

## **BAB 2**

### **TINJAUAN PUSTAKA**

#### **2.1. Tempat Penelitian**

Dinas Tata Ruang dan Pemukiman Kabupaten Purwakarta dibentuk berdasarkan Peraturan Daerah Kabupaten Purwakarta Nomor 9 Tahun 2016 tentang Pembentukan dan Susunan Perangkat Daerah Kabupaten Purwakarta :

- 1) Peraturan Bupati Purwakarta Nomor 148 Tahun 2016 tentang Kedudukan, Susunan Organisasi, Tugas dan Fungsi Perangkat.
- 2) Peraturan Bupati Purwakarta Nomor 160 Tahun 2016 tentang Perincian Tugas dan Fungsi Dinas Tata Ruang dan Pemukiman.

Tugas pokok Dinas Tata Ruang dan Pemukiman Kabupaten Purwakarta adalah, melaksanakan sebagian urusan pemerintahan daerah di bidang pekerjaan umum lingkup keciptakaryaan, penataan ruang, perumahan dan pemukiman, berdasarkan asas otonomi dan tugas pembantuan.

Fungsi Dinas Tata Ruang dan Pemukiman Kabupaten Purwakarta adalah :

- 1) Perumusan kebijakan teknis dan bidang keciptakaryaan, penataan ruang, dan perumahan.
- 2) Penyelenggaraan sebagian urusan pemerintahan dan pelayanan umum di bidang pekerjaan umum lingkup keciptakaryaan, penataan ruang dan perumahan.
- 3) Pembinaan dan pelaksanaan tugas meliputi bidang tata bangunan, pembinaan, pengawasan dan pengendalian, perumahan, pemukiman, dan tata ruang.
- 4) Pelaksanaan pelayanan teknis dan administratif dinas.
- 5) Pelaksanaan tugas lain yang diberikan oleh bupati sesuai dengan tugas dan fungsinya.

### **2.1.1. Visi dan Misi Dinas Tata Ruang dan Pemukiman Kabupaten Purwakarta**

#### **1) Visi Dinas Tata Ruang dan Pemukiman Kabupaten Purwakarta**

Terwujudnya tata ruang dan bangunan serta sarana dan prasarana perumahan dan pemukiman yang berkarakter dan berwawasan lingkungan. Visi Dinas Tata Ruang dan Pemukiman Kabupaten Purwakarta merupakan penjabaran dari visi pemerinatah Kabupaten Purwakarta dan misi ke – 2 (dua) Kabupaten Purwakarta sebagaimana telah tertuang dalam RPJMD Kabupaten Purwakarta tahun 2013 – 2018. Visi pembangunan jangka menengah Kabupaten Purwakarta tahun 2013 – 2018 adalah “Purwakarta Berkarakter”.

#### **2) Misi Dinas Tata Ruang dan Pemukiman Kabupaten Purwakarta**

Dinas Tata Ruang dan Pembangunan Kabupaten Purwakarta menetapkan misi sebagai berikut :

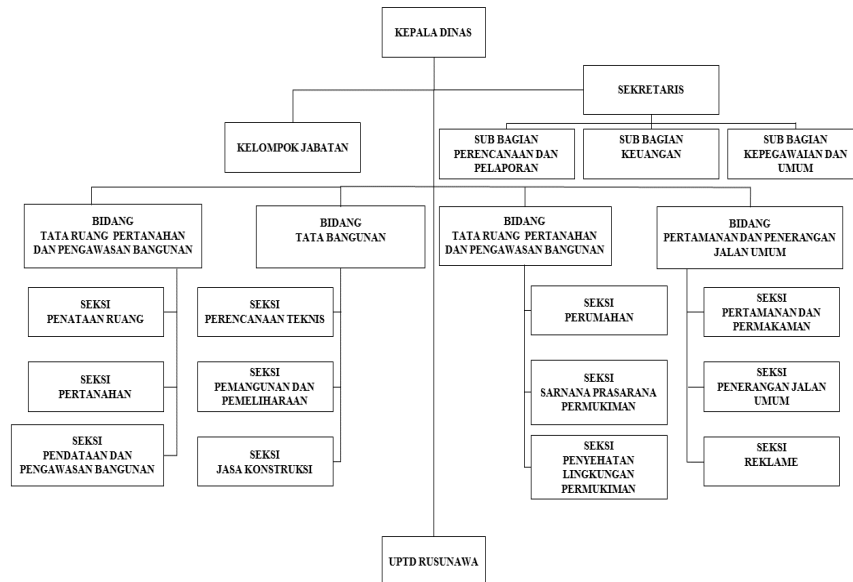
- 1) Mewujudkan sumber daya aparatur yang profesional dalam rangka meningkatkan pelayanan pada masyarakat.
- 2) Meningkatkan sarana dan prasarana perumahan dan permukiman dalam rangka terpenuhnya kebutuhan dasar masyarakat dan lingkungan yang bersih, dan nyaman.
- 3) Mewujudkan pelayanan infrastruktur wilayah di bidang penyediaan prasarana dan sarana air bersih.
- 4) Mewujudkan penyelenggaraan penataan ruang secara transparan, efektif dan partisipatif melalui kegiatan pengaturan, pembinaan, pelaksanaan dan pengawasan agar terwujud penataan ruang yang nyaman, nyaman, produktif dan berkelanjutan.
- 5) Mewujudkan tata kelola pertanahan untuk kepentingan pemerintah daerah dan pembangunan fasilitas publik agar terwujud tertib administrasi pertanahan.

### **2.1.2. Struktur Organisasi Dinas Tata Ruang dan Pemukiman Kabupaten Purwakarta**

Susunan Ststruktur Organisasi Dinas Tata Ruang dan Pemukiman Kabupaten Purwakarta Terdiri dari :

- 1) Kepala Dinas;
- 2) Sekretaris , membawahi :
  - a. Sub Bagian Perencanaan dan Pelaporan;
  - b. Sub Bagian Keuangan;
  - c. Sub Bagian Kepegawaian dan Umum.
- 3) Bidang Tata Ruang Pertanahan Dan Pengawasan Bangunan, membawahi :
  - a. Seksi Penataan Ruang;
  - b. Seksi Pertanahan;
  - c. Seksi Pendataan dan Pengawasan Bangunan.
- 4) Bidang Tata Bangunan, membawahi :
  - a. Seksi Perancangan Teknis;
  - b. Pembangunan dan Pemeliharaan;
  - c. Seksi Jasa Kontruksi.
- 5) Bidang Tata Ruang Pertanahan Dan Pengawasan Bangunan, membawahi :
  - a. Seksi Perumahan;
  - b. Seksi Sarana Prasarana Pemukiman;
  - c. Seksi Penyehatan Lingkungan Pemukiman.
- 6) Bidang Pertamanan Dan Penerangan Jalan Umum, membawahi :
  - a. Seksi Pertamanan dan Pemakaman;
  - b. Seksi Penerangan Jalan Umum (PJU);
  - c. Seksi Reklame.
- 7) Unit Pelaksana Teknis Daerah (UPTD);
- 8) Kelompok Jabatan.

Berikut adalah bagan struktur organisasi Dinas Tata Ruang dan Pemukiman Kabupaten Purwakarta dapat dilihat dalam Gambar 2.1 Struktur Organisasi Dinas Tata Ruang dan Pemukiman Kabupaten Purwakarta



**Gambar 2. 1 Struktur Organisasi Distarkim**

## 2.2. Landasan Teori

Landasan teori merupakan penjelasan berbagai konsep dasar teori – teori yang berkaitan dalam pengembangan sistem pemantauan dan kontrol lampu penerangan jalan umum berbasis internet of things ini. Beberapa teori terkait dengan pembangunan perangkat lunak ini adalah konsep yang digunakan, perangkat lunak perangkat keras dan bahasa pemrograman yang dibutuhkan dalam membangun Sistem Pemantauan Dan Kontrol Lampu Penerangan Jalan Umum Berbasis Internet Of Things.

### 2.2.1. Penerangan Jalan Umum

Penerangan jalan umum atau penerangan lampu jalan merupakan salah satu sistem penerangan yang berada di luar ruangan. Sistem lampu penerangan jalan umum yang berjalan dengan baik berguna menunjang keselamatan bagi pengguna trotoar jalan maupun pengemudi kendaraan

[10]. Lampu penerangan jalan umum (PJU) merupakan sarana penting bagi kehidupan masyarakat di malam hari, karena PJU selain berfungsi untuk menerangi jalan juga mempunyai fungsi lain seperti keamanan, keselamatan, dan fungsi keindahan [1]. Lampu PJU mempermudah pengguna jalan melihat dengan jelas jalan atau medan yang akan dilalui pada malam hari, sehingga dapat meningkatkan keselamatan lalu lintas dan keamanan dari para pengguna jalan. Clarke mengatakan bahwa “*Better lighting will deter offender who benefit from the cover of darkness*”, atau dalam bahasa artinya : Penerangan (jalan) yang baik akan menghalangi penyerang atau penjahat yang mengambil kesempatan dari kegelapan malam[11].

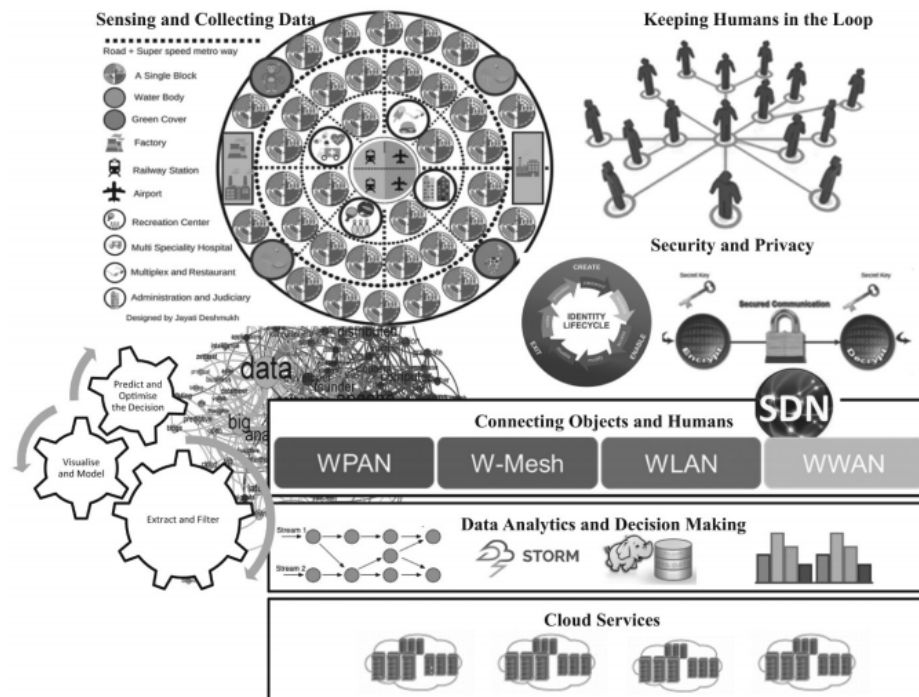
#### **2.2.1.1. Dasar Perencanaan Perancangan Penerangan Jalan Umum**

- 1) Perencanaan penerangan jalan umum terkait dengan hal – hal berikut :
  - a) Volume lalu – lintas, baik kendaraan maupun lingkungan yang bersinggungan seperti pejalan kaki, pengayuh sepeda, dan lain – lain.
  - b) Tipikal potongan melintang jalan, situasi (*lay-out*) jalan dan persimpangan jalan.
  - c) jalan, seperti alinyemen horisontal, alinyemen vertikal, dan lain – lain.
  - d) Jenis perkerasan yang memperngaruhi pantulan cahaya lampu penerangan.
  - e) Pemilihan jenis dan sumber cahaya, data fotorentik dan lokasi listrik,
  - f) Tingkat kebutuhan, biaya operasi, biaya pemeliharaan dan lain – lain.
  - g) Rencana jangka panjang pengembangan jalan dan pengermbangan daerah sekitar.
  - h) Data kecelakaan dan kerawanan lokasi.

- 2) Perhatian khusus dari beberapa tempat dalam perencanaan penerangan jalan antara lain sebagai berikut :
  - a) Lebar ruang milik jalan yang bervariasi.
  - b) Tempat – tempat dimana kondisi lengkung horisontal (tikungan) tajam.
  - c) Tempat luas, seperti persimpangan.
  - d) Jalan – jalan berpohon.
  - e) Jalan – jalan dengan median yang sempit, terutama untuk pemasangan lampu di bagian median.
  - f) Jembatan sempit/panjang, jalan layang dan jalan bawah tanah (terowongan).
  - g) Tempat – tempat lain dimana lingkungan jalan banyak berinterferensi dengan jalannya[10]. Internet Of Thing (IOT)

Setelah empat dekade sejak munculnya Internet oleh ARPANET, istilah internet mengacu pada kategori besar aplikasi dan protokol yang dibangun di atas jaringan komputer yang canggih dan saling berhubungan, melayani miliaran pengguna di seluruh dunia dalam 24/7. Fokusnya telah bergeser ke arah integrasi manusia dan perangkat tanpa adanya batasan untuk meliputi ranah fisik dengan lingkungan manusia, dan terciptalah *Internet Of Things* (IOT).

*Internet Of Things* (IOT) terdiri dari 2 pilar utama yaitu “*Internet*” dan “*Things*”, jadi setiap objek yang mampu terhubung ke internet akan masuk ke dalam kategori “*Things*” seperti mencakup seperangkat entitas yang lebih umum seperti *smartphone*, *sensors*, manusia dan objek lainnya. Konteksnya mampu berkomunikasi dengan entitas lain, membuatnya dapat diakses kapan saja, dimana saja. Secara garis besar dengan *Internet of Things* (IOT) objek harus dapat diakses tanpa batasan waktu atau tempat. Pada Gambar 2.2 menunjukkan *ecosystem Internet of Things* (IOT).



**Gambar 2. 2 Ecosystem Internet of Things [13].**

Internet of Things (IOT) juga diidentifikasi sebagai enabler untuk interaksi mesin-ke-mesin, manusia-ke-mesin, dan manusia dengan lingkungan. Dengan meningkatnya jumlah smart device dan adopsi protocol baru seperti IPv6, *Internet of Things* (IOT) diperkirakan akan bergeser kearah perpaduan *smart networking* dan *autonomous networks* dari objek-objek yang dilengkapi internet. *Internet of Things* (IOT) menawarkan banyak keuntungan untuk berbagai aplikasi, termasuk *emergency management*, *smart farming*, perawatan kesehatan dan lain-lain. Peran penting lain dari *Internet of Things* (IOT) adalah membangun sistem kolaboratif yang mampu merespons secara efektif terhadap peristiwa yang ditangkap melalui sensor dengan efektif [13]. Pada penelitian ini konsep *Internet of Things* (IOT) digunakan sebagai konsep utama lampu penerangan jalan umum dengan menggunakan Arduino uno, sensor yang digunakan adalah sensor arus untuk mendeteksi lampu mati dan sensor cahaya untuk menyalakan dan mematikan lampu secara otomatis.

### 2.2.2.1. Internet Of Things (IOT) Architectures

Block arsitektur *Internet Of Things* (IOT) adalah perangkat sensorik, *remote service invocation* dan komunikasi jaringan. Arsitektur sistem holistic untuk *Internet Of Things* (IOT) perlu menjamin secara sempurna pengoperasian komponennya dan menyatukan *hardware* dan *software* secara bersamaan, untuk mencapai hal ini, pertimbangan yang cermat diperlukan dalam merancang *failure recovery* dan *scalability* [13]. Model arsitektur *Internet Of Things* (IOT) sebagai berikut:

#### 1) *SOA-Based Architecture*

*Service-Oriented Architecture* (SOA) mungkin penting untuk penyedia layanan dan pengguna, memastikan *interoperability* di antara perangkat heterogen. Terdapat 4 lapisan fungsi yang berbeda sebagai berikut:

- a) *Sensing Layer*, integrasi dengan objek hardware yang tersedia untuk mendapatkan status dari lingkungan.
- b) *Network Layer*, infrastruktur untuk mendukung koneksi *wireless* atau koneksi kabel.
- c) *Service Layer*, untuk membuat dan mengelola layanan yang diperlukan oleh pengguna atau aplikasi.
- d) *Interfaces Layer*, terdiri dari metode interaksi dengan pengguna atau aplikasi.

Secara umum, dalam arsitektur SOA sistem yang kompleks dibagi menjadi beberapa subsistem yang modular sehingga memberikan cara mudah untuk mempertahankan keseluruhan sistem dengan merawat komponen individualnya, dengan ini dapat memastikan bahwa jika terjadi kegagalan komponen, sisa sistem masih dapat beroperasi secara normal, ini adalah nilai yang sangat besar untuk desain yang efektif dari arsitektur aplikasi *Internet Of Things* (IOT) dimana *reability* adalah parameter yang paling signifikan. SOA telah digunakan secara intensif di WSN, karena tingkat abstraksi yang sesuai dan keunggulan yang berkaitan dengan desain modularnya [13].

#### 2) *API-Oriented Architecture*

Pendekatan konvensional untuk mengembangkan solusi berorientasi layanan *Remote Method Invocation* (RMI) sebagai sarana untuk menggambarkan, menemukan dan memanggil layanan, karena *overhead* dan *complexity* yang dipaksakan oleh teknik ini, *API* web dan *Representational State Transfer* (REST) sebagai metode yang memiliki solusi alternatif yang menjanjikan. Sumber daya yang dibutuhkan berkisar dari bandwidth jaringan ke kapasitas komputasi dan penyimpanan data, dipicu oleh permintaan konversi data permintaan *respons* yang terjadi secara teratur selama *service calls*. Format pertukaran data ringan seperti *JavaScript Object Notation* (JSON) dapat mengurangi overhead, terutama untuk smart devices dan sensor dengan sumber daya terbatas, dengan mengganti file XML berukuran besar, ini membantu dalam menggunakan *communication channel* dan pemrosesan di perangkat lebih efisien. Membangun API untuk aplikasi Internet Of Things (IOT) membantu penyedia layanan menarik lebih banyak pelanggan sambil berfokus pada fungsionalitas produk dari pada presentasi. Selain itu, lebih mudah untuk mengaktifkan *multitenancy* dengan fitur keamanan API seperti *OAuth*, API yang memang mampu meningkatkan eksposisi dan komersialisasi layanan. Dengan API menyediakan pemantauan layanan dan alat bisa lebih efisien dari pada berorientasi dengan layanan sebelumnya. [13].

#### 2.2.2.2. Monitoring and Actuating

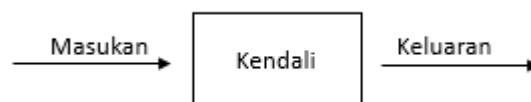
Pemantauan perangkat melalui API dapat membantu menjadi lebih mudah. API dapat melaporkan pengguna daya, kinerja peralatan dan status sensor sehingga alat dapat melakukan tindakan setelah mengirim perintah yang telah ditentukan. *Real-time application* memanfaatkan API untuk melaporkan sistem, sedangkan pengembang secara bebas memiliki pilihan pemanggilan API tanpa memerlukan akses perangkat secara fisik atau biasa disebut dengan *Remote Device*, dapat membantu identifikasi cara kinerja

melalu penerapan deteksi *anomaly* pada data yang dikumpulkan sehingga dapat meningkatkan produktivitas [13]

### 2.2.2.3. Sistem Kendali

Sitem kendali telah memegang peranan yang sangat penting dalam perkembangan ilmu teknologi. Peranan sistem kendali meliputi semua bidang kehidupan. Dalam peralatan, misslanya proses pada industri persawat terbang, peluru kendali, dan lain – lain. Sedangkan dalam bidang non teknis meliputi bidang biologi, ekonomi, social, kedokteran dan lain – lain. Sistem kendali yang semakin berkembang dapat meningkatkan kinerja sisitem, kualitas produksi, dan menekan biaya produksi. Sistem kendali dapat dikataka sebagai hubungan snatra komponen yang membentuk sebuah konfigurasi sistem, yang akan menghasilkan tanggapan sisitem yang diharapkan. Harus ada yang dikendalikan, yang merupakan sistem fisis, biasa disebut dengan kendalian [14].

Sistem kendali merupakan bagian yang terintegrasi dari sistem kehidupan dan lain sebagainya. Denagn sistem kendali memungkinkan variable yang ingin dikendalikan mencapai nilai yang diinginkan dengan mekanisme umpan balik den pengendalian. Dengan sistem kendali memungkinkan adanya sistem yang stabil, akurat, dan tepat waktu. Sistem kendali juga dapat dirancang untuk melakukan pengendalian secara otomaits. Di dalam industry banyak ditemukan sistem ini menggunakan “*Programmable Logic Controller*”. Gambar 2.3 menunjukan sistem kendali.



**Gambar 2. 3 Sistem Kendali [14]**

Masukan dan keluaran atau *input* dan *output* merupakan variable atau besaran fisis. Keluaran merupakan hal yang dihasilkan oleh kendalian yang

berarti yang dikendalikan, sedangkan masukan adalah yang mempengaruhi kendalian atau yang mengatur keluaran [15].

Dibawah ini merupakan definisi sistem kendali, diantaranya :

- 1) Sistem, adalah kombinasi dari banyak komponen yang bekerja secara bersamaan untuk mencapai tujuan tertentu.
- 2) Proses (alamiah), adalah urutan operasi yang kontinu atau perkembangan ditandai dengan urutan perubahan secara perlahan yang terjadi langkah demi langkah dengan cara yang relatif tetap dan memberikan suatu hasil atau akhiran.
- 3) Proses (artifisial), adalah operasi yang dilakukan secara bersamaan, terdiri dari beberapa aksi yang pergerakan secara sistematis diarahkan pada suatu atau dikendalikan.
- 4) *Plant*, merupakan bagian peralatan berfungsi secara bersamaan dan membentuk suatu operasi tertentu.
- 5) Gangguan adalah sinyal yang cenderung mempengaruhi nilai keluaran suatu sistem, terdapat 2 gangguan yaitu, gangguan internal dan eksternal.
- 6) *Feedback Control Sistem* adalah sistem kendali yang mempunyai elemen umpan balik, berfungsi untuk mengamati keluaran atau output untuk dibandingkan dengan masukannya
- 7) *Process Control System* merupakan sistem kendali umum dimana banyak digunakan oleh industri, contohnya seperti untuk mengendalikan tekanan, temperature, aliran, ketinggian air, dan lain – lain.
- 8) *Open Loop System* merupakan sistem kendali dimana tak terdapat elemen yang mengamati keluaran yang terjadi untuk dibandingkan dengan masukannya, meskipun menggunakan sebuah pengendali (*controller*) untuk memperoleh keluaran yang diinginkan.
- 9) *Closed Loop System*, merupakan sebutan lain dari sistem kendali umpan balik.

### 2.2.3. Internet

Internet adalah sekumpulan jaringan berbeda yang saling terhubung bersama sebagai suatu kesatuan dengan menggunakan berbagai macam protokol, salah satunya adalah protokol TCP/IP (*Transmission Control Protocol/Internet Protocol*), TCP/IP adalah protokol yang paling banyak digunakan di internet. Protokol TCP/IP merupakan cara standar untuk mengakses internet dengan memaketkan dan mengalamatkan komputer sehingga data dapat dikirim ke komputer yang dekat atau juga yang paling jauh dengan kurun waktu yang cepat tanpa kerusakan atas kehilangan data[16].

#### 2.2.3.1. Web Server

*Web Server* adalah suatu perangkat lunak yang dijalankan pada komputer server dan berfungsi agar dokumen internet server yang mampu untuk melayani koneksi perpindahan data dalam protokol *web server* disamping email. *Middleware* adalah perangkat lunak yang bekerja sama dengan *web server* dan berfungsi menterjemahkan kode – kode tertentu, menjalankan kode – kode tersebut dan memungkinkan berinteraksi dengan basis data. Dikarenakan web server dirancang untuk menampilkan data seperti teks, *hypertext*, gambar, *video*, musik, dan lainnya. Macam – macam web server antara lain, Apache (*Open Source*), Xitami, IIS, dan PWS. Sedangkan, web browser adalah salah satu perangkat lunak disisi *client* yang digunakan untuk mengakses informasi website[17]

### 2.2.4. Object Oriented (OO)

*Object Oriented* adalah sebuah objek memiliki keadaan sesaat (*state*) dan perilaku (*behavior*). *State* sebuah objek adalah kondisi objek tersebut yang dinyatakan dalam attribute/properties. Sedangkan perilaku sebuah objek dapat di definisikan bagaimana sebuah objek bertindak/beraksi dan memberikan reaksi. Perilaku objek dinyatakan dalam *operation*.

Menurut Schmuller, atribut dan operation bila disatukan akan memberikan fitur. Himpunan objek yang sejenis adalah contoh instance dari sebuah *class*. Ada dua macam aplikasi yang tidak cocok dikembangkan dengan metode OO. Pertama adalah aplikasi yang sangat berorientasi ke database. Aplikasi yang sangat berorientasi kepenyimpanan dan kepemanggilan data sangat tidak cocok dikembangkan dengan OO karena akan banyak kehilangan manfaat dari penggunaan RDBMS (*Relational Database Management System*) untuk penyimpanan data. Akan tetapi RDBMS juga mempunyai keterbatasan dalam penyimpanan dan pemanggilan struktur data yang kompleks seperti multimediam data spesial dan lain – lain. Bentuk aplikasi lain yang kurang cocok dengan pendekatan OO aplikasi yang membutuhkan banyak algoritma. Beberapa aplikasi yang melibatkan perhitungan yang kompleks [18].

Pada penelitian ini konsep OO digunakan untuk pengkodean pada sistem yang akan dibangun. Berikut adalah konsep – konsep dalam pemrograman berorientasi objek :

#### 1) *Class*

Kelas (*Class*), merupakan penggambaran satu objek yang memiliki atribut yang sama. Kelas mirip dengan tipe data pemrograman non objek, akan tetapi lebih komprehensif karena terdapat struktur sekaligus karakteristiknya. Kelas baru dapat dibentuk lebih spesifik dari kelas pada umumnya, kelas merupakan jantung pada pemrograman berorientasi objek.

#### 2) *Object*

Objek, merupakan teknik dalam menyelesaikan masalah yang kerap muncul dalam pengembangan perangkat lunak. Teknik ini merupakan teknik yang efektif dalam menemukan cara yang tepat dalam membangun sistem dan menjadi metode yang paling banyak dipakai oleh para pengembang perangkat lunak. Orientasi objek merupakan teknik pemodelan sistem ril yang berbasis objek.

#### 3) *Abstraction*

Kemampuan sebuah program untuk melewati aspek informasi yang diolah adalah kemampuan untuk fokus pada inti permasalahan, setiap objek dalam sistem melayani berbagai model dari pelaku abstrak yang dapat melakukan kerja, laporan dan perubahan serta berkomunikasi dengan objek lain dalam sistem, tanpa harus menampakkan kelebihan diterapkan.

#### 4) *Encapsulation*

Proses memastikan pengguna sebuah objek tidak dapat menggantikan keadaan dari sebuah objek dengan cara yang tidak sesuai prosedur. Artinya, hanya metode yang terdapat dalam objek tersebut yang diberi izin untuk mengakses keadaan yang diinginkan. Setiap objek mengakses interface yang menyebutkan bagaimana objek lainnya dapat terintegrasi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.

#### 5) *Polymorfism*

Suatu fungsionalitas yang diimplikasikan dengan berbagai cara yang berbeda. Pada program berorientasi objek, pembuat program dapat memiliki berbagai implementasi untuk sebagian fungsi tertentu.

#### 6) *Inheritance*

Seperti yang sudah disebutkan sebelumnya, objek adalah contoh/instance dari sebuah *class*. Hal ini mempunyai konsekuensi yang penting yaitu sebagai instance sebuah *class*, sebuah objek mempunyai semua karakteristik dari *class*nya inilah yang disebut dengan Inheritance (pewarisan sifat). Dengan demikian apapun attribute dan operation dari *class* akan dimiliki pula oleh semua objek yang diinherit/diwariskan dari *class* tersebut. Sifat ini tidak hanya berlaku untuk objek terhadap *class*, akan tetapi juga berlaku untuk *class* terhadap *class* yang lainnya.

### 2.2.5. *Unified Modelling Language (UML)*

*Unified Modelling Language* adalah suatu alat bantu yang sangat handal dalam dunia pengembangan sistem yang berorientasi objek. Hal ini disebabkan karena UML menyediakan sistem yang berorientasi pemodelan visual yang memungkinkan bagi pengembang sistem untuk cetak biru atas visi mekanisme yang efektif untuk berbagai dan mengkomunikasikan rancangan mereka dengan yang lain. *Unified Modelling Language* merupakan kesatuan dari bahasa pemodelan yang dikembangkan Booch, *Object Modelling Technique (OMT)* dan *Oriented Software Engineering (OOSE)*.

Metode ini menjadikan proses analisis dan desain ke dalam empat tahapan iteratif, yaitu identifikasi kelas – kelas, dan objek – objek, identifikasi semantik dari hubungan objek dan kelas tersebut, perincian interface dan implementasi. Keunggulan metode Booch adalah pada detail dan kayanya dengan notasi dan elemen, pemodelan OMT yang dikembangkan oleh Rumbaugh didasarkan pada analisis terstruktur dan pemodelan *entry-relationship*. Tahapan utama dalam metodologi ini adalah analisis, desain sistem, desain objek dan implementasi. Keunggulan metode ini adalah adanya notasi yang mendukung konsep OO.

Metode OOSE dari Jacobson lebih memberi penekanan pada *use case*, OOSE memiliki tiga tahapan yaitu membuat *model requirement* dan analisis, desain dan implementasi, dan model pengujian. Keunggulan metode ini adalah mudah dipelajari karena memiliki notasi sederhana namun mencakup seluruh tahapan dalam rekayasa perangkat lunak. Dengan UML, metode Booch, OMT dan OSE digabungkan dengan membuang elemen – elemen yang tidak praktis ditambah dengan elemen – elemen dari metode lain yang lebih efektif dan elemen – elemen baru yang belum ada pada metode terdahulu sehingga UML lebih ekspresif dan seragam daripada metode lainnya.

Pada penelitian ini konsep UML digunakan untuk menggambarkan bagaimana sistem bekerja hubungan antar *class* memberi gambaran kepada

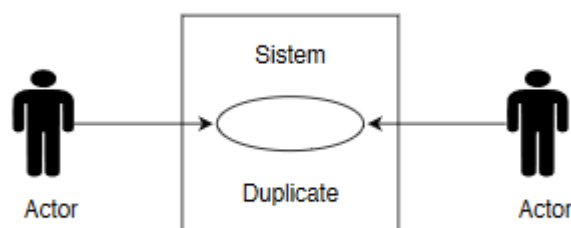
pengguna sistem dan memberikan detail pemanggilan fungsi – fungsi dan method dalam *class* [18].

#### 2.2.5.1. Diagram UML

UML menyediakan 4 macam diagram untuk memodelkan aplikasi perangkat lunak berorientasi objek, yaitu :

##### 1) *Use case Diagram*

*Use case diagram* adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara pengguna sebuah sistem dengan sistemnya sendiri melalui sebuah alur bagaimana sistem digunakan. Urutan langkah – langkah yang menerangkan antara pengguna dan sistem disebut *scenario*. Setiap *scenario* mendeskripsikan urutan kejadian. Setiap urutan diinisialisasi oleh orang, sistem yang lain, oerangkat keras atau urutan waktu. Dengan demikian *use case* dapat dikatakan serangkaian *scenario* yang digabungkan bersama – sama dengan tujuan umum pengguna. Diagram *use case* menunjukkan 3 aspek dari sistem yaitu : *actor*, *use case*, dan sistem atau sub sistem *boundary*. *Actor* mewakili peran orang, sistem yang lain atau alat ketika berkomunikasi dengan *use case*. Gambar 2.4 menunjukkan *Use case diagram* dalam UML[18].

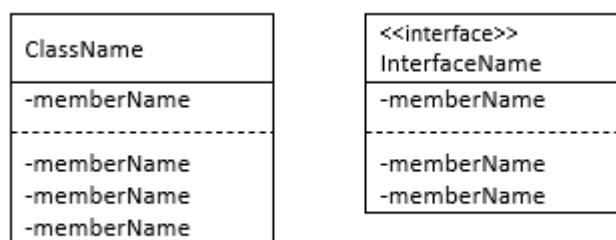


**Gambar 2. 4 Use Case Diagram [18]**

##### 2) *Class Diagram*

*Class diagram* adalah diagram statis, dapat digunakan untuk memvisualisasikan, menggambarkan, dan mendokumentasikan berbagai aspek dan juga membangun kode eksekusi dari aplikasi

perangkat lunak. *Class* diagram menggambarkan attribute operation dan juga *constraint* yang terjadi pada sistem OO karena itu adalah satu – satunya diagram UML yang dapat dipetakan langsung dengan bahasa berorientasi objek. *Class* diagram menunjukkan koleksi *class* antar muka, asosiasi, kolaborasi, dan *constraint*. Dikenal juga sebagai diagram struktural[18]. Gambar 2.5 menunjukkan *Class Diagram* dalam UML.



**Gambar 2. 5 Class Diagram [18]**

### 3) Activity Diagram

Activity Diagram adalah bagian dari UML yang menggambarkan aspek dinamis dari sistem. Logika procedural, proses bisnis dan aliran kerja suatu bisnis bisa dengan mudah dideskripsikan dalam *activity diagram*. *Activity diagram* memiliki peran seperti *flowchart*, perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa. Tujuan dari *activity diagram* untuk menangkap tingkah laku dinamis dari sistem dengan cara menunjukkan aliran pesan dari satu aktifitas ke aktifitas lainnya. Secara umum digunakan untuk menggambarkan diagram alur yang terdiri dari banyak aktifitas dalam sistem dengan beberapa fungsi tambahan seperti percabangan, aliran paralel, *swim lane*, dan sebagainya. Sebelum menggambarkan sebuah *activity diagram* perlu mengetahui elemen yang akan digunakan dalam *activity diagram*. Elemen utama dalam *activity diagram* adalah aktifitas itu sendiri, aktifitas adalah fungsi yang dilakukan oleh sistem.

Setelah aktifitas teridentifikasi, selanjutnya bagaimana semua elemen tersebut berasosiasi dengan *constraint* dan kondisi. Langkah selanjutnya perlu penjabaran teta letak dari keseluruhan aliran, agar bisa ditransformasikan ke activity diagram[18].

