BAB II

TINJAUAN PUSTAKA

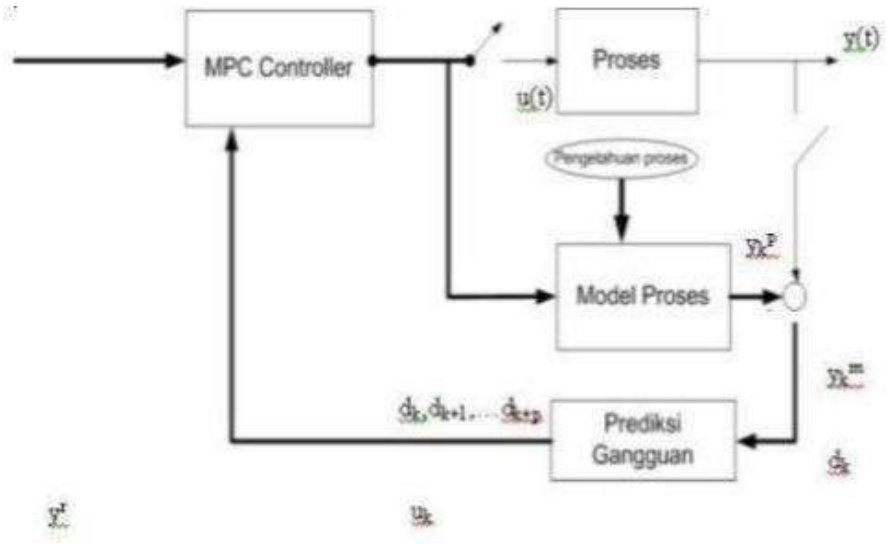
II.1 Landasan Teori

Landasan teori adalah teori-teori yang relevan dan dapat digunakan untuk menjelaskan variabel-variabel penelitian. Landasan teori ini juga berfungsi sebagai dasar untuk memberi jawaban sementara terhadap rumusan yang diajukan serta membantu dalam penyusunan *instrument* penelitian.

II.1.1 Pengenalan Algoritma *Model Predictive Control*

Gambar II.1 identifikasi model proses berikut akan menjelaskan secara umum langkah kerja dari proses identifikasi. Dalam beberapa kasus, model proses yang didapat terkadang tidak sesuai dengan kondisi proses sesungguhnya. Akibatnya, optimalisasi dari pengontrol *MPC* sangat bergantung dari kualitas model proses yang didapat. Sehingga terkadang perlu dilakukan identifikasi proses secara berulang untuk dapat memastikan akurasi model proses yang dihasilkan. Walaupun demikian, akurasi model proses tidak akan pernah mencapai kesempurnaan dikarenakan teknik linierisasi suatu elemen proses yang taklinier. Untuk lebih menyempurnakan model proses yang dihasilkan, diperlukan perbaikan model proses identifikasi dengan pengetahuan tentang proses sebenarnya. Model proses identifikasi dilinierisasi dengan model *Finite Impulse Response* (FIR) dan *Auto-Regressive* (ARX) untuk respon masukan step. Perbaikan model dilakukan dengan membandingkan hasil yang didapat pada

model FIR dan ARX serta menggabungkan data informasi pengetahuan tentang proses. [4]



Gambar II.1 Identifikasi Model Proses

II.1.1.1 Langkah - langkah algoritma *Model Predictive Control*

Saat pengidentifikasian proses dilakukan, pengontrol MPC akan bekerja berdasarkan model proses yang dihasilkan. Dari alur diagram blok berikut ini, yang terpenting adalah kemampuan pengontrol MPC dalam mengestimasi adanya gangguan untuk keadaan sekarang dan selanjutnya, sehingga kualitas (*performance*) dari sistem kontrol dapat terjaga pada kondisi kestabilan yang maksimal.

Bahasan berikut, menekankan bagaimana perancangan dan tahap-tahapan implementasi strategi kontrol *Model Predictive Control*.

1. Analisis Proses

Tujuan melakukan kajian ilmiah mengenai objek *plant* yang akan dikontrol adalah mendapatkan persamaan dinamika dari proses. Analisis proses akan memberikan formulasi yang jelas dari objektif sistem kontrol dan keterbatasan (*limitations*) yang sesuai dengan pengertian MPC sebenarnya. Atau dengan kata lain, analisis proses akan memberikan konfigurasi kontrol proses dalam terminologi MPC. *Input* dan *Output* proses digolongkan dalam 4 katagori yang berbeda. Penggolongan ini berdasarkan penggunaannya dalam pengontrolan proses:

