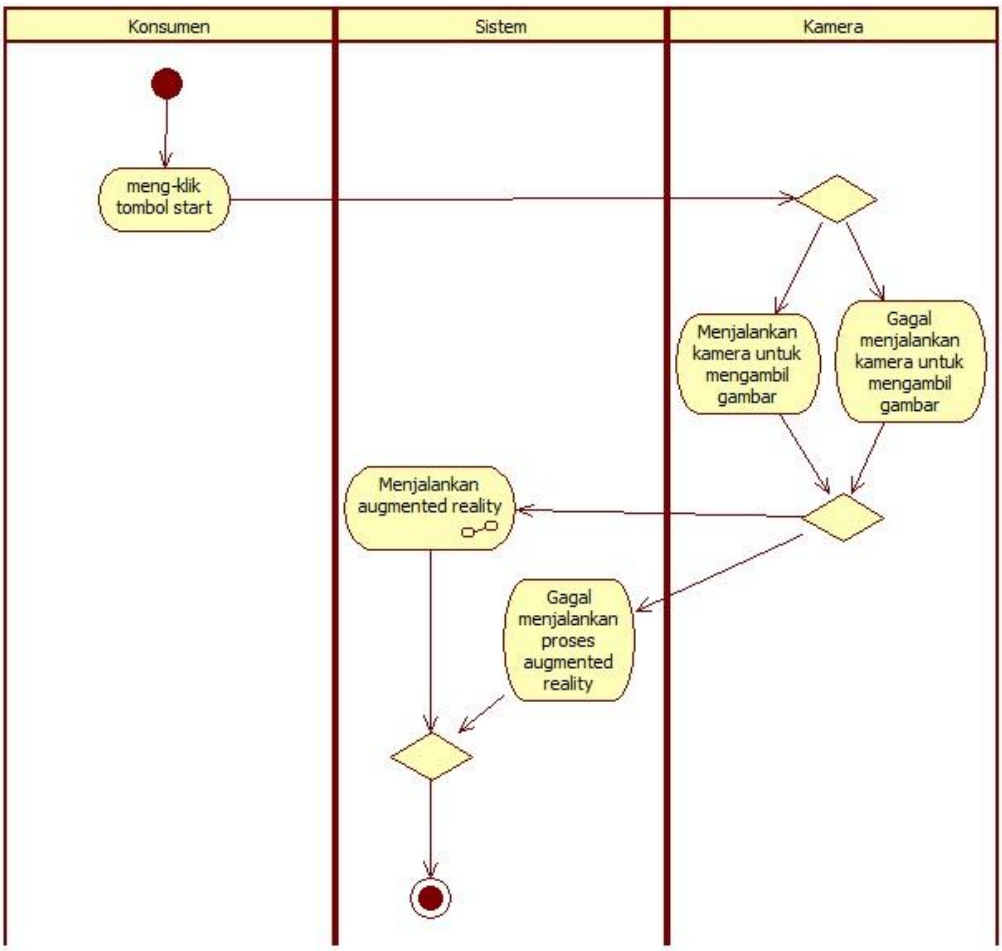
Gambar 2. 4. *Use Case Diagram*

Diagram Use Case berguna dalam tiga hal :

a. Menjelaskan fasilitas yang ada *(requirements*) : *Use Case* baru selalu menghasilkan fasilitas baru ketika sistem di analisa, dan design menjadi lebih jelas.
b. Komunikasi dengan klien : Penggunaan notasi dan simbol dalam diagram *Use Case* membuat pengembang lebih mudah berkomunikasi dengan klien-kliennya.
c. Membuat test dari kasus-kasus secara umum : Kumpulan dari kejadian-kejadian untuk *Use Case* bisa dilakukan test kasus layak untuk kejadian-kejadian tersebut.

2. Activity Diagram

Pada dasarnya diagram *Activity* sering digunakan oleh *flowchart*. Diagram ini berhubungan dengan diagram *Statechart*. Diagram *Statechart* berfokus pada obyek yang dalam suatu proses (atau proses menjadi suatu obyek), diagram *Activity* berfokus pada aktifitas-aktifitas yang terjadi yang terkait dalam suatu proses tunggal. Jadi dengan kata lain, diagram ini menunjukkan bagaimana aktifitas-aktifitas tersebut bergantung satu sama lain. Sebagai contoh, perhatikan proses yang terjadi. "Pengambilan uang dari bank melalui ATM." Ada tiga aktifitas kelas (orang, dan lainnya) yang terkait yaitu : *Customer*, ATM, and Bank. Proses berawal dari lingkaran *start* hitam pada bagian atas dan berakhir di pusat lingkaran *stop* hitam/putih pada bagian bawah. Aktivitas digambarkan dalam bentuk kotak persegi. Lihat gambar di bawah ini, agar lebih jelas :

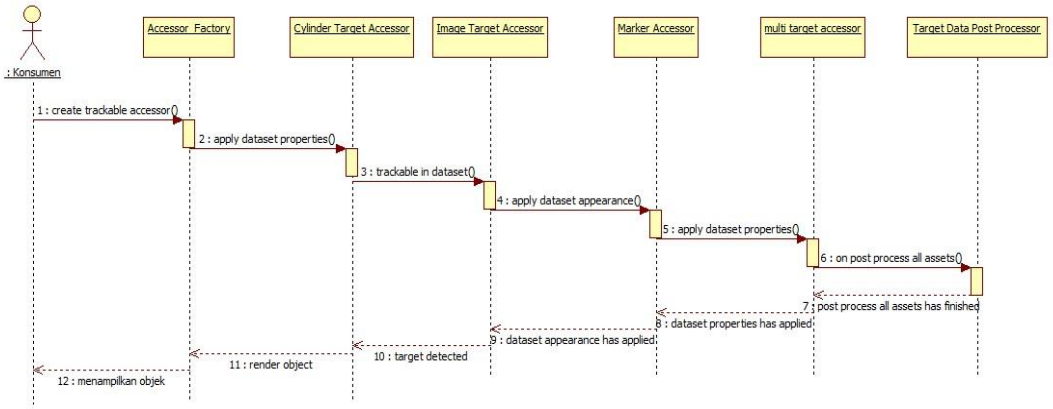


Gambar 2. 5. *Activity Diagram*

Diagram Activity dapat dibagi menjadi beberapa jalur kelompok yang menunjukkan obyek mana yang bertanggung jawab untuk suatu aktifitas. Peralihan tunggal (*single transition*) timbul dari setiap adanya *activity* (aktifitas), yang saling menghubungi pada aktifitas berikutnya. Sebuah *transition* (transisi) dapat membuat cabang ke dua atau lebih percabangan *exclusive transition* (transisi eksklusif). Label *Guard Expression* (ada didalam [ ]) yang menerangkan *output* (keluaran) dari percabangan. Percabangan akan menghasilkan bentuk menyerupai bentuk intan. T*ransition* bisa bercabang menjadi beberapa aktifitas paralel yang disebut *Fork***.** *Fork* beserta *join* (gabungan dari hasil *output fork*) dalam diagram berbentuk *solid bar* (batang penuh).

3. Sequence Diagram

Diagram Class dan diagram *Object* merupakan suatu gambaran *model statis*. Namun ada juga yang bersifat *dinamis*, seperti Diagram *Interaction*. Diagram *sequence* merupakan salah satu *diagram Interaction* yang menjelaskan bagaimana suatu operasi itu dilakukan; *message* (pesan) apa yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu. Obyek-obyek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut. Di bawah ini adalah diagram Sequence untuk pembuatan Hotel *Reservation*. Obyek yang mengawali urutan *message* adalah '*aReservation Window'*.



Gambar 2. 6. *Sequence Diagram*

Pemanggilan diri sendiri disebut dengan iterasi. *Expression* yeng dikurung dengan "[ ]", adalah *condition* (keadaan kondisi). Pada diagram dapat dibuat *note* (catatan). Pada gambar, terlihat seperti selembar kertas yang berisikan teks. *Note* bisa diletakan dimana saja pada diagram UML.

4. Class Diagram

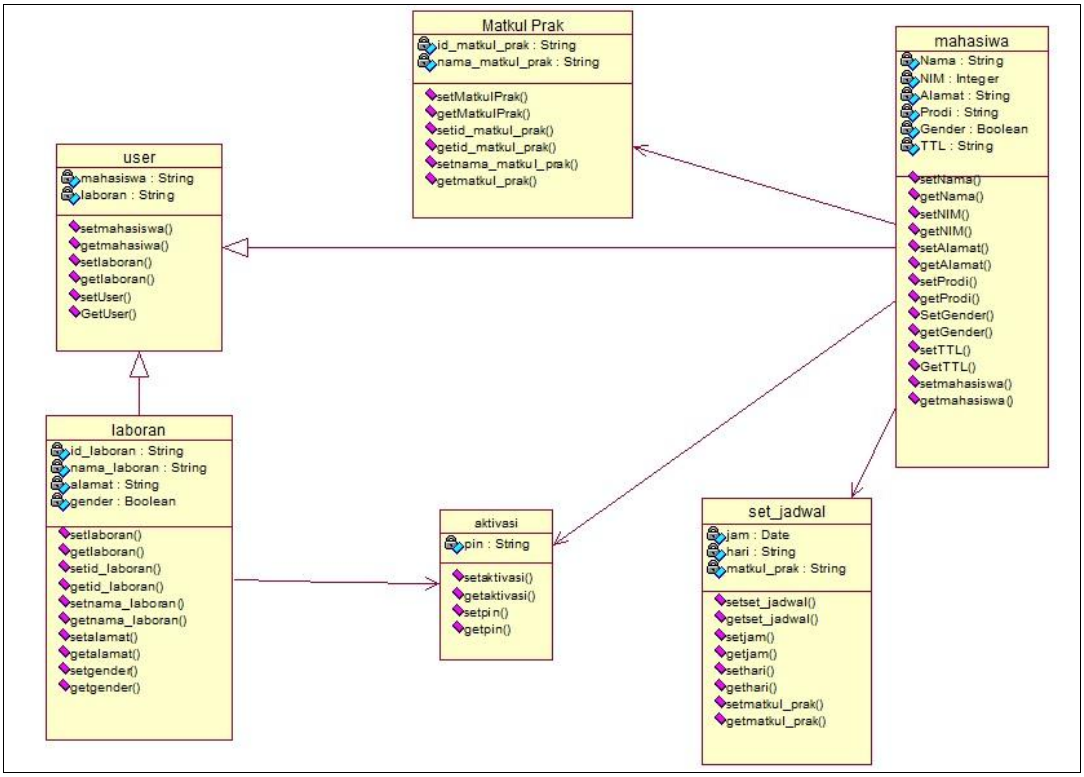
Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain b*error*ientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class* diagram menggambarkan struktur dan deskripsi *class*,

package dan objek beserta hubungan satu sama lain seperti *containment* , pewarisan, asosiasi, dan lain-lain. *Class* memiliki tiga area pokok yaitu nama (dan *stereotype*), Atribut, dan Metoda.

Atribut dan metoda dapat memiliki salah satu sifat berikut :

a. *Private*, tidak dapat dipanggil dari luar class yang bersangkutan
b. *Protected*, hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak yang mewarisinya
c. *Public*, dapat dipanggil oleh siapa saja

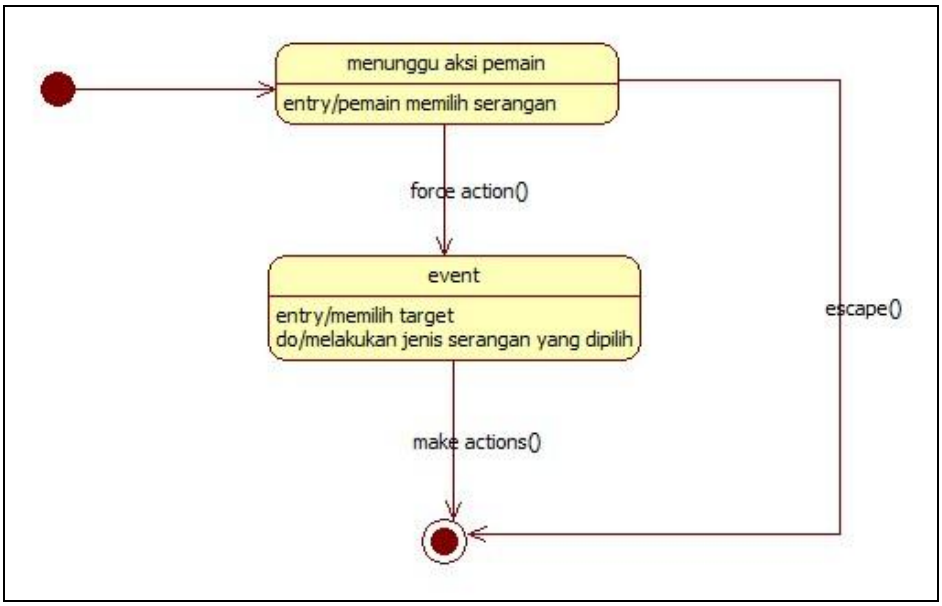
Class dapat merupakan implementasi dari sebuah interface, yaitu *class* abstrak yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class*.Dengan demikian interface mendukung resolusi metoda pada saat *run-time.* Sesuai dengan perkembangan *class model*, *class* dapat dikelompokkan menjadi *package*. Kita juga dapat membuat diagram yang terdiri atas *package*.



Gambar 2. 7. *Class Diagram*

5. *State Machine Diagram* (*Statechart* diagram in versi 1.x)

Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu *state* ke*state* lainnya) suatu objek pada sistem sebagai akibat dari *stimuli* yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*). Dalam UML, *state* digambarkan berbentuk segiempat dengan sudut membulat dan memiliki nama sesuai kondisinya saat itu. Transisi antar *state* umumnya memiliki kondisi *guard* yang merupakan syarat terjadinya transisi yang bersangkutan, dituliskan dalam kurung siku. *Action* yang dilakukan sebagai akibat dari *event* tertentu dituliskan dengan diawali garis miring. Titik awal dan akhir digambarkan berbentuk lingkaran berwarna penuh dan berwarna setengah.



Gambar 2. 8. *State Machine Diagram*

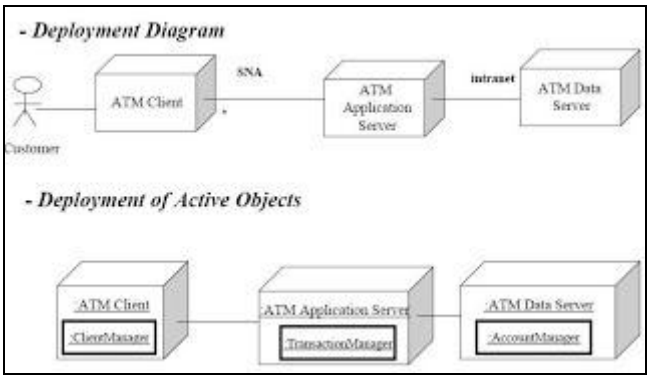
6. *Component Diagram*

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya. Komponen piranti lunak adalah modul berisi *code* , baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time, link time* , maupun *run time*. Umumnya komponen terbentuk dari beberapa *class* dan atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil.

Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

7. Deployment Diagram

Deployment/physical diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal Sebuah *node* adalah server,*workstation* , atau piranti keras lain yang digunakan untuk men- *deploy* komponen dalam lingkungan sebenarnya. Hubungan antar *node* (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini.



Gambar 2. 9. *Deployment Diagram*

Tujuan Penggunaan UML adalah sebagai berikut:

1. Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
2. Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan.
3. Memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.
4. UML bisa juga berfungsi sebagai sebuah (*blue print)* cetak biru karena sangat lengkap dan detail. Dengan cetak biru ini maka akan bias diketahui in*form*asi secara detail tentang coding program atau bahkan membaca program dan menginterpretasikan kembali ke dalam bentuk diagram (*reserve enginering*).

Dalam penelitian ini UML digunakan untuk memodelkan proses bisnis yang terjadi dan mendokumentasikan system perangkat lunak , sehingga memudahkan dalam mengembangkan sebuah rancangan sistem .

