

BAB II

TEORI DASAR

Pada bab ini dibahas tentang perancangan, konstruksi atap, motor DC, *pulley*, sensor hujan, sensor cahaya (*LDR*), Arduino, *software* pendukung berupa *Arduino Software (IDE)*, *Solidwoks* dan *SimWise 4D*.

2.1 Konsep Dasar Perancangan

Perancangan adalah suatu proses untuk menterjemahkan kebutuhan pemakai informasi kedalam suatu alternatif rancangan yang diinginkan kepada pemakai informasi untuk dapat dipertimbangkan. Perancangan merupakan langkah awal dalam membuat sebuah produk. Hasil perancangan adalah gambar, dimensi dan spesifikasi produk. Produk tidak dapat dibuat apabila gambar dan spesifikasi produk tersebut belum ada.

Hasil perancangan harus mudah dibaca oleh orang yang akan membuat produk. Jika hasil perancangan berupa gambar dan spesifikasi tidak jelas maka produk akan sulit untuk dibuat karena kurangnya data yang dibutuhkan untuk membuat produk tersebut. Jika hasil perancangan berupa gambar dan spesifikasi jelas maka produk akan mudah dibuat karena data yang dibutuhkan mudah dibaca.

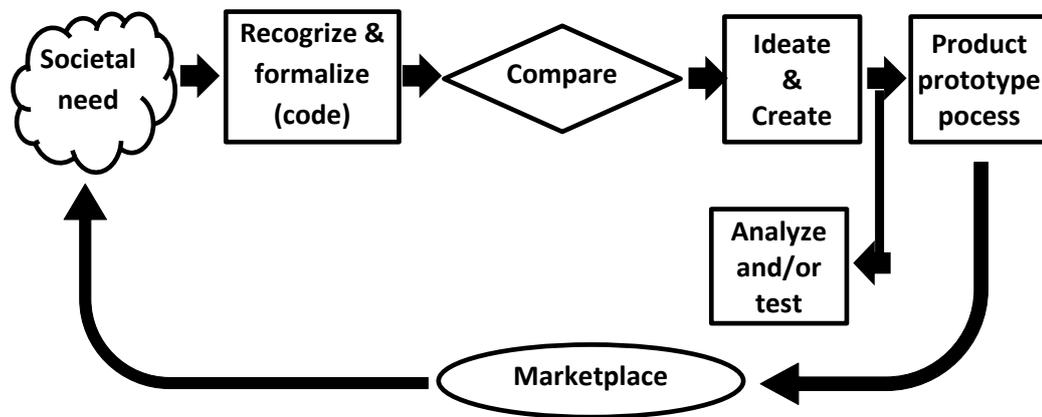
Dalam merancang sebuah produk ada beberapa hal yang harus diperhatikan. Beberapa hal yang harus diperhatikan dalam merancang sebuah produk diantaranya konsep rancangan produk yang dibuat harus jelas, data produk yang dirancang harus lengkap dan jelas serta hasil rancangan harus mudah dibaca.

2.1.1 Syarat dan Langkah-langkah Perancangan

Perancangan (*design*) secara umum dapat didefinisikan sebagai formulasi suatu rencana untuk memenuhi kebutuhan manusia. Secara sederhana perancangan dapat diartikan sebagai kegiatan pemetaan dari ruang fungsional (tidak terlihat/imajiner) ke ruang fisik (terlihat dan dapat diraba/dirasa) untuk memenuhi tujuan-tujuan akhir perancang secara spesifik atau obyektif.

Dalam prosesnya, perancangan adalah kegiatan yang biasanya dilakukan secara berulang-ulang (*iterative*). Kegiatan perancangan umumnya dimulai dengan didapatkannya persepsi tentang kebutuhan masyarakat kemudian dijabarkan dan disusun dengan spesifik, ide dan penguasaan kreasi kemudian dianalisis dan diuji. Jika hasilnya sudah memenuhi syarat perancangan maka akan dibuat prototipe. Jika prototipe terbaik sudah dipilih maka

selanjutnya produk dilempar ke pasaran. Pasar akan memberikan tanggapan apakah kebutuhan telah terpenuhi atau belum. Secara skematis kegiatan iteratif ini ditunjukkan pada gambar 2.1.



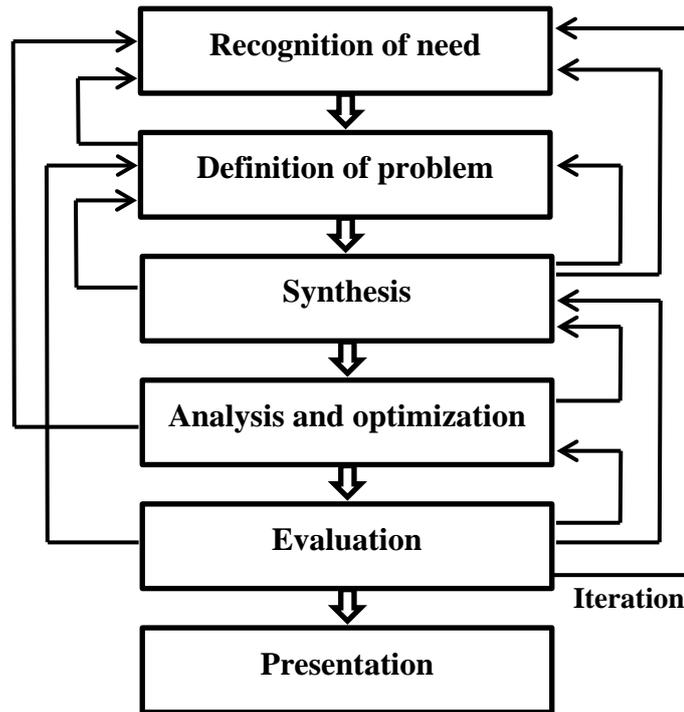
Gambar 2.1 Proses Iteratif dalam Perancangan

Kegiatan perancangan dalam bidang teknik yang dilakukan oleh para ahli teknik (insinyur) selama ini telah mampu meningkatkan kesejahteraan dan kualitas hidup manusia baik dalam meningkatkan kesehatan fisik masyarakat, memakmurkan masyarakat dalam hal materi dan memudahkan manusia untuk melakukan aktivitasnya. Hasil perancangan insinyur ini terdapat dalam berbagai bentuk produk dan jasa. Dengan demikian perancangan dalam bidang teknik atau *engineering design* selanjutnya dapat didefinisikan sebagai rangkaian kegiatan iteratif yang mengaplikasikan berbagai teknik dan prinsip-prinsip *scientific* yang bertujuan untuk mendefinisikan peralatan, proses, atau sistem secara *detail* sehingga dapat direalisasikan.

Dari pengertian umum di atas, *mechanical design* dapat diartikan sebagai perancangan sesuatu atau sistem dari “*mechanical nature*” seperti mesin, komponen, struktur, peralatan, instrumentasi, dan lain-lain. Dalam cakupan yang lebih spesifik *mechanical design* adalah kegiatan yang berhubungan dengan penciptaan (*creation machinery*) berdasarkan standar pada industri agar hasil penciptaannya berfungsi dengan baik, aman dan andal.

2.1.2 Proses Perancangan Teknik

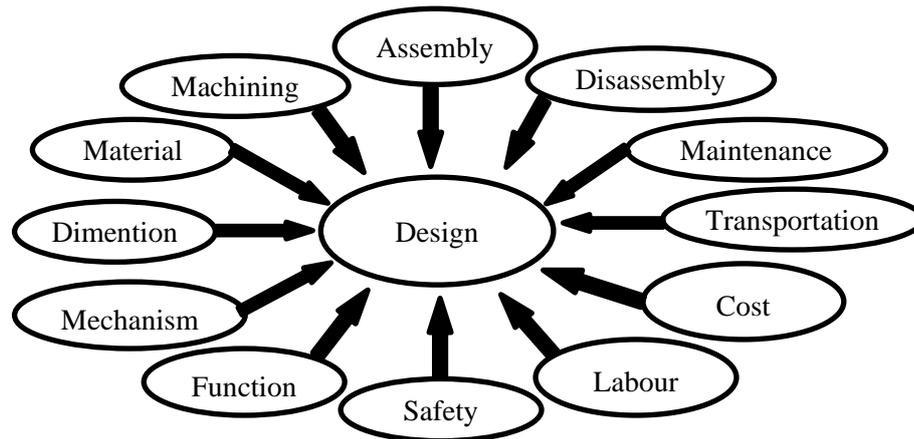
Skema proses *engineering design* yang lengkap ditunjukkan pada gambar 2.2. Proses *engineering design* dimulai dengan identifikasi kebutuhan dan keputusan untuk melakukan sesuatu tentang kebutuhan itu. Setelah dilakukan iterasi berkali-kali, proses *design* akan berhenti pada *detail design* yang siap dipresentasikan untuk selanjutnya dibuat prototipe, *testing*, dan akhirnya masuk proses produksi.



Gambar 2.2 Tahapan Proses *Design*

Identifikasi dan formulasi kebutuhan (*recognition of need*) adalah kegiatan yang membutuhkan tingkat kreativitas yang tinggi. Tahap *recognition of need* sering rancu dengan berbagai kondisi emosional manusia seperti *uneasiness* atau perasaan bahwa ada sesuatu yang salah. *Background Research* sangat diperlukan untuk memberikan informasi dalam memahami dan mendefinisikan masalah secara lengkap dan *detail*. Jika tahap *recognition of need* dilakukan dengan baik maka dapat ditetapkan tujuan (*goal*) dari *design*.

Tahap *Problem Definition* harus melibatkan semua spesifikasi yang berhubungan dengan sistem yang akan di-*design*. Spesifikasi tersebut adalah kuantitas *input* dan *output*, karakteristik dan dimensi serta ruangan yang diperlukan, dan semua kendala atau batasan *design*. Spesifikasi inilah yang akan menentukan biaya, jumlah yang akan dibuat, umur teknis yang diinginkan, kondisi operasi, dan keandalan *machinary*. Contoh spesifikasi yang berhubungan dengan sistem yang akan di-*design* adalah fungsi (kecepatan, temperatur operasi, tekanan), keamanan (kekuatan, defleksi, getaran) dan lain-lain. Sebagai contoh untuk *machine design*, berbagai fungsi dan kendala yang harus dipertimbangkan ditunjukkan pada Gambar 2.3.



Gambar 2.3 Berbagai Jenis Kendala yang Perlu Dipertimbangkan dalam Perancangan

Setelah *problem* didefinisikan dan seluruh spesifikasi ditetapkan tahap berikutnya adalah *Synthesis*. Dalam tahap *Synthesis* semua kemungkinan alternatif solusi digali dan dipertimbangkan. Tahap *Synthesis* sering juga disebut tahap *ideation and invention* dimana di-*generate* kemungkinan solusi secara kreatif sebanyak mungkin.

Alternatif-alternatif rancangan yang didapatkan selanjutnya masuk ke dalam *analysis and optimization* untuk menentukan apakah rancangan tersebut dapat memenuhi spesifikasi dan performansi yang diinginkan, ditolak, atau perlu dimodifikasi. Tahap *analysis and optimization* menghasilkan rancangan yang paling optimum untuk dipilih. Jika analisis menunjukkan bahwa rancangan tidak memenuhi spesifikasi dan performansi yang diinginkan maka harus dilakukan iterasi. Hasil rancangan yang paling optimum dipilih dan selanjutnya dapat dilakukan *detailed design*. Dalam *detailed design* dihasilkan gambar teknik yang lengkap, spesifikasi material, identifikasi vendor, spesifikasi manufaktur, dan lainnya.

Evaluation merupakan salah satu tahapan penting dalam proses *design* secara keseluruhan. Tahap *evaluation* melibatkan pembuatan prototipe dan pengujian yang dapat dilakukan di laboratorium. Hasil pengujian prototipe inilah yang akan membuktikan apakah rancangan yang dihasilkan dapat memenuhi spesifikasi dan performansi yang diinginkan.

Tahap terakhir adalah *presentation*. Hasil rancangan perlu dikomunikasikan dengan proses selanjutnya seperti *manufacturing*, *assembling* dan sosialisasi. Komunikasi dapat dilakukan dalam tiga cara yaitu komunikasi secara tertulis, lisan, dan dalam bentuk grafik atau gambar. Dengan demikian insinyur harus menguasai ketiga teknik tersebut untuk dapat mempresentasikan rancangannya.

2.2 Konstruksi Atap

Konstruksi atap adalah bagian dari suatu bangunan yang berfungsi sebagai penutup seluruh ruangan yang ada dibawahnya terhadap pengaruh panas, hujan, angin, debu atau untuk keperluan perlindungan. Syarat-syarat atap yang harus dipenuhi antara lain:

1. Konstruksi atap harus kuat menahan beratnya sendiri dan tahan terhadap tekanan maupun tiupan angin,
2. Pemilihan bentuk atap yang akan dipakai hendaknya sedemikian rupa, sehingga menambah keindahan serta kenyamanan bertempat tinggal bagi penghuninya,
3. Rangka atap perlu diberi lapisan pengawet supaya tidak mudah dimakan rayap,
4. Bahan penutup atap harus tahan terhadap pengaruh cuaca, dan
5. Kemiringan atau sudut lereng atap harus disesuaikan dengan jenis bahan penutupnya maka kemiringannya dibuat lebih landai.

2.2.1 Macam-macam Atap

Berdasarkan konstruksinya atap memiliki berbagai macam bentuk yang disesuaikan dengan iklim, budaya dan daerah sekitar. Macam-macam bentuk atap tersebut adalah atap datar, atap sandar, atap pelana, atap tenda, atap menara, dan atap joglo.

a. Atap Datar

Atap datar adalah atap yang permukaan atap-nya datar. Meskipun bentuk atap ini dinamakan atap datar, akan tetapi kenyataannya permukaan atap datar selalu dibuat sedikit miring untuk menyalurkan air hujan ke lubang talang. Atap datar banyak digunakan pada rumah-rumah mewah seperti rumah bertingkat atau gedung-gedung pencakar langit. Bentuk fisik atap dapat dilihat pada gambar 2.4.



Gambar 2.4 Atap Datar

b. Atap Sandar

Atap sandar (atap *temple*) adalah atap yang terdiri dari sebuah bidang atap miring yang bagian tepi atasnya bersandar atau menempel pada tembok bangunan induk (tembok yang menjulang tinggi). Konstruksi atap sandar pada umumnya menggunakan jenis konstruksi setengah kuda-kuda dengan kemiringan tertentu. Untuk bahan penutup genteng sudut kemiringan atap sandar sekitar 30° atau 40°. Untuk bahan penutup dari semen asbes gelombang dan seng gelombang sudut kemiringan atapnya sekitar 20° atau 25°. Pada pemasangan atap berbahan semen asbes gelombang dan seng gelombang tidak memerlukan reng untuk menentukan jarak antara atap satu dengan atap yang lainnya. Bentuk fisik atap sandar dapat dilihat pada gambar 2.5.