Di bawah ini merupakan komponen – komponen dalam *activity diagram* :

a) *Activity node*

Menggambarkan bentuk notasi beberapa proses yang beroperasi dalam kontrol nilai data.

b) *Activity edge*

Menggambarkan bentuk *edge* yang menghubungkan aliran aksi secara langsung, dimana menghubungkan *input* dan *output* dari aksi tersebut.

c) *Inisial State*

Bentuk lingkaran berisi penuh menunjukan awal dari suatu proses.

d) *Decision*

Bentuk dengan suatu flow yang masuk beserta dua atau lebih *activity node* yang keluar. *Activity node* yang keluar ditandai untuk mengindikasikan beberapa kondisi.

e) *Fork*

Satu bar hitam dengan satu *activity node* yang masuk beserta dua atau lebih *activity node* yang keluar

f) *Join*

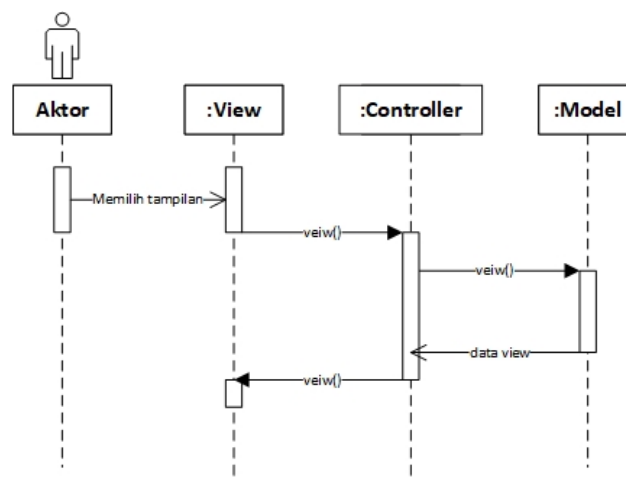
Satu bar hitam dengan dua atau lebih *activity node* yang masuk beserta satu *activity node* yang keluar, tercatat pada akhir proses secara bersamaan. Semua *actions* yang menuju *join* harus lengkap sebelum proses dapat berlanjut.

g) *Final State*

Bentuk lingkaran berisi penuh yang berbeda di dalam lingkaran kosong, menunjukan akhir dari suatu proses

#### 4) *Sequence Diagram*

Sequence Diagram, digunakan untuk menggambarkan perilaku pada sebuah *scenario*. Diagram ini menunjukkan sejumlah contoh objek dan *message* (pesan) yang diletakan diantara objek – objek ini di dalam *use case*. Komponen utama *sequence diagram* terdiri atas objek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progress vertical*. Manfaat *sequence diagram* adalah memberikan gambaran detail dari setiap *use case diagram* yang dibuat sebelumnya [18]. Gambar 2.6 menunjukkan *Sequence Diagram*.



**Gambar 2. 6 *Sequence Diagram* [18]**

#### 2.2.6. **Ftizing**

*Frizting* merupakan salah satu aplikasi untuk mendesain rangkaian elektronika. Sama halnya seperti aplikasi (*Eagle*, *Protel*, *Pad2pad*, dan lain – lain). *Frizting* juga memiliki *Board Designer*. Pada penelitian ini *frizting* digunakan untuk mendesain komponen arduino, sensor – sensor dan modul yang digunakan [21] *Frizting* dapat dilihat pada Gambar 2.7 *Frizting*



**Gambar 2. 7 Frizting[21]**

Frizting berbeda dengan aplikasi – aplikasi lainnya, berikut adalah kelebihan yang ada pada aplikasi frizting :

- a) Open Source, menyediakan banyak library untuk mikrokontroller buatan seperti *Arduino*, *Raspberry Pi*, *Basic Stamp*, *Pic.exe*, dan lain – lain. Librarynya tersedia banyak diinternet dan bisa didownload secara gratis.
- b) Gambar rangkaian bisa di *share* di internet, gambar rangkaiannya bersifat *Open Hardware*.
- c) Dapat menggambar rangkaian di *BreadBoard*, gambar rangkaiannya mudah dimengerti, bisa langsung ditiru ke *BreadBoard*.

#### **2.2.7. Visual Studio Code**

Visual Studio Code adalah sebuah aplikasi editor kode yang tidak hanya tersedia untuk Windows, tapi juga tersedia untuk sistem operasi Linux dan Mac OS.

#### **2.2.8. Hypertext PreProcessor (Php)**

PHP Hypertext Preprocessor (Php) adalah bahasa serverside scripting yang menyatu dengan HTML untuk membuat halaman web yang dinamis menurut (Nugroho, 2004) PHP banyak dipakai untuk pemrograman situs Web dinasi. Karena PHP merupakan server-side scripting maka sintaks dan perintah – perintah PHP akan dieksekusi di server kemudian hasilnya dikirim ke browser dalam format HTML. Dengan demikian kode program yang ditulis dalam PHP tidak akan terlihat oleh user sehingga keamanan halaman web lebih terjamin. PHP dirancang untuk membentuk suatu

tampilan berdasarkan permintaan terkini, seperti menampilkan isi baris ke halaman web.

Beberapa kelebihan PHP dari bahasa pemrograman web, antara lain :

- 1) Bahasa pemrograman php adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya.
- 2) PHP memiliki tingkat akses yang lebih cepat.
- 3) PHP memiliki tingkat lifecycle yang cepat sehingga selalu mengikuti perkembangan teknologi internet.
- 4) PHP juga mendukung akses ke beberapa database yang sudah baik dan bersifat gratis ataupun komersial. Database itu antarlain: MySQL, PostgreSQL, infomix, dan MicrosoftSQL Server. Web yang mendukung PHP dapat ditemukan dimana – mana seperti dari Apache, IIS, AOServer, phttp, Fhttp. PWS, Lightpd hingga Xitami dengan konfigurasi yang relative mudah[23].

#### **2.2.9. Cascading Style Sheet (Css)**

CSS (Cascading Style Sheet) adalah salah satu bahasa desain web (Style Sheet Language) yang mengontrol format tampilan sebuah halaman web yang ditulis dengan menggunakan penanda (markup kanguange). Biasanya CSS digunakan untuk medesain sebuah halaman HTML san XHTML. Tetapi sekarang CSS bisa diaplikasikan untuk segala sokumen XML, termasuk SVG dan XUL bahkan Android.

CSS dibuat untuk memisahkan konten utama dengan tampilan dokumen yang meliputi layout warna, dan font. Pemisah ini dapat meningkatkan daya akses konten pada web, menyediakan lebih banyak fleksibilitas dan kontrol dalam sfesifikasi dari sebuah formatting dan mengurangi kerumitan dalam penulisan kode dan struktur dari konten, contohnya teknik tableless pada desain web.

CSS juga memungkinkan sebuha halaman untuk ditampilkan dalam berbagai style dengan menggunakan metode pembawaan yang berbeda,

seperti on-screen, by voice, dan lain – lain. Semenata itu, pemilik konten web bisa menentukan link yang menghubungkan konten dengan file CSS.

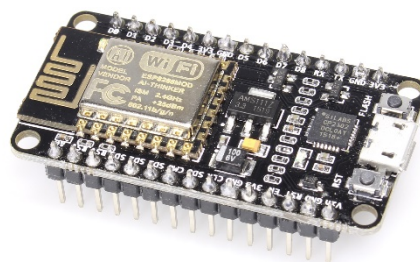
#### 2.2.10 MySQL

*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 di bawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersia[31].