2.2.4. IOT *(Internet of Things*)

Internet of Things, atau dikenal juga dengan singkatan **IoT**, merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus-menerus. Adapun kemampuan seperti berbagi data, *remote contro*l, dan sebagainya, termasuk juga pada benda di dunia nyata. Contohnya bahan pangan, elektronik, koleksi, peralatan apa saja, termasuk benda hidup yang semuanya tersambung ke jaringan lokal dan global melalui sensor yang tertanam dan selalu aktif. Pada dasarnya, *Internet of Things* mengacu pada benda yang dapat diidentifikasikan secara unik sebagai representasi virtual dalam struktur berbasis Internet. Istilah *Internet of Things* awalnya disarankan oleh Kevin Ashton pada tahun 1999 dan mulai terkenal melalui *Auto-ID Center* di MIT.

Casagras (*Coordination and support action for global RFID-related activities and standardisation*) mendefinisikan *Internet of Things*, sebagai sebuah infrastruktur jaringan global, yang menghubungkan benda-benda fisik dan virtual melalui eksploitasi data *capture* dan kemampuan komunikasi. Infrastruktur terdiri dari jaringan yang telah ada dan internet berikut pengembangan jaringannya. Semua ini akan menawarkan identifikasi obyek, sensor dan kemampuan koneksi sebagai dasar untuk pengembangan layanan dan aplikasi ko-operatif yang independen. Ia juga ditandai dengan tingkat otonom data capture yang tinggi, *event transfer*, konektivitas jaringan dan *interoperabilitas*.

SAP (*Systeme, Anwendungen und Produkte*) mendefinisikannya sebagai berikut: Dunia di mana benda-benda fisik diintegrasikan ke dalam jaringan in*form*asi secara berkesinambungan, dan di mana benda-benda fisik tersebut berperan aktif dalam proses bisnis. Layanan yang tersedia berinteraksi dengan ‘obyek pintar’ melalui internet, mencari dan mengubah status mereka sesuai

dengan setiap in*form*asi yang dikaitkan, disamping memperhatikan masalah privasi dan keamanan.

ETP EPOSS mendefinisikannya sebagai Jaringan yang dibentuk oleh hal-hal atau benda yang memiliki identitas, pada dunia maya yang beroperasi di ruang itu dengan menggunakan kecerdasan antarmuka untuk terhubung dan berkomunikasi dengan pengguna, konteks sosial dan lingkungan.

Metode yang digunakan oleh *Internet of Things* adalah *nirkabel* atau pengendalian secara otomatis tanpa mengenal jarak. Pengimplementasian *Internet of Things* sendiri biasanya selalu mengikuti keinginan si *developer* dalam mengembangkan sebuah aplikasi yang ia ciptakan, apabila aplikasinya itu diciptakan guna membantu monitoring sebuah ruangan maka pengimplementasian *Internet of Things* itu sendiri harus mengikuti alur diagram pemrograman mengenai sensor dalam sebuah rumah, berapa jauh jarak agar ruangan dapat dikontrol, dan kecepatan jaringan internet yang digunakan. Perkembangan teknologi jaringan dan Internet seperti hadirnya *IPv6*, 4G, dan Wimax, dapat membantu pengimplementasian *Internet of Things* menjadi lebih optimal, dan memungkinkan jarak yang dapat di lewati menjadi semakin jauh, sehingga semakin memudahkan kita dalam mengontrol sesuatu[2].

Dalam penelitan ini konsep IOT digunakan untuk mempermudah peternak dalam mengontrol suhu serta pakan dan minum ayam dengan menggunakan *Wifi* . Dengan IOT peternak dapat mengontrol secara *real time* berapa suhu kandang dan mengetahui apakah pakan dan minum sudah terisi.

2.2.5. Wireless Fidelity (Wifi)

Wifi merupakan kependekan dari *Wireless Fidelity* yaitu sebuah media penghantar komunikasi data tanpa kabel yang bisa digunakan untuk komunikasi atau men*transfer* program dan data dengan kemampuan yang sangat cepat karena media penghantar menggunakan sinyal radio yang bekerja pada frekuensi tertentu.

Istilah wifi banyak dikenal oleh masyarakat sebagai media untuk internet saja, namun sebenarnya bisa juga difungsikan sebagai jaringan tanpa kabel (*nirkabel*) seperti di perusahaan-perusahaan besar dan juga di warnet. Jaringan

nirkabel tersebut biasa diistilahkan dengan LAN (*local area network*). Sehingga antara komputer dilokasi satu bisa saling berhubungan dengan komputer lain yang letaknya berbeda.

Sedangkan untuk penggunaan internet, wifi memerlukan sebuah titik akses yang biasa disebut dengan hotspot untuk menghubungkan dan mengontrol antara pengguna wifi dengan jaringan internet pusat. Sebuah *hotspot* pada umumnya dilengkapi dengan *password* yang bisa meminimalisasi siapa saja yang bisa menggunakan fasilitas tersebut. Ini sering digunakan oleh pengguna rumahan, restoran, swalayan, kafe dan hotel. Namun ada juga *hotspot* yang tidak diberi *password,* sehingga siapa saja boleh menggunakan fasilitas tersebut. Misalnya taman hiburan dan stadion. Sebagai contoh di Surabaya yaitu taman Sulawesi.

Untuk masalah frekuensi kerja dari sebuah berdasarkan standard dari IEEE (*Electrical and Electronis Engineers*), itu dibagi menjadi empat bagian yaitu :

- 802.11b 11 Mb/s 2.4 GHz B
- 802.11a 54 Mb/s 5 GHz A
- 802.11g 54 Mb/s 2.4 GHz b, g
- 802.11n 100 Mb/s 2.4 GHz b, g, n

Jenis “b” merupakan produk Wi-Fi yang pertama. Sedangkan tipe “g” dan “n” adalah salah satu produk yang memiliki penjualan paling banyak diminati masyarakat pada sekitar tahun 2005.

Berdasarkan bentuk fisiknya, wi-fi dibedakan menjadi dua yaitu :

- Bentuk PCI
- Bentuk USB

Jenis PCI biasanya digunakan pada sebuah PC (*personal computer*), sedangkan untuk jenis usb penggunaannya lebih portabel bisa untuk laptop ataupun PC. Hal ini dikarenakan didesain dengan jenis colokan USB. Sehingga lebih memudahkan pengguna.

Kelebihan dari menggunakan wifi yaitu lebih *portable*, artinya kita tidak repot dengan memikirkan kabel penghubung ke *hotspot*. Dan juga, akses transfer data lebih cepat (ini bisa diperoleh jika peralatan kita masih dalam jangkauan pusat

hotspot. Namun ada juga kelemahan dari wi-fi ini, yaitu sering di *hack* (dibobol) oleh orang lain.

2.2.6. Aplikasi *Mobile*

Aplikasi adalah program yang digunakan orang untuk melakukan sesuatu pada sistem komputer. *Mobile* dapat diartikan sebagai perpindahan yang mudah dari satu tempat ke tempat yang lain, misalnya telepon *mobile* berarti bahwa terminal telepon yang dapat berpindah dengan mudah dari satu tempat ke tempat lain tanpa terjadi pemutusan atau terputusnya komunikasi. Sistem aplikasi *mobile* merupakan aplikasi yang dapat digunakan walaupun pengguna berpindah dengan mudah dari satu tempat ketempat lain lain tanpa terjadi pemutusan atau terputusnya komunikasi. Aplikasi ini dapat diakses melalui perangkat nirkabel seperti *pager*, seperti telepon seluler dan PDA. Adapun karakteristik perangkat *mobile* yaitu[3]:

1. Ukuran yang kecil

   Perangkat *mobile* memiliki ukuran yang kecil. Konsumen menginginkan perangkat yang terkecil untuk kenyamanan dan mobilitas mereka.
2. Memory yang terbatas

   Perangkat *mobile* juga memiliki memory yang kecil, yaitu primary (RAM) dan secondary (disk).
3. Daya proses yang terbatas

   Sistem *mobile* tidaklah setangguh rekan mereka yaitu desktop.
4. Mengkonsumsi daya yang rendah

   Perangkat *mobile* menghabiskan sedikit daya dibandingkan dengan mesin desktop.
5. Kuat dan dapat diandalkan

   Karena perangkat *mobile* selalu dibawa kemana saja, mereka harus cukup kuat untuk menghadapi benturan-benturan, gerakan, dan sesekali tetesan-tetesan air.
6. Konektivitas yang terbatas

Perangkat *mobile* memiliki *bandwith* rendah, beberapa dari mereka bahkan tidak tersambung.

7. Masa hidup yang pendek

Perangkat-perangkat konsumen ini menyala dalam hitungan detik kebanyakan dari mereka selalu menyala.