Gambar 2.5 Atap Sandar

c. Atap Pelana

Atap pelana adalah atap yang terdiri dari dua bidang atap miring yang tepi atasnya bertemu pada satu garis lurus dan tepi bawahnya teritis. Pada tepi teritis atap pelana dipasangkan talang air yang berfungsi untuk menghimpun air hujan supaya air hujan tidak bertebaran ke segala arah. Talang air yang dipasang pada atap pelana umumnya berbahan dari genteng biasa (genteng kampung) atau seng gelombang. Atap rumah dengan bentuk atap pelana banyak dijumpai di pedesaan seperti Bali, Jawa Timur, Jawa Tengah dan Jawa Barat. Gambar fisik atap pelana dapat dilihat pada gambar 2.6.

d. Atap Tenda

Atap tenda adalah atap yang bentuk atap-nya menyerupai tenda. Atap ini banyak digunakan untuk bangunan kantor, pendopo, dan bangunan untuk tempat tinggal. Gambar fisik atap tenda dapat dilihat pada gambar 2.7.



Gambar 2.6 Atap Pelana



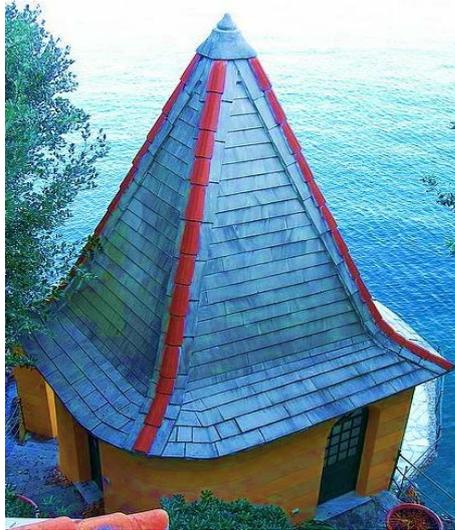
Gambar 2.7 Atap Tenda

e. Atap Menara

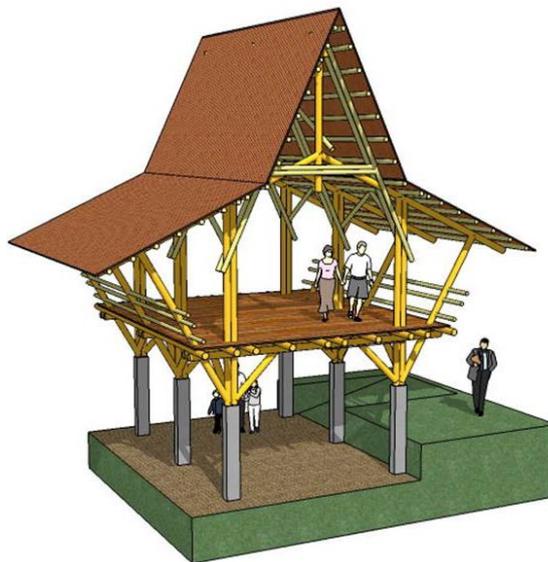
Atap menara adalah atap yang bentuk atap-nya menyerupai bentuk menara. Atap menara mempunyai empat bidang atap dengan sudut apitnya sama besar, serta ujung-ujung bagian atasnya saling bertemu pada satu titik yang cukup tinggi. Bentuk atap menara banyak ditemukan pada bangunan gereja dan rumah adat. Bentuk fisik atap menara dapat dilihat pada gambar 2.8.

f. Atap Joglo

Atap joglo adalah atap yang merupakan atap jurai luar yang patah ke dalam seolah-olah terdiri dari dua bagian yaitu bagian bawah yang mempunyai sudut lereng atap lebih kecil atau landai dan bagian atas tampak bagian-bagian bidang atap yang berbentuk trapesium. Bentuk fisik atap joglo dapat dilihat pada gambar 2.9.



Gambar 2.8 Atap Menara



Gambar 2.9 Atap Joglo

2.2.2 Struktur Atap

Struktur atap adalah bagian bangunan yang berfungsi menahan atau mengalirkan beban-beban dari atap. Struktur atap terbagi menjadi 2 (dua) bagian yaitu rangka atap dan penopang rangka atap. Rangka atap berfungsi menahan beban dari bahan penutup atap. Susunan rangka atap berupa balok-balok (dari kayu/bambu/baja) yang diposisikan secara vertikal dan horizontal. Berdasarkan susunan rangka atap muncul istilah gording, kasau dan reng. Susunan rangka atap dapat menghasilkan lekukan pada atap (jurai dalam/luar) dan menciptakan bentuk atap tertentu. Penopang rangka atap adalah balok kayu yang disusun membentuk segitiga atau disebut dengan istilah kuda-kuda. Kuda-kuda tersebut ditempatkan pada posisi bawah rangka atap, fungsinya untuk menyangga rangka-rangka atap. Bagian atas

kuda-kuda penyangga rangka atap disangkutkan pada balok bubungan, sementara kedua kakinya dihubungkan dengan kolom struktur untuk mengalirkan beban ke tanah.

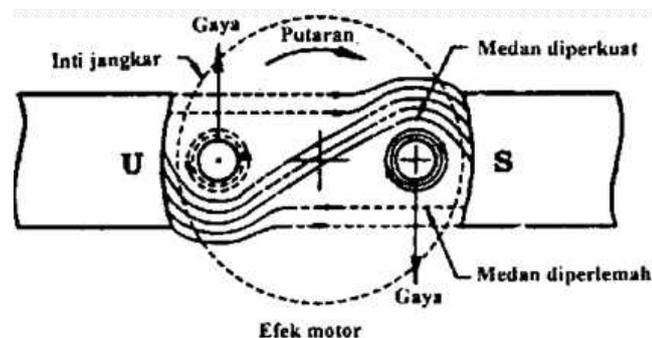
2.3 Motor DC

Motor DC adalah salah satu jenis motor listrik dengan suplai tegangan arus searah (DC) yang dapat mengubah energi listrik menjadi energi mekanik berupa gerak rotasi poros. Bentuk fisik motor DC dapat dilihat pada gambar 2.10.