Pada penelitian ini MySQL digunakan sebagai platform penyimpanan data dan implementasi model rancangan database yang sudah di buat.

#### 2.2.11 NodeMCU

NodeMCU adalah sebuah platform IoT yang bersifat *opensource*. Terdiri dari perangkat keras berupa *System On Chip* ESP8266 dari ESP8266 buatan *Espressif System* juga firmware yang digunakan, yang menggunakan bahasa pemrograman Lua. [Sumardi, 2016]. NodeMCU dapat dilihat pada Gambar 2.8 NodeMCU



**Gambar 2. 8 NodeMCU [12]**

Istilah NodeMCU secara *default* sebenarnya mengacu pada firmware yang digunakan dari perangkat keras *development kit* NodeMCU bisa

dianalogikan sebagai board arduino-nya ESP8266. Pada penelitian ini akan memanfaatkan NodeMCU sebagai tempat pertukaran data.

### 2.2.12 Sensor

Sensor adalah suatu perangkat yang terdeteksi perubahan energi yang berada di alam seperti energi listrik energi fisika, energi kimia, energi biologi, energi mekanik dan sebagainya. Di dalam sebuah sensor terdapat transducer yang berfungsi untuk mengubah besaran mekanis, magnetis, panas, sinar, dan kimia menjadi besaran listrik berupa tegangan, resistansi dan arus listrik[25].

#### 2.2.12.1 Relay

*Relay* merupakan komponen elektronika yang memiliki fungsi bekerja sebagai saklar mekanik yang digerakan oleh energi listrik. *Relay* menggunakan gaya elektromagnetik untuk memutuskan atau menghubungkan suatu rangkaian elektronika yang satu dengan yang lainnya. Relay dapat dilihat pada Gambar 2.9 Relay



Gambar 2. 9 Relay [26]

*Relay* terdiri dari coil dan contact. *Coil* adalah gulungan kawat yang mendapat arus listrik, sedangkan *contact* adalah sejenis saklar yang pergerakannya tergantung dari ada tidaknya arus listrik di *coil*. *Contact* ada dua jenis yaitu *normally open* (kondisi awal sebelum diaktifkan *open*), dan *normally closed* (kondisi awal sebelum diaktifkan *closed*)[26].

Pada penelitian ini menggunakan *relay* 2 CH sebagai penghubung dan pemutus arus listrik pada penerangan jalan umum.

#### 2.2.12.2 PZEM-004T

PZEM-004T adalah sensor yang dapat digunakan untuk mengukur tegangan rms, arus rms dan daya aktif yang dapat dihubungkan melalui arduino ataupun platform opensource lainnya. Sensor PZEM004T dapat dilihat pada Gambar 2.10 Sensor PZEM004T



**Gambar 2. 10 Sensor PZEM004T [12]**

Modul ini terutama digunakan untuk mengukur tegangan AC, arus, daya aktif, frekuensi, faktor daya dan energi aktif.

#### 2.2.12.3 Photocell

Photo Cell merupakan resistor rangkaian elektronik yang berisi komponen LDR(Light Dependent Resistor) di dalamnya, berfungsi sebagai saklar otomatis, menyalakan dan mematikannya bisa di atur berdasarkan sensor cahaya. Photocell dapat dilihat pada Gambar 2.11 Photocell



**Gambar 2. 11 Photocell [30]**

Prinsip kerja resistor dengan sensitivitas cahaya. Apabila kondisi gelap maka nilai resistansi akan rendah sehingga arus mengalir dan lampu akan menyala. Sebaliknya pada kondisi terang, nilai resistansi menjadi tinggi arus tidak dapat mengalir dan lampu akan mati.

#### **2.2.13 *Smart Energi***

Teknologi smart energi diperkenalkan bertujuan untuk mengidentifikasi potensi antar sub-sektor, sistem smart energi memasukkan seluruh sistem energi dalam pendekatannya untuk mengidentifikasi desain infrastruktur energi yang sesuai. Salah satu poin utamanya bahwa analisis teknologi dan sektor bersifat kontekstual, dan konsep utama lainnya adalah bahwa berbagai sub-sektor mempengaruhi yang lain dan harus dipertimbangkan solusi terbaik harus diidentifikasi[30].

Pada penelitian ini smart energy (solar energy) digunakan untuk memanfaatkan penerangan dari matahari di konversi menjadi tenaga arus listrik dan disimpan didalam baterai sehingga menjadi sumber daya listrik utama ke microcontroller dan alat lainnya.

#### **2.2.14 *Pembangkit Listrik Tenaga Surya***

Pembangkit listrik tenaga surya adalah sebuah daya alternatif dari daya konvensional (PLN), PLTS yang bersumber dari sinar matahari ini tidak memiliki keterbatasan di Benua Asia Tenggara terutama di Indonesia,

karena termasuk Negara Tropis hampir satu tahun penuh matahari akan menyinari dataran ini[30].

#### **2.2.14.1 Solar Panel**

Solar Panel adalah alat yang terdiri dari sel surya yang mengubah energi cahaya menjadi listrik arus DC dengan sel – sel fotovoltaik. Solar panel termasuk ramah lingkungan karena bisa memiliki umur lebih dari 20 tahun, dan dalam kurun waktu tersebut solar panel akan mengalami penurunan efisiensi yang signifikan. Bisa dilihat pada Gambar 2.12 Solar Panel

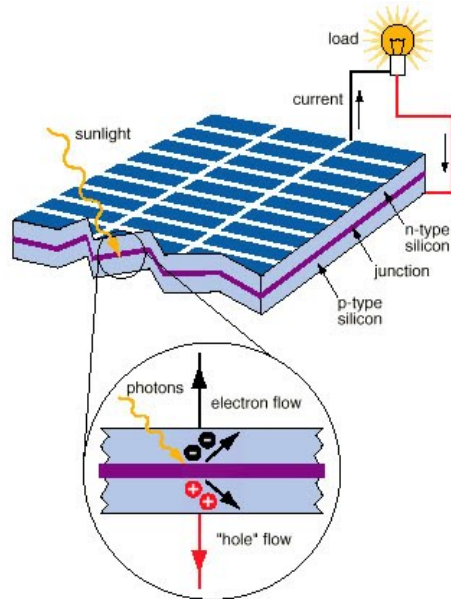


**Gambar 2. 12 Solar Panel [30]**

Posisi solar panel menghadap langsung ke sinar matahari agar pencahayaan bisa maksimum. Solar panel memiliki perlindungan overheating yang baik, perlindungan overheating sangat penting dikarenakan panel surya mengubah dari 20% energi surya yang ada menjadi listrik, sementara sisanya akan terbuang sebagai panas, dan tanpa perlindungan yang memadai kejadian overheating dapat menurunkan efisiensi panel surya [30].