2.2.6.1. Android

Android adalah sebuah sistem operasi pada *handphone* yang bersifat terbuka dan berbasis pada sistem operasi Linux. Android bisa digunakan oleh setiap orang yang ingin menggunakannya pada perangkat mereka. Android menyediakan plat*form* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang akan digunakan untuk bermacam peranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T- Mobile, dan Nvidia. Pada saat perilisan perdana Android, 5 November 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, Google merilis kode–kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler. Berikut merupakan kelebihan dan kekurangan dari *Android* [4].

1. Kelebihan Android
   a. *Multitasking*

      Android yang mampu membuka beberapa aplikasi sekaligus tanpa harus menutup salah satunya.

   b. Kemudahan dalam Notifikasi

      Setiap ada SMS, Email, atau bahkan artikel terbaru dari RSS Reader, akan selalu ada notifikasi di *Home Screen* Ponsel Android dengan Lampu LED Indikator yang berkedip-kedip.

   c. Akses Mudah Aplikasi *Android* lewat Google Android App Market

Dengan Google Android App Market dapat mendownload berbagai aplikasi dengan gratis maupun berbayar. Ada banyak ribuan aplikasi dan games yang siap untuk didownload di ponsel *Android*.

d. Pilihan Ponsel yang beranekaragam

Android tersedia di ponsel dari berbagai produsen, mulai dari Sony Ericsson, Motorola, HTC sampai Samsung.Setiap pabrikan ponsel pun menghadirkan ponsel *Android* dengan gaya masing-masing.

e. Bisa menginstal ROM yang dimodifikasi

Banyak *Costum* ROM yang bisa dipakai di ponsel *Android* guna merubah tampilan.

f. Widget

Dengan adanya *Widget* di *homescreen* pengguna bisa dengan mudah mengakses berbagai setting dengan cepat dan mudah.

g. Google Maniac

Kelebihan *Android* lainnya jika pengguna setia layanan Google mulai dari Gmail sampai Google Reader, ponsel Android telah terintegrasi dengan layanan Google, sehingga dapat dengan cepat mengecek email dari Gmail.

2. Kelemahan *Android*

a. Koneksi Internet yang terus menerus

Ponsel berbasis sistem ini memerlukan koneksi internet yang simultan/terus menerus aktif. Koneksi internet GPRS selalu aktif setiap waktu, dimana pengguna harus siap berlangganan paket GPRS yang sesuai dengan kebutuhan.

b. Iklan

Aplikasi di Ponsel *Android* memang bisa didapatkan dengan mudah dan gratis, namun konsekuensinya di setiap aplikasi tersebut, akan selalu iklan yang terpampang, entah itu bagian atas atau bawah aplikasi.

Dalam Aplikasi ini *android* digunakan untuk menampilkan mengatur dan menampilkan in*form*asi tentang suhu serta pakan dan minum ayam.

2.2.7. Android Studio

Android Studio adalah sebuah IDE untuk pengembangan aplikasi di *platform Android*. Saat ini usia Android Studio masih tergolong muda, baru versi 0.2.3 (masih *early access preView*). SDK sebelumnya di-*bundle* bersama dengan Eclipse, sementara Android Studio menggunakan IntelliJ IDEA Community Edition. Beberapa pendukung IntelliJ IDEA sering mengatakan bahwa Eclipse terlalu rumit bagi pemula. Android Studio menggunakan *Gradle* untuk manajemen proyeknya. *Gradle* adalah *build automation tool* yang dapat dikonfigurasi melalui DSL berbasis Groovy. Ini yang membedakan *Gradle* dari *Ant* atau *Maven* yang memakai XML. Penggunaan DSL berbasis Groovy menyebabkan Gradle lebih fleksibel dan dapat diprogram dengan mudah.

Android Studio *build system* digunakan untuk membangun, test, menjalankan, dan membuat paket dari aplikasi yang dibangun. Build system tidak tergantung (*independent*) dari Android Studio, jadi *developer* dapat memanggilnya dalam Android Studio atau dengan menggunakan *command line*. Setelah menuliskan code pada aplikasi yang akan dibangun, *developer* dapat menggunakan *fitur build system* untuk:

1. Kostumisasi, konfigurasi, dan meng-*extends* proses *build*
2. Membuat beberapa apk untuk aplikasi android dengan fitur yang berbeda menggunakan project yang sama
3. Menggunakan ulang *code* dan *resources*

Fleksibilitas dari system pengembangan Android Studio memungkinkan untuk mencapai keseluruhan hal ini tanpa harus memodifikasi file inti dari *project*. Android Studio ini adalah lingkungan pengembangan baru dan terintegrasi penuh, yang baru saja dirilis oleh Google untuk sistem operasi *Android*. Android Studio dirancang untuk menjadi peralatan baru dalam pengembangan aplikasi dan juga memberi alternative lain selain Eclipse yang saat ini menjadi IDE yang paling dipakai. Saat memulai proyek baru dengan Android

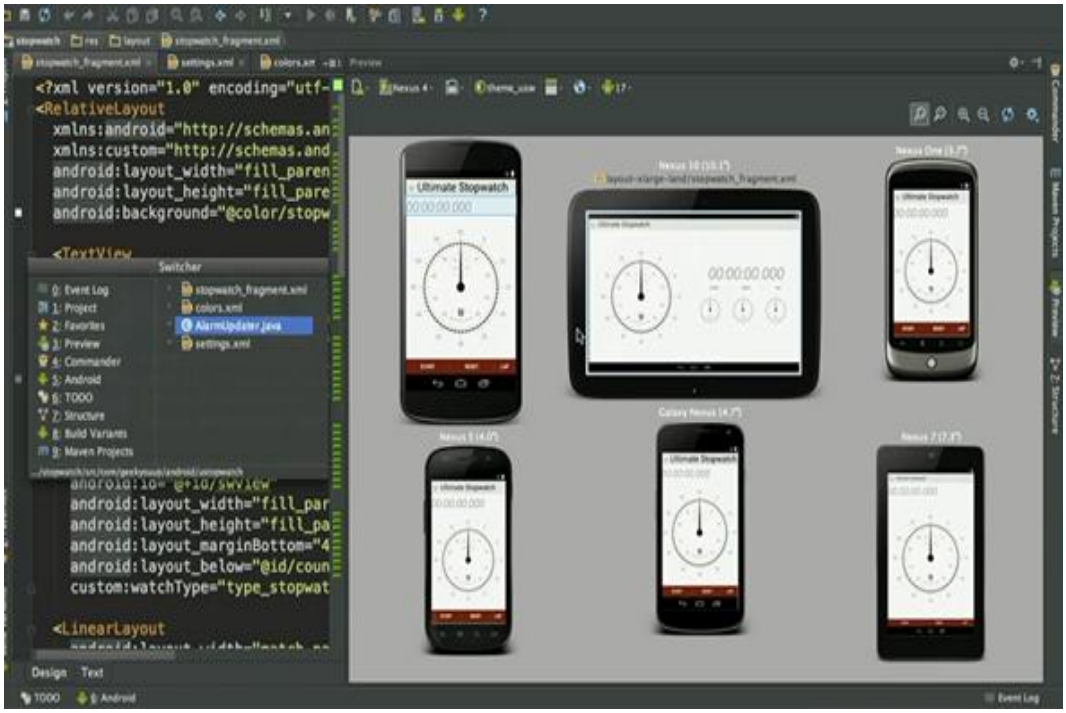
Studio, struktur proyek akan muncul bersama dengan hampir semua berkas yang ada di dalam direktori SDK, peralihan ke sistem manajemen berbasis Gradle ini memberikan fleksibilitas yang lebih besar pada proses pembangunannya.

Android Studio developer dapat melihat perubahan visual apapun yang dilakukan pada aplikasi secara langsung.Dapat terlihat perbedaannya jika dipasang pada beberapa perangkat Android berbeda, termasuk konfigurasi dan resolusi secara bersamaan. Fitur lain di Android Studio adalah alat alat baru untuk *packing* dan memberi label kode. Dengan begitu mengijinkan *developer* tetap menjadi yang teratas ketika berurusan dengan banyak kode. Program ini juga menggunakan sistem *drag and drop* untuk memindahkan komponen melalui *interface User*.