Gambar 2.10 Bentuk Fisik Motor DC

Prinsip kerja motor DC yaitu jika ada arus listrik DC melewati kumparan dalam medan magnet (*fluks magnet*), maka akan timbul gaya elektromagnetik yang menghasilkan torsi pada poros. Skematis prinsip kerja motor DC dapat dilihat pada gambar 2.11.



Gambar 2.11 Skematis Prinsip Kerja Motor DC

2.4 Pulley

Pulley adalah salah satu dari berbagai macam transmisi yang berfungsi untuk mengkonversikan kecepatan (putaran) dari motor (mesin) ke penggerak akhir. Pengkonversian kecepatan (putaran) dari motor (mesin) menggunakan *pulley* dikenal dengan sebutan sistem *pulley*. Pada penggunaannya *pulley* selalu terdapat lebih dari 2 (dua) buah *pulley* yang dihubungkan dengan sabuk (*belt*).

Sistem *pulley* dengan sabuk sering digunakan (difungsikan) pada mesin sederhana seperti mesin kompresor, mesin diesel, dan mesin penggiling padi. Sistem *pulley* selain menggunakan sabuk (*belt*) sebagai penghubungnya, sistem *pulley* juga dapat dihubungkan menggunakan tali atau kabel. Bentuk fisik *pulley* dapat dilihat pada gambar 2.12.



Gambar 2.12 Pulley

2.5 Relay Modul

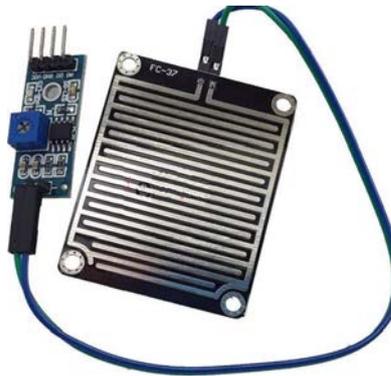
Relay modul adalah salah satu jenis relay yang dapat difungsikan sebagai saklar (kontak) suatu rangkaian elektronika. Relay modul pada umumnya dikompatibel dengan Raspberry Pi atau berbagai jenis Mikrokontroler, terutama Mikrokontroler Arduino. Raspberry Pi dan Mikrokontroler berfungsi untuk mengendalikan ON/OFF saklar (kontak) relay modul dengan cara mengirimkan sinyal ke *input* relay modul. Bentuk fisik relay modul dapat dilihat pada gambar 2.13.



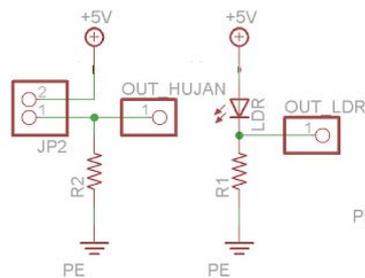
Gambar 2.13 Relay Modul

2.6 Sensor Hujan

Sensor hujan adalah salah satu jenis sensor yang terbuat dari beberapa rangkaian elektronika yang di dalamnya terdapat komponen utama berupa resistor. Resistor pada sensor hujan berfungsi sebagai pendeteksi air dengan memanfaatkan hambatan untuk mengetahui level air yang terdeteksi oleh sensor hujan. Hambatan resistor pada sensor hujan memiliki nilai *output* 5V (lima volt) dalam kondisi sensor hujan kering, dan akan menurun nilai *output*-nya apabila sensor hujan dalam kondisi basah. Bentuk fisik sensor hujan dan skematik rangkaian sensor hujan dapat dilihat pada gambar 2.14 dan gambar 2.15.



Gambar 2.14 Sensor Hujan



Gambar 2.15 Skematik Rangkaian Sensor Hujan

2.7 Sensor Cahaya (*Light Dependent Resistor*)

Sensor *Light Dependent Resistor* (*LDR*) adalah jenis resistor yang nilai hambatan atau nilai resistansinya tergantung pada intensitas cahaya yang diterima oleh sensor *LDR*. Pada umumnya sensor *LDR* terdapat 2 (dua) macam yang banyak digunakan dalam rangkaian elektronika. Macam sensor *LDR* tersebut adalah sensor *LDR* positif dan sensor *LDR* negatif.

a) Sensor *LDR* Positif

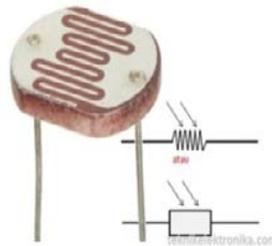
Prinsip kerja sensor *LDR* positif yaitu jika sensor *LDR* positif mendeteksi atau menerima cahaya maka nilai hambatannya kecil, dan begitu sebaliknya jika sensor tidak menerima cahaya maka nilai hambatannya menjadi besar, atau dengan kata lain fungsi sensor *LDR* positif adalah untuk menghantarkan arus listrik dalam kondisi terang dan menghambat arus listrik dalam kondisi gelap.