### 2.2.14.2 Cara Kerja Panel Surya

Panel surya bekerja dengan menggunakan prinsip p-n junction, semikonduktor yang biasanya digunakan adalah silikon. Untuk mengubah sifat kelistrikan silikon diperlukan doping dengan atom yang memiliki elektrovalensi 3 seperti Boron, bisa dilihat pada Gambar



**Gambar 2. 13 Cara Kerja Panel Surya [30]**

Silikon bertipe ini, mempunyai kekurangan elektron karena boron merupakan unsur yang memiliki valensi 3 sehingga muncul hole yang bersifat sebagai akseptor. Sedangkan untuk memperoleh silikon tipe-n diperlukan doping dengan atom. Atom fosfor merupakan unsur yang memiliki valensi 5, ketika mendoping maka akan muncul elektron berlebih yang bersifat donor.

Pada saat kedua jenis semikonduktor melakukan kontak, maka elektron di semikonduktor tipe-n akan bergerak menuju hole pada semikonduktor tipe-p. Terbentuklah kutub positif pada tipe-p dan negatif pada tipe-n. Karena elektron dan hole berpasangan maka terbentuk medan listrik. Ketika cahaya matahari menyinari solar panel maka tiap satu foton akan menghasilkan 1 elektron dan 1 hole yang juga masuk jangkauan medan listrik, elektron akan bergerak ke semikonduktor tipe-n dan hole akan

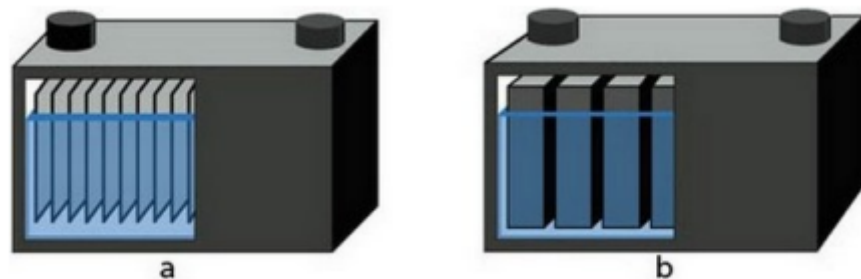
bergerak ke semikonduktor tipe-p Apabila diberi jalur arus eksternal, maka elektron akan bergerak ke tipe-p dan bersatu dengan hole, aliran elektron berherak menghasilkan arus, dan medan listrik akan menghasilkan tegangan. Diperolehlah daya listrik yang diperlukan[30]

#### 2.2.14.3 Daya Lsitrk Pada Panel Surya

Panel surya memiliki nila WP (Watt Peak) menggambarkan besaran daya listrik tertinggi yang dapat dihasilkan oleh suatu soler panel. Solar panel 50 WP menandakan bahwa daya listrik maksimal yang dapat dihasilkan oleh panel surya tersebut adalah 50 watt setiap jam pada kondisi saaat matahari terik, beberapa faktor yang dapat mempengaruhi kinerja panel surya seperti cuaca mendung yang mengakibatkan cahaya matahari tidak maksimal[30]

#### 2.2.14.4 Baterai / Aki

Aki adalah media penyimpanan muatan listrik, aki dibedakan berdasarkan aplikasi dan konstruksi. Berdasarkan aplikasi aki dibedakan untuk engine starter dan deep cyclr. Aki otomotif umumnya dibuat utuk bisa menyuplai arus listrik yang besar pada saat awal menghidupkan mesin, sedangkan aki deep cycle biasanya digunakan untuk sistem solar sell, dan back up powe dimana aki mampu mengalami discharge dengan muatan llistriknya tinggal sedikit[30] Gambar 2.14 Jenis Aki



**Gambar 2. 14 Jenis Aki [30]**

Jenis aki starter sebaiknya tidak mengalami discharge hingga melampaui 50% kapasitas muatan listrik untuk menjaga keawetan aki. Berkurangnya

kapasitas aki karena proses pembentukan kristal sulfat yang menempel pada pelat ketika muatan aki tidak penuh atau di bawah 50%.

#### 2.2.14.5 Solar Charge Controller

Solar Charge Controller adalah alat pengisian solar cell ke baterai, digunakan untuk mengatur arus searah dari panel surya ke baterai dan arus yang diambil dari baterai ke beban. Controller dapat juga mengatasi pengisian baterai yang sudah terisi penuh dan kelebihan regangan dari panel surya, Gambar adalah solar charger controller. Solar Charge Controller dapat dilihat pada Gambar 2.15 Solar Charge Controller



**Gambar 2. 15 Solar Charge Controller [29]**

Selain mengatur arus, controller juga memiliki kemampuan mendeteksi kapasitas baterai pada saat pengisian arus. Jika baterai terisi penuh maka pengisian arus secara otomatis dari solar panel akan berhenti. Arus listrik yang berasal dari baterai tidak mungkin bisa berbalik arah ke solar panel karena controller terdapat diode protection yang hanya melewatkan arus listrik dari solar panel ke baterai [29].

#### 2.2.14.6 Power Inverter DC to AC

Power Inverter adalah suatu rangkaian atau perangkat elektronika yang dapat mengubah arus listrik searah (DC) ke arus listrik (AC) pada tegangan dan frekuensi yang dibutuhkan sesuai dengan perancangan rangkaianannya. Sumber – sumber arus listrik searah atau arus DC yang merupakan Input dari Power Inverter tersebut dapat berupa Baterai, Aki maupun Sel Surya. Power inverter dapat dilihat pada Gambar 2.16 Power Inverter DC to AC



**Gambar 2. 16 Power Inverter DC to AC[29]**

Bentuk – bentuk gelombang yang dapat dihasilkan oleh power inverter adalah gelombang persegi (Square Wave) dan gelombang sinus (Sinus Wave), gelombang sinus yang dimodifikasi dan gelombang pulsa lebar tergantung pada desain rangkaian inverter.

#### **2.2.15 Metode Pengujian**

Pengujian perangkat lunak merupakan proses eksekusi program atau perangkat lunak dengan tujuan mencari kesalahan atau kelemahan dari program tersebut. Proses dilakukan dengan mengevaluasi atribut dan kemampuan program, suatu program yang diuji akan dievaluasi apakah hasil yang dikeluarkan berupa *output* telah sesuai dengan yang diinginkan atau tidak. Beberapa metode pengujian seperti *white box* dan *black box* yang telah dikenal dan banyak digunakan oleh pengembang perangkat lunak.

##### **2.2.15.1 Pengujian Black Box**

*Black box testing* merupakan metode pengujian dengan pendekatan yang mengasumsikan sebuah sistem perangkat lunak atau program sebagai sebuah kotak hitam (*black box*). Pendekatan ini hanya mengevaluasi program dari output atau hasil akhir yang dikeluarkan oleh program tersebut. Struktur program dan kode-kode yang ada di dalamnya tidak

termasuk dalam pengujian ini. Keuntungan dari metode pengujian ini adalah mudah dan sederhana. Namun, pengujian dengan metode ini tidak dapat mendeteksi kelemahan pada pengkodean dalam suatu program. Metode pengujian *black box* merupakan metode pengujian dengan pendekatan yang mengasumsikan sebuah sistem perangkat lunak atau program sebagai sebuah kotak hitam (*black box*).