Selain itu, lingkungan baru ini juga mendukung *Google Cloud Messaging*. Sebuah fitur yang mengijinkan untuk mengirim data dari server ke perangkat *Android* melalui *cloud*, cara terbaik untuk mengirim *Reminder* pada apps. Program ini juga membantu untuk melokalisasi aplikasi anda, memberi gambaran visual untuk tetap memprogram sambil mengontrol alur dari aplikasi. Kemudahan yang diberikan oleh Android Studio adalah sebagai berikut :

1. Lingkungan pengembangan yang mantap dan bersifat langsung.
2. Cara termudah untuk menguji per*forma* pada perangkat dengan tipe lain.
3. Wizard dan template berisi elemen – elemen umum yang ada di semua pemrograman Android.
4. Editor dengan fitur lengkap dengan banyak peralatan ekstra untuk mempercepat pengembangan aplikasi anda. Layout editor yang memungkinkan untuk *drag and drop* komponen UI, pratinjau *layout* pada beberapa konfigurasi layar, dan banyak lagi.
5. Berbasis Gradle.
6. Android spesifik refactoring dan perbaikan yang cepat.
7. Alat Lint untuk menangkap kinerja, kegunaan, versi kompatibilitas dan masalah lainnya.
8. ProGruard dan app-*signature*.

9. Built-in dukungan untuk *Google Cloud platform*, sehingga mudah untuk mengintegrasikan *Google Cloud Messaging* dan *App Engine* sebagai komponen *server-side*.



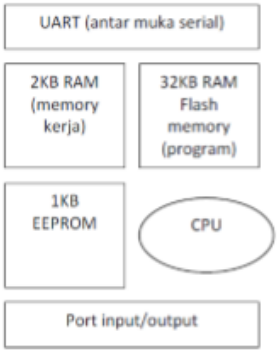
Gambar 2. 10. *Android Studio*

Dalam Penelitian ini android studio digunakan untuk membangun aplikasi android dibuat .

2.2.8. Arduino

Arduino adalah *platform* pembuatan prototipe elektronik yang bersifat opensource hardware yang berdasarkan pada perangkat keras dan perangkat lunak yang fleksibel dan mudah digunakan.Arduino pada awalnya dikembangkan di Ivrea, Italia. Plat*form* arduino terdiri dari arduino *board*, *shield*, bahasa pemrograman arduino, dan arduino *development environment*. Arduino *board* biasanya memiliki sebuah chip dasar *mikrokontroler* Atmel AVR ATmega8 berikut turunannya. Bahasa pemrograman arduino adalah bahasa pemrograman yang umum digunakan untuk membuat perangkat lunak yang ditanamkan pada arduino board. Bahasa pemrograman arduino mirip dengan bahasa pemrograman C++.

Arduino berfungsi sebagai pengatur kontrol alat. Arduino menjalankan perintah sesuai program yang diberikan dan memproses banyak data dari inputan sensor. Gambarn Blok pada Arduino dapat dilihat dalam gambar 2.11



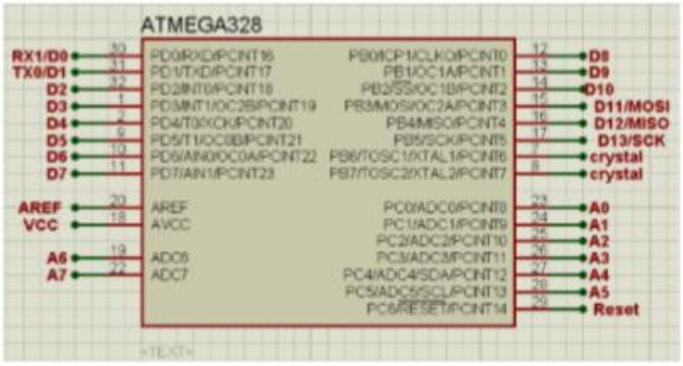
Gambar 2. 11. Gambaran blok pada Arduino

Gambar 2.11 merupakan gambaran blok yang ada pada Arduino. Terdiri dari blok memori, blok komunikasi, blok I/O dan blok prossesor. Perangkat masukan dan keluaran atau sering dikenal I/O mikrokontroler adalah suatu perangkat yang menghubungkan mikrokontroler dengan dunia luar (rangkaian lain), perangkat ini dibutuhkan sebagai media komunikasi dengan perangkat lain atau perubah tipe sinyal. Beberapa perangkat I/O yang dimiliki ATmega328 antara lain:

1. UART (*Universal Asynchronous Receiver Transmitter*) digunakan sebagai komunikasi serial.
2. SPI (*Serial Peripheral Interface*) merupakan port komunikasi serial sinkron.
3. 12C bus (*Intergrated Circuit bus*) merupakan antamuka serial bus yang dikembangkan oleh philips.
4. *Analog to Digital Conversion* (ADC) adalah rangkaian yang digunakan untuk mengubah data analog ke data digital.

2.2.8.1. Konfigurasi Pin Arduino

Konfigurasi Pin Arduino dibagi menjadi beberapa bagian antara lain untuk jalur komunikasi , mengatur kecepatan kerja alat. Keterangan pin pada arduino pro mini dapat dilihat dalam gambar 2.12 Konfigurasi Pin Arduino



Gambar 2. 12. Konfigurasi Pin Arduino

Keterangan:

1. Pin D0 sampai D13 merupakan Pin I/O yang ada pada Arduino *Pro Mini*. Pin - pin tersebut merupakan pin I/0 *digital* (*high - low*). Artinya pembacaan yang dapat dilakukan oleh pin tersebut 0 *volt* atau 5 *volt*. Pin - pin digital juga dapat berfungsi sebagai jal ur komunikasi UART, I2C, dan SPI.
2. Pin A0 sampai A7 merupakan Pin Analog. Pin - pin ini membaca inputan berupa data analog. Data analog yang dapat dibaca oleh pin tersebut antara 0 sampai 1023. Nilai 0 sampai 1023 tersebut mewakili tegangan 0 sampai 5 *volt*.
3. VREF merupakan pin tegangan referensi pada Arduino.
4. VCC merupakan pin untuk tegangan yang masuk.
5. *Crystal* merupakan pin untuk mengatur kecepatan kerja alat.
6. *Reset* merupakan pin untuk memulai program dari awal.

Dalam penelitian ini arduino digunakan untuk pengontrolan suhu, pakan dan minum.

2.2.9. Sensor

Sensor adalah suatu alat yang digunakan untuk mengukur atau mendeteksi kejadian alam seperti panas, asap, gas dan mengubahnya menjadi representasi nilai digital atau analog bergantung dari jenis sensor yang digunakan.

2.2.9.1. Sensor Ultrasonik

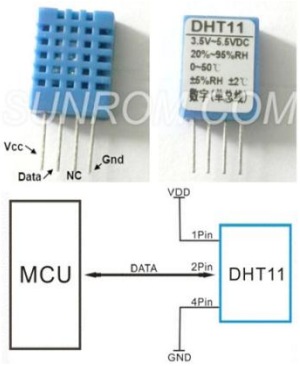
Sensor ultrasonik adalah sebuah sensor yang mengubah besaran fisis (bunyi) menjadibesaran listrik. Pada sensor ini gelombang ultrasonik dibangkitkan melalui sebuah benda yang disebut *piezoelektrik. Piezoelektrik* ini akan menghasilkan gelombang ultrasonik dengan frekuensi 40 kHz ketika sebuah *osilator* diterapkanpada benda tersebut. Sensor ultrasonik secara umum digunakan untuk suatu pengungkapan taks entuh yang beragam seperti aplikasi pengukuran jarak. Alat ini secara umum memancarkan gelombang suara *ultrasonik* menuju suatu target yang memantulkan balik gelombang kearah sensor. Kemudian sistem mengukur waktu yang diperlukan untuk pemancaran gelombang sampai kembali ke sensor dan menghitung jarak target dengan menggunakan kecepatan suara dalam medium. Rangkaian penyusun sensor *ultrasonik* ini terdiri dari ***transmitter***, ***reiceiver***, dan ***komparator***. Selain itu, gelombang *ultrasonik* dibangkitkan oleh sebuah kristal tipis bersifat ***piezoelektrik*.**

Dalam penelitian ini sensor *ultrasonik* digunakan untuk mengukur ketinggian air minum dan ketinggian pakan.