Sensor *LDR* positif mempunyai nilai hambatan dari 500 Ohm (Ω) sampai 200 Kilo Ohm ($k\Omega$). Nilai hambatan sensor *LDR* positif sewaktu-waktu dapat berubah-ubah atau naik turun. Naik turunnya nilai hambatan akan sebanding dengan jumlah cahaya yang diterima sensor *LDR* positif. Pada umumnya nilai hambatan yang terdapat pada sensor *LDR* positif mencapai 200 Kilo Ohm ($k\Omega$) pada kondisi gelap dan akan menurun hingga maksimal 500 Ohm (Ω) pada kondisi terang.

b) Sensor *LDR* Negatif

Prinsip kerja sensor *LDR* negatif yaitu jika sensor *LDR* negatif mendeteksi atau menerima cahaya maka nilai hambatannya besar, dan begitu sebaliknya jika sensor tidak menerima cahaya maka nilai hambatannya menjadi kecil, atau dengan kata lain fungsi sensor *LDR* negatif adalah untuk menghantarkan arus listrik dalam kondisi gelap dan menghambat arus listrik dalam kondisi terang.

Sensor *LDR* negatif mempunyai nilai hambatan dari 500 Ohm (Ω) sampai 200 Kilo Ohm ($k\Omega$). Nilai hambatan sensor *LDR* negatif sewaktu-waktu dapat berubah-ubah atau naik turun. Naik turunnya nilai hambatan akan sebanding dengan jumlah cahaya yang diterima sensor *LDR* negatif. Pada umumnya nilai hambatan yang terdapat pada sensor *LDR* negatif mencapai 200 Kilo Ohm ($k\Omega$) pada kondisi terang dan akan menurun hingga maksimal 500 Ohm (Ω) pada kondisi gelap. Bentuk fisik sensor cahaya *LDR* dapat dilihat pada gambar 2.16.



Gambar 2.16 Sensor Cahaya (*LDR*)

2.8 Arduino

Arduino adalah kit elektronika atau papan rangkaian elektronika *open source* yang di dalamnya terdapat komponen utama yaitu sebuah chip mikrokontroler dengan jenis AVR. Mikrokontroler AVR adalah chip atau IC (Integrated Circuit) yang dapat diprogram menggunakan komputer. Pemrograman pada mikrokontroler AVR bertujuan untuk memfungsikan mikrokontroler AVR agar dapat membaca *input*, memproses *input* dan kemudian menghasilkan *output* sesuai yang diinginkan. Bentuk fisik Arduino dapat dilihat pada gambar 2.17.

2.9 Arduino Software (*IDE*)

IDE merupakan singkatan dari *Integrated Development Environment*, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. *IDE* disebut sebagai lingkungan, karena melalui Arduino *software (IDE)* Arduino dapat diprogram untuk melakukan fungsi-fungsi yang dinamakan melalui sintak pemrograman. Sintak pemrograman Arduino menggunakan bahasa pemrograman sendiri yang

menyerupai bahasa C. Bahasa pemrograman Arduino sudah diubah untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. *Arduino software* sebelum dijual ke pasaran, telah ditanam suatu program bernama *Bootlader* yang berfungsi sebagai penengah antara *compiler* Arduino dengan mikrokontroller.



Gambar 2.17 Arduino

Arduino software (IDE) dibuat dari bahasa pemrograman JAVA. *Arduino software (IDE)* juga dilengkapi dengan library C/C++ yang biasa disebut *Wiring* yang membuat operasi *input* dan *output* menjadi lebih mudah. *Arduino software (IDE)* dikembangkan dari *software processing* yang dirombak menjadi *Arduino software (IDE)* khusus untuk pemrograman dengan Arduino. Gambar tampilan awal *Arduino software (IDE)* dapat dilihat pada gambar 2.18.



Gambar 2.18 Tampilan Awal *Arduino Software (IDE)*

2.10 *Solidworks*

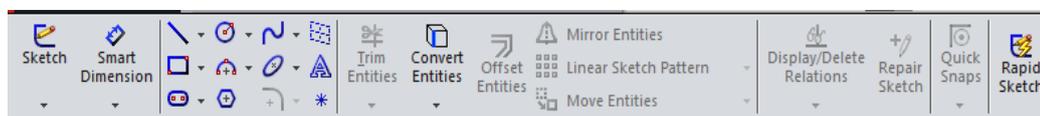
Solidworks adalah salah satu CAD *software* yang dibuat oleh *Dassault Systemes*. *Software Solidworks* digunakan untuk merancang *part* permesinan atau susunan *part* permesinan yang berupa *assembling* dengan tampilan 3D untuk merepresentasikan *part* sebelum *real part*-nya dibuat atau tampilan 2D (*drawing*) untuk gambar proses permesinan. *Solidworks* pertama kali diperkenalkan pada tahun 1995 sebagai pesaing untuk program CAD seperti *Pro-Engineer*, *NX Siemens*, *I-Deas*, *Unigraphics*, *Autodesk Inventor*, *Autodeks*

Autocad dan *Catia*. *Solidworks Corporation* didirikan pada tahun 1993 oleh Jon Hirschtick, dengan merekrut tim insinyur profesional untuk membangun sebuah perusahaan yang mengembangkan perangkat lunak CAD 3D, dengan kantor pusatnya di Concord, Massachusetts, dan merilis produk pertama *Solidworks 95* pada tahun 1995. Pada tahun 1997 *Dassault Systemes*, yang terdapat pada *Cad software* dikenal dengan *Catia Cad software*, mengakuisisi perusahaan dan sekarang ini memiliki 100% dari saham *Solidworks*. *Solidworks* dipimpin oleh John Mc.Elenev dari tahun 2001 hingga Juli 2007, dan sekarang dipimpin oleh Jeff Ray. menurut informasi WIKI Saat ini banyak industri manufaktur yang sudah memakai *software Soliworks*.

Solidworks saat ini digunakan oleh lebih dari 3/4 juta insinyur dan desainer di lebih dari 80.000 perusahaan di seluruh dunia. Dahulu di Indonesia orang familiar dengan *Autocad* untuk desain perancangan gambar teknik, tapi sekarang dengan mengenal *Solidworks*, *Autocad* sudah jarang digunakan untuk menggambar bentuk 3D. Untuk pemodelan pada industri pengecoran logam dalam hal pembuatan *pattern* (pola/model), program 3D yang terdapat pada *software Solidworks* sangat membantu dalam pekerjaan, sebab akan memudahkan operator *pattern* untuk menterjemahkan gambar menjadi *pattern/model casting* pengecoran logam dan tentunya akan mengurangi kesalahan pembacaan gambar yang bisa mengakibatkan kesalahan pada produk yang dihasilkan.

2.10.1 Menggambar 2D dengan *Basic Sketch*

Sketch merupakan dasar dalam pembuatan sebuah objek dasar pada *solidworks*. Pada menu *sketch* terdapat berbagai perintah dasar seperti *line*, *circle*, *arc*, *polygon* dan lain-lain. Baris *command* (perintah) pada menu *sketch* dikategorikan menjadi dua bagian pokok yaitu *command draw* dan *command modify*. Perintah (*command*) pada menu *sketch* ditunjukkan pada gambar 2.19.

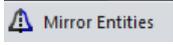


Gambar 2.19 Perintah *Sketch*

1. *Command Modify*

Command modify digunakan untuk membuat objek dengan cara memodifikasi *sketch* yang sudah ada. Beberapa perintah pokok yang dikategorikan dalam *command modify* dapat dilihat pada tabel 2.1.

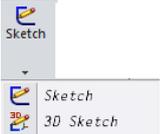
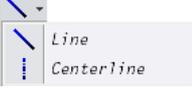
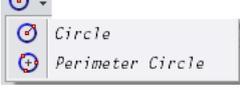
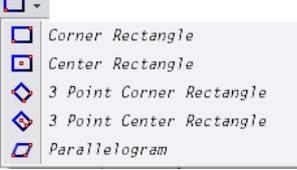
Tabel 2.1 Perintah pokok pada menu *modify*

| No | Toolbar | Command | Fungsi |
|----|---|------------------------|--|
| 1 |  | <i>Mirror Entities</i> | Membuat bentuk geometri yang dicerminkan dari bentuk yang dipilih dengan menentukan garis percerminan |
| 2 |  | <i>Offset Entities</i> | Membuat bentuk geometri yang sebangun dengan bentuk objek yang dipilih dengan menentukan jarak tertentu dari objek aslinya |

2. *Command Draw*

Command draw digunakan untuk membuat objek gambar pada *sketch*. Beberapa perintah pokok pada menu *draw* dapat dilihat pada tabel 2.2.

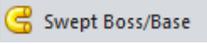
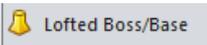
Tabel 2.2 Perintah pokok pada menu *draw*

| No | Toolbar | Command | Fungsi |
|----|---|---------------------------------|---|
| 1 |  | <i>Sketch</i> | Membuat 2D <i>sketch</i> |
| | | <i>3D sketch</i> | Membuat 3D <i>sketch</i> |
| 2 |  | <i>Smart dimension</i> | Memberikan ukuran dasar suatu <i>sketch</i> |
| 3 |  | <i>Line</i> | Membuat garis lurus |
| | | <i>Centerline</i> | Membuat garis tengah |
| 4 |  | <i>Circle</i> | Membuat bentuk lingkaran |
| | | <i>Perimeter Circle</i> | Membuat lingkaran berdasarkan titik acuan |
| 5 |  | <i>Corner Rectangle</i> | Membuat segi empat dengan dua titik diagonal |
| | | <i>Center Rectangle</i> | Membuat segi empat berdasarkan titik pusat acuan |
| | | <i>3 Point Corner Rectangle</i> | Membuat segi empat berdasarkan diagonal dengan 3 titik acuan |
| | | <i>3 Point Center Rectangle</i> | Membuat segi empat berdasarkan pusat bangun dan 2 titik acuan |
| | | <i>Parallelogram</i> | Membuat bangun jajar genjang berdasarkan 3 titik acuan |

2.10.2 Pembuatan Gambar Komponen 3D dengan *Part Modeling*

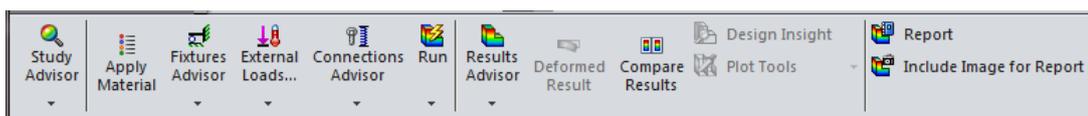
Part merupakan ruang gambar pemodelan 3D komponen suatu objek gambar. Dasar pembuatan *part* adalah gambar *sketch*. Setelah proses pembuatan *sketch* selesai, menu *sketch* secara otomatis akan beralih ke menu *features*. Beberapa *command* yang terdapat pada menu *features* dapat dilihat pada tabel 2.3.