2.2.9.2. Sensor Suhu DHT11

DHT11 adalah sensor digital yang dapat mengukur suhu dan kelembaban udara di sekitarnya. Sensor ini sangat mudah digunakan bersama dengan Arduino. Memiliki tingkat stabilitas yang sangat baik serta fitur kalibrasi yang sangat akurat. Koefisien kalibrasi disimpan dalam OTP program memori, sehingga ketika internal sensor mendeteksi sesuatu, maka module ini menyertakan koefisien tersebut dalam kalkulasinya, DHT11 ini termasuk sensor yang memiliki kualitas terbaik, dinilai dari respon, pembacaan data yang cepat, dan kemampuan anti-interference. Ukurannya yang kecil, dan dengan transmisi sinyal hingga 20 meter,dengan sepsifikasi: *Supply Voltage*: +5 V, *Temperature range* : 0-50 °C *error of* ± 2 °C, *Humidity* : 20-90% RH ± 5% RH *error*,dengan sesifikasi *digital interfacing system*. membuat produk ini

cocok digunakan untuk banyak aplikasi-aplikasi pengukuran suhu dan kelembaban.



Gambar 2. 13. Sensor Kelembaban udara/Humidity (DHT11)

Tabel 2. 1. Tabel karakteristik sensor kelembaban udara/*Humidity*

Model	DHT11
Power supply	3-5.5V DC
Output signal	*digital signal via single-bus*
Measuring range	*humidity* 20-90% RH ± 5% RH *error temperature* 0-50 °C *error of* ± 2 °C
Accuracy	*humidity* +-4%RH (Max +-5%RH); *temperature* +-2.*0Celsius*
Resolution or sensitivity	*humidity* 1%RH; *temperature* 0.1Celsius
Repeatability	*humidity* +-1%RH; *temperature* +-1Celsius
Humidity hysteresis	+-1%RH
Long-term Stability	+-0.5%RH/*year*
Sensing period	*Average*: 2s
Interchangeability	*fully interchangeable*
Dimensions size	12*15.5*5.5mm

Dari penjelasan (Tabel 2.1) diatas bahwa struktur yang merupakan cara kerja dari sensor kelembaban udara/*Humidity DHT11* memiliki empat buah kaki yaitu: pada bagian kaki(V_{CC}), dihubungkan ke bagian Vss yg bernilai sebesar 5V,pada board arduino uno dan untuk bagian kaki *GND* dihubungkan ke *ground (GND)* pada *board arduino uno*, sedangkan pada bagian kaki data yang merupakan keluaran *(Output)* dari hasil pengolahan data analog dari *sensor DHT11* yang dihubungkan ke bagian *analog input (pin3),* yaitu pada bagian pin *PWM (Pulse Width Modulation)* pada *board arduino uno* dan yang tak ketinggalan terdapat satu kaki tambahan yaitu kaki *NC (Not Connected),* yang tidak dihubungkan ke pin manapun. Sensor kelembaban lain yang banyak dikembangkan adalah jenis sensor serat optik yang menggunakan serat optik sebagai bahan sensor. Berbagai metode dan bahan untuk sensor telah dikembangkan pada sensor serat optik ini.

Dalam penelitian ini sensor suhu DHT11 digunakan untuk mengukur suhu kandang ayam.

2.2.10. Lampu *Dimmer*

Rangkaian *dimmer* lampu pijar berfungsi untuk mengatur tingkat intensitas cahaya penerangan lampu pijar. Rangkaian ini bisa diatur mulai dari yang redup hingga ke remang-remang sampai ke nyala lampu yang terang. Dan juga bisa membuat rangkaian dimmer pengatur nyala lampu dengan pola sederhana. Di dalam rangkaian dimmer ini, terdapat 3 komponen penting guna mengatur kerja dimmer ini. Komponen TRIAC berfungsi untuk mengatur besaran tegangan AC yang masuk ke perangkat lampu ini. Sementara komponen DIAC dan VR berfungsi untuk mengatur bias TRIAC guna menentukan titik on dan off pada komponen TRIAC ini. Daya *output* rangkaian dimmer ini dapat digunalan untuk mengendalikan intensitas cahaya lampu pijar dengan daya 5 *Watt*. Pada rangkaian ini potensiometer berfungsi sebagai sensor mekanis pengatur besar kecilnya lampu atau transd*User* pasif yang perlu mendapatkan daya dari luar.

Beberapa komponen yang dipakai dalam rangkaian dimmer lampu adalah sebagai berikut :

1. Potensiometer 220K
2. TRIAC BT138
3. DIAC BR100
4. Resistor 1K
5. Kapasitor 100nF/500V
6. Lampu Pijar
7. Steker

Potensiometer adalah resistor tiga terminal dengan sambungan geser yang membentuk pembagi tegangan dapat disetel. Jika hanya dua terminal yang digunakan (salah satu terminal tetap dan terminal geser), potensiometer berperan sebagai resistor variabel atau *Rheostat*. Potensiometer yang dioperasikan oleh suatu mekanisme dapat digunakan sebagai transd*User*, misalnya sebagai sensor pada *dimmer* lampu.

TRIAC atau *Triode for Alternating Current* (Trioda untuk arus bolak-balik) adalah sebuah komponen elektronik yang kira-kira ekivalen dengan dua SCR yang disambungkan antiparalel dan kaki gerbangnya disambungkan bersama. Nama lain TRIAC adalah Bidirectional Triode Thyristor. Ini menunjukkan saklar dua arah yang dapat mengalirkan arus listrik ke kedua arah ketika dipicu (dihidupkan). Ini dapat dipicu baik dengan tegangan positif atau pun negatif pada elektrode gerbang. Sekali dipicu, komponen ini akan terus menghantarkan sampai arus yang mengalir lebih rendah dari arus genggamnya, misal pada akhir paruh siklus dari arus bolak-balik. Hal tersebut membuat TRIAC sangat cocok untuk mengendalikan tegangan AC, memungkinkan pengendalian arus yang sangat tinggi dengan arus kendali yang sangat rendah.

Dalam Penelitian ini digunakan untuk menerangkan dan meredupkan lampu sehingga dapat menyesuaikan dengan suhu kandang.

2.2.11. Web

Website merupakan halaman situs sistem in*form*asi yang dapat diakses secara cepat. Website ini didasari dari adanya perkembangan teknologi in*form*asi dan komunikasi. Melalui perkembangan teknologi in*form*asi, tercipta suatu

jaringan antar komputer yang saling berkaitan. Jaringan yang dikenal dengan istilah internet secara terus-menerus menjadi pesan–pesan elektronik, termasuk *e-mail*, transmisi file, dan komunikasi dua arah antar individu atau komputer.

Dalam pembangunan web ini hal yang dibutuhkan adalah PHP,Mysql,Javascript, HTML dan JSON untuk pertukaran data.

2.2.11.1. Web Service

Web service adalah aplikasi sekumpulan data (*database*), perangkat lunak (*software*) atau bagian dari perangkat lunak yang dapat diakses secara remote oleh berbagai piranti dengan sebuah perantara tertentu. Secara umum,*web service* dapat diidentifikasikan dengan menggunakan URL seperti hanya web pada umumnya. Namun yang membedakan *web service* dengan web pada umumnya adalah interaksi yang diberikan oleh *web service*. Berbeda dengan URL *web* pada umumnya, URL *web service* hanya menggandung kumpulan in*form*asi, perintah, konfigurasi atau sintaks yang berguna membangun sebuah fungsi-fungsi tertentu dari aplikasi[5].

Web service dapat diartikan juga sebuah metode pertukaran data, tanpa memperhatikan dimana sebuah*database* ditanamkan, dibuat dalam bahasa apa sebuah aplikasi yang mengkonsumsi data, dan di plat*form* apa sebuah data itu dikonsumsi. *Web service* mampu menunjang interoperabilitas. Sehingga *web service* mampu menjadi sebuah jembatan penghubung antara berbagai sistem yang ada.

Menurut W3C *Web services Architecture Working Group* pengertian *Web service* adalah sebuah sistem *software* yang di desain untuk mendukung interoperabilitas interaksi mesin ke mesin melalui sebuah jaringan. *Interface web service* dideskripsikan dengan menggunakan *form*at yang mampu diproses oleh mesin (khususnya WSDL). Sistem lain yang akan berinteraksi dengan *web service* hanya memerlukan SOAP, yang biasanya disampaikan dengan HTTP dan XML sehingga mempunyai korelasi dengan standar Web (Web Services Architecture Working Group, 2004).