Tabel 2.3 Beberapa *command* pada menu *features*

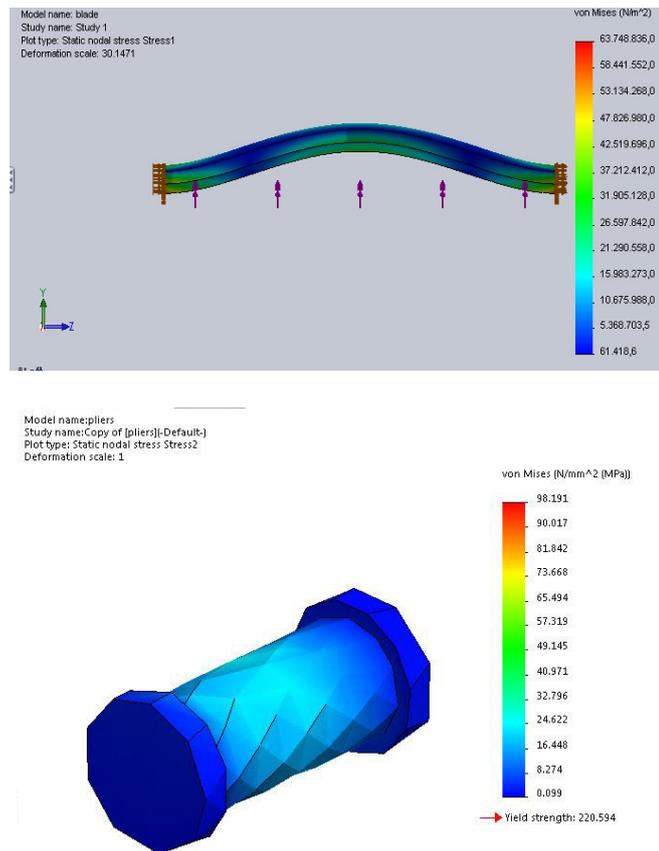
| No | Toolbar | Command | Fungsi |
|----|---|--------------------------|---|
| 1 | <i>Extrude</i>  | <i>Extrude Boss/Base</i> | Memberikan tinggi, tebal, atau kedalaman dari sebuah profil tertutup dengan ukuran tertentu |
| 2 | <i>Revolve</i>  | <i>Revolve Boss/Base</i> | Membuat bentuk silindris dengan cara memutar suatu bentuk profil terhadap sumbu yang ditentukan |
| 3 | <i>Swept</i>  | <i>Swept Boss/Base</i> | Membuat objek yang terbentuk dari <i>sketch</i> atau profil melalui garis edar |
| 4 | <i>Loft</i>  | <i>Loft Boss/Base</i> | Membuat objek dengan perpaduan beberapa bentuk atau potongan yang berbeda |

2.10.3 Proses *Solidworks Simulation*

Solidworks simulation adalah salah satu toolbar *solidworks* yang berfungsi meragakan benda kerja yang telah dirancang (digambar) dalam *solidworks* yang bertujuan untuk mengetahui sifat mekanik benda kerja seperti kekuatan benda kerja, tegangan benda kerja, ketangguhan benda kerja, kekuatan luluh benda kerja, kekerasan benda kerja, dan faktor keamanan (*safty factor*) benda kerja. *Solidworks simulation* selain dapat mengetahui sifat mekanik benda kerja, *solidworks simulation* juga dapat mengetahui aliran fluida (*flow effect*) dan perpindahan panas (*thermal effect*) benda kerja. Perintah *solidwork simulation* pada menu *office products* ditunjukkan pada gambar 2.20 dan contoh benda kerja yang telah dilakukan *solidwork simulation* dapat dilihat pada gambar 2.21.



Gambar 2.20 Perintah *Solidworks Simulation*



Gambar 2.21 Contoh Benda Kerja yang Telah Dilakukan *Solidwork Simulation*

2.11 *Sim Wise 4D*

SimWise 4D adalah *Software* yang digunakan untuk mensimulasikan gambar yang didesain dari *Software Solidworks*. Gambar yang didesain dari *Software Solidworks* agar dapat disimulasikan, gambar tersebut dikonversi ke *SimWise 4D* terlebih dahulu, kemudian gambar dapat disimulasikan dengan langkah-langkah berikutnya.

SimWise 4D difungsikan untuk mengetahui poin-poin pada gambar yang disimulasikan. Poin-poin tersebut berupa *Stress Analysis*, *Constraint Tension*, *Constraint Force*, *Constraint Torque*, *Constraint Length*, dan *Constraint Displacement*. Tampilan awal *SimWise 4D* dapat dilihat pada gambar 2.22.



Gambar 2.22 Tampilan Awal *Sim Wise 4D*