Web pada umumnya digunakan untuk melakukan *respon* dan *request* yang dilakukan antara *client* dan *server*. Sebagai contoh, seorang pengguna

layanan *web* tertentu mengetikan alamat *url web* untuk membentuk sebuah *request*. *Request* akan sampai pada *server*, diolah dan kemudian disajikan dalam bentuk sebuah respon. Dengan singkat kata terjadilah hubungan *client-server* secara sederhana.

Sedangkan pada *web service* hubungan antara *client* dan *server* tidak terjadi secara langsung. Hubungan antara *client* dan *server* dijembatani oleh file *web service* dalam *form*at tertentu. Sehingga akses terhadap *database* akan ditanggani tidak secara langsung oleh *server*, melainkan melalui perantara yang disebut sebagai *web service*. Peran dari *web service* ini akan mempermudah distribusi sekaligus integrasi database yang tersebar di beberapa *server* sekaligus.

2.2.12. *JavaScipt Object Notation* (JSON)

JavaScript Object Notation (JSON) adalah *form*at pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. JSON merupakan *form*at teks yang tidak bergantung pada bahasa pemprograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data.JSON terbuat dari dua struktur:

1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau *associative array*.
2. Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemprograman moderen mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena *form*at data mudah dipertukarkan dengan bahasa-bahasa pemrograman

yang juga berdasarkan pada struktur data ini. JSON menggunakan bentuk sebagai berikut:

1. Objek

Objek adalah sepasang nama/nilai yang tidak terurutkan. Objek dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap nama diikuti dengan : (titik dua)dan setiap pasangan nama/nilai dipisahkan oleh , (koma).

2. Larik

Larik adalah kumpulan nilai yang terurutkan. Larik dimulai dengan [(kurung kotak buka) dan diakhiri dengan] (kurung kotak tutup). Setiap nilai dipisahkan oleh , (koma).

3. Nilai

Nilai (*value*) dapat berupa sebuah string dalam tanda kutip ganda, atau angka, atau true atau false atau null, atau sebuah *objek* atau sebuah larik. Struktur - struktur tersebut dapat disusun bertingkat.

4. String

String adalah kumpulan dari nol atau lebih karakter *Unicode*, yang dibungkus dengan tanda kutip ganda. Di dalam string dapat digunakan *backslash escapes* "\" untuk membentuk karakter khusus. Sebuah karakter mewakili karakter tunggal pada string. String sangat mirip dengan string C atau Java.

5. Angka

Angka adalah sangat mirip dengan angka di C atau Java, kecuali *form*at oktal dan heksadesimal tidak digunakan.

6. Spasi kosong (*whitespace*)

Spasi kosong dapat disisipkan di antara pasangan tanda-tanda tersebut, kecuali beberapa detil *encoding* yang secara lengkap dipaparkan oleh bahasa pemprograman yang bersangkutan.

2.2.13. PHP

PHP (*Hypertext Prepocessor)*, merupakan bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah situs web dan bisa digunakan bersamaan dengan HTML. PHP diciptakan oleh

Rasmus Lerdorf pertama kali tahun 1994. Pada awalnya PHP adalah singkatan dari "*Personal Home Page Tools*". Selanjutnya diganti menjadi FI ("*Form*s Interpreter"). Sejak versi 3.0, nama bahasa ini diubah menjadi "PHP: *Hypertext Prepocessor*" dengan singkatannya "PHP". PHP versi terbaru adalah versi ke-5. Berdasarkan *survey* Netcraft pada bulan Desember 1999, lebih dari sejuta site menggunakan PHP, di antaranya adalah NASA, Mitsubishi, dan RedHat.

PHP adalah bahasa pemrograman script server-side yang didesain untuk pengembangan web, tetapi juga bisa digunakan sebagai bahasa pemrograman umum *(wikipedia). PHP* pertama kali di kembangkan pada tahun 1995 oleh *Rasmus Lerdorf*, namun sekarang dikelola oleh *The PHP Group*. Situs resmi PHP beralamat di *http://www.php.net*. Pada awalnya PHP adalah singkatan dari *Personal Home Page*, namun karena dalam perkembangannya PHP tidak hanya digunakan untuk membuat halaman web pribadi, PHP saat ini merupakan singkatan dari PHP: *Hypertext Preprocessor*, sebuah kepanjangan rekursif, yakni permainan kata dimana kepanjangannya berisi juga singkatan itu sendiri.

Sistem kerja dari PHP diawali dengan permintaan yang beasal dari halaman website oleh *browser*. Berdasarkan URL atau alamat website dalam jaringan internet, *browser* akan menemukan sebuah alamat dari webserver, mengidentifikasi halaman yang dikehendaki, dan menyampaikan segala in*form*asi yang dibutuhkan oleh *webserver*. Selanjutnya *webserver* akan mencarikan berkas yang diminta dan menampilkan isinya di *browser. Browser* yang mendapatkan isinya segera menerjemahkan kode HTML dan menampilkannya.

2.2.14. MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris : *database management system*) atau DBMS yang *multithread, multi-User,* dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi *GNU General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

Kehandalan suatu sistem basisdata (DBMS) dapat diketahui dari cara kerja pengoptimasi-nya dalam melakukan proses perintah-perintah *SQL* yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya. Sebagai peladen basis data, *MySQL* mendukung operasi basisdata transaksional maupun operasi basis data non-transaksional. Pada modus operasi non-transaksional, *MySQL* dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak peladen basisdata kompetitor lainnya. Namun pada modus non-transaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modus non-transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi *blogging* berbasis web , CMS, dan sejenisnya. Untuk kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modus basisdata transaksional, hanya saja sebagai konsekuensinya unjuk kerja MySQL pada modus transaksional tidak secepat unjuk kerja pada modus non-transaksional [6].

2.2.15. JavaScript

JavaScript merupakan bahasa skrip yang populer di internet dan dapat bekerja di sebagian besar penjelajah web populer seperti *Internet Explorer* (IE), Mozilla Firefox, Netscape dan Opera. Kode *JavaScript* dapat disisipkan ke dalam halaman *web* menggunakan *tag SCRIPT*. *JavaScript* adalah bahasa (pemrograman) yang hebat, sulit dan kurang begitu dipahami pada bidang bahasa pemrograman di komputer. Akan tetapi, kemampuan inti yang dimiliki oleh JavaScript untuk membuat aplikasi-aplikasi sehingga membuat perubahan dari cara pandang pemakaian Internet pada dekade ini. Salah satu contoh yang baik dari aplikasi tersebut adalah *Google Maps*[7].

Keunggulan JavaScript, yang juga dikenal sebagai ECMAScript, hadir pada semua web browser, sehingga memiliki kemampuan untuk menghasilkan keluaran yang sama pada semua *platform* yang didukung oleh *browser*. Seperti yang sudah disebutkan di atas, Google Maps, yang dapat berjalan di atas Linux, Windows dan Mac OS. Dengan pertumbuhan pustaka (*library*) javascript baru-baru ini, hal ini memudahkan untuk menavigasi dokumen, memilih elemen DOM, membuat animasi, menangani *event*, dan mengembangkan aplikasi Ajax. Berbeda

teknologi terkenal lain yang didorong oleh kepentingan komersial, JavaScript benar-benar teknologi yang *cross-platform*, bahasa pemrograman *client-side* yang bersifat bebas (untuk dimodifikasi dan gratis tentunya) dan diadopsi secara universal.

JavaScript dieksekusi pada client side (komputer pengguna): Sebuah server website mengirim javascript ke peramban milik pengguna, dan *browser* tersebut menginterpretasikan dan menjalankan kodenya. Semua ini terjadi dalam sebuah sandbox, yang menjaga agar javascript tidak menyentuh *internal* sistem, sehingga mencegah *malicious code* (kode jahat) menginfeksi komputer pengguna.

2.2.16. HTML

HyperText Markup Language (HTML) adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai in*form*asi di dalam sebuah penjelajah web Internet dan pem*form*atan hiperteks sederhana yang ditulis dalam berkas *form*at ASCII agar dapat menghasilkan tampilan wujud yang terintegerasi. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata dan disimpan dalam *form*at ASCII normal sehingga menjadi halaman web dengan perintah-perintah HTML. Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan SGML (*Standard Generalized Markup Language*), HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman web. HTML saat ini merupakan standar Internet yang didefinisikan dan dikendalikan penggunaannya oleh *World Wide Web Consortium* (W3C). HTML dibuat oleh kolaborasi Caillau TIM dengan Berners-lee Robert ketika mereka bekerja di CERN pada tahun 1989 (CERN adalah lembaga penelitian fisika energi tinggi di Jenewa).

HTML dokumen tersebut mirip dengan dokumen tulisan biasa, hanya dalam dokumen ini sebuah tulisan bisa memuat instruksi yang ditandai dengan kode atau lebih dikenal dengan TAG tertentu. Sebagai contoh jika ingin membuat tulisan ditampilkan menjadi tebal seperti: TAMPIL TEBAL, maka penulisannya dilakukan dengan cara: `< b> TAMPIL TEBAL</b>`. Tanda `< b>` digunakan untuk mengaktifkan instruksi cetak tebal, diikuti oleh tulisan yang ingin

ditebalkan, dan diakhiri dengan tanda untuk menonaktifkan cetak tebal tersebut.

HTML lebih menekankan pada penggambaran komponen-komponen struktur dan *form*at di dalam halaman web daripada menentukan penampilannya. Sedangkan penjelajah web digunakan untuk menginterpretasikan susunan halaman ke gaya built-in penjelajah web dengan menggunakan jenis tulisan, tab, warna, garis, dan perataan *text* yang dikehendaki ke komputer yang menampilkan halaman web. Salah satu hal Penting tentang eksistensi HTML adalah tersedianya *Lingua franca* (bahasa Komunikasi) antar komputer dengan kemampuan berbeda. Pengguna Macintosh tidak dapat melihat tampilan yang sama sebagaimana tampilan yang terlihat dalam pc berbasis Windows. Pengguna Microsoft Windows pun tidak akan dapat melihat tampilan yang sama sebagaimana tampilan yang terlihat pada pengguna yang menggunakan Produk-produk Sun Microsystems. namun demikian pengguna-pengguna tersebut dapat melihat semua halaman web yang telah di*form*at dan berisi Grafika dan Pranala[8].

HTML5 dan JavaScript telah menjadi sangat menarik (dan menyenangkan) dengan munculnya kerangka kerja dan lanskap mobile. PhoneGap adalah sebuah framework open source untuk cepat membangun aplikasi cross-plat*form* mobile menggunakan HTML5, JavaScript dan CSS. Prosedur pengembangan aplikasi *cross-platform* adalah sebagai bungkus aplikasi Anda dengan PhoneGap dan menyebarkan ke *platform mobile*. Membangun aplikasi untuk setiap perangkat membutuhkan kerangka kerja dan bahasa yang berbeda. PhoneGap memecahkan ini dengan menggunakan standar berbasis teknologi Web untuk menjembatani aplikasi Web dan perangkat *mobile*. Sebuah aplikasi Web tradisional hanya tidak bekerja tanpa jaringan. Salah satu solusi untuk masalah ini adalah dengan menggunakan dua fitur dari HTML5 Standar:

1. *Offline* aplikasi Web
2. Penyimpanan *database Client-side*.

Pengguna dapat menggunakan fungsi awan pada perangkat *mobile*, bekerja secara *offline* dengan aplikasi dikerahkan secara lokal pada database lokal, dan berbagi data dengan sisa awan ketika akan online lagi[9].

2.2.17. Metode Kualitatif

Metode penelitian kualitatif merupakan sebuah cara yang lebih menekankan pada aspek pemahaman secara mendalam terhadap suatu permasalahan. Penelitian kualitatif ialah penelitian riset yang bersifat deskriptif dan cenderung menggunakan analisis serta lebih menonjolkan proses dan makna. Tujuan dari metodologi ini ialah pemahaman secara lebih mendalam terhadap suatu permasalahan yang dikaji. Dan data yang dikumpulkan lebih banyak kata ataupun gambar-gambar daripada angka.

Adapun ciri pokok metode penelitian kualitatif ada lima, yaitu antara lain:

1. Menggunakan lingkungan alamiah sebagai sumber data : Sumber data yang digunakan dalam penelitian kualitatif berupa lingkungan alamiah. Kajian utama dalam penelitian kualitatif yaitu peristiwa-peristiwa yang terjadi dalam kondisi dan situasi sosial. Penelitian dilakukan ketika berinteraksi langsung di tempat kejadian. Peneliti melakukan pengamatan, mencatat, mencari tahu, menggali sumber yang berkaitan dengan peristiwa yang terjadi pada saat itu. Hasil yang diperoleh segera disusun saat itu juga. Apa yang telah diamati pada dasarnya tidak lepas dari konteks lingkungan dimana tingkahlaku itu berlangsung.
2. Memiliki sifat deskriptif analitik: Data yang diperoleh dari hasil pengamatan, wawancara, dokumentasi, analisis, catatan lapangan, disusun peneliti di lokasi penelitian, bukan dalam bentuk angka-angka. Peneliti melakukan analisis data dengan memperbanyak in*form*asi, mencari hubungannya, membandingkan, dan menemukan hasil atas dasar data sebenarnya (bukan dalam bentuk angka). Hasil analisis data berupa pemaparan yang berkenaan dengan situasi yang diteliti dan disajikan dalam bentuk uraian narasi. Pemaparan data tersebut umumnya adalah menjawab dari pertanyaan dalam rumusan masalah yang ditetapkan.

3. Tekanan pada proses bukan hasil : Data dan in*form*asi yang dibutuhkan dalam penelitian kualitatif berkaitan dengan pertanyaan untuk mengungkapkan proses dan bukan hasil dari suatu kegiatan. Pertanyaan menuntut gambaran keadaan sebenarnya tentang kegiatan, tahap-tahap, prosedur, alasan-alasan dan interaksi yang terjadi dimana dan pada saat dimana proses itu berlangsung.
4. Bersifat induktif : Penelitian kualitatif diawali mulai dari lapangan yaitu fakta empiris. Peneliti terjun langsung ke lapangan, mempelajari suatu proses penemuan yang terjadi secara alami dengan mencatat, menganalisis dan melaporkan serta menarik kesimpulan dari proses berlangsungnya penelitian tersebut. Hasil temuan penelitian dari lapangan dalam bentuk konsep, prinsip, teori dikembangkan bukan dari teori yang telah ada. Penelitian kualitatif menggunakan proses induktif artinya dari data yang terpisah-pisah namun saling berkaitan erat.
5. Mengutamakan makna : Makna yang diungkapkan berkisar pada persepsi orang mengenai suatu peristiwa yang akan diteliti tersebut. Contoh: penelitian yang dilakukan tentang peran kepala sekolah dalam pembinaan guru. Peneliti memfokuskan perhatian pada pendapat kepala sekolah tentang guru yang dibinanya, mencari in*form*asi dan pandangan kepala sekolah tentang keberhasilan dan kegagalannya membina guru, apa saja yang dialami dalam membina guru, mengapa gurunya gagal dibina, dan kenapa hal itu terjadi. Selain mencari in*form*asi kepada kepala sekolah, peneliti mencari in*form*asi dari guru sebagai bahan perbandingan supaya dapat diperoleh pandangan mengenai mutu pembinaan yang dilakukan kepala sekolah. Ketepatan in*form*asi dari partisipan diungkap oleh peneliti agar dapat menginterpretasikan hasil penelitian secara tepat dan sahih.

Berdasarkan ciri-ciri tersebut dapat disimpulkan bahwa penelitian kualitatif dimulai dari lapangan yang berdasarkan pada lingkungan alami, bukan pada teori. Data dan in*form*asi yang diperoleh dari lapangan ditarik makna dan

konsepnya, melalui pemaparan secara deskriptif analitik dan tanpa menggunakan angka, karena lebih mengutamakan prosesnya.

Dalam dunia pendidikan, penelitian kualitatif bertujuan untuk menggambarkan suatu proses kegiatan pendidikan yang didasarkan pada apa yang terjadi di lapangan sebagai bahan kajian untuk menemukan kelemahan dan kekurangannya sehingga dapat ditentukan upaya perbaikannya ;menganalisis suatu fakta, gejala dan peristiwa pendidikan yang terjadi di lapangan, menyusun hipotesis yang berkenaan dengan prinsip dan konsep pendidikan didasarkan pada data dan in*form*asi yang terjadi di lapangan.

Dalam penelitian ini digunakan metode kualitatif dalam melakukan riset penelitian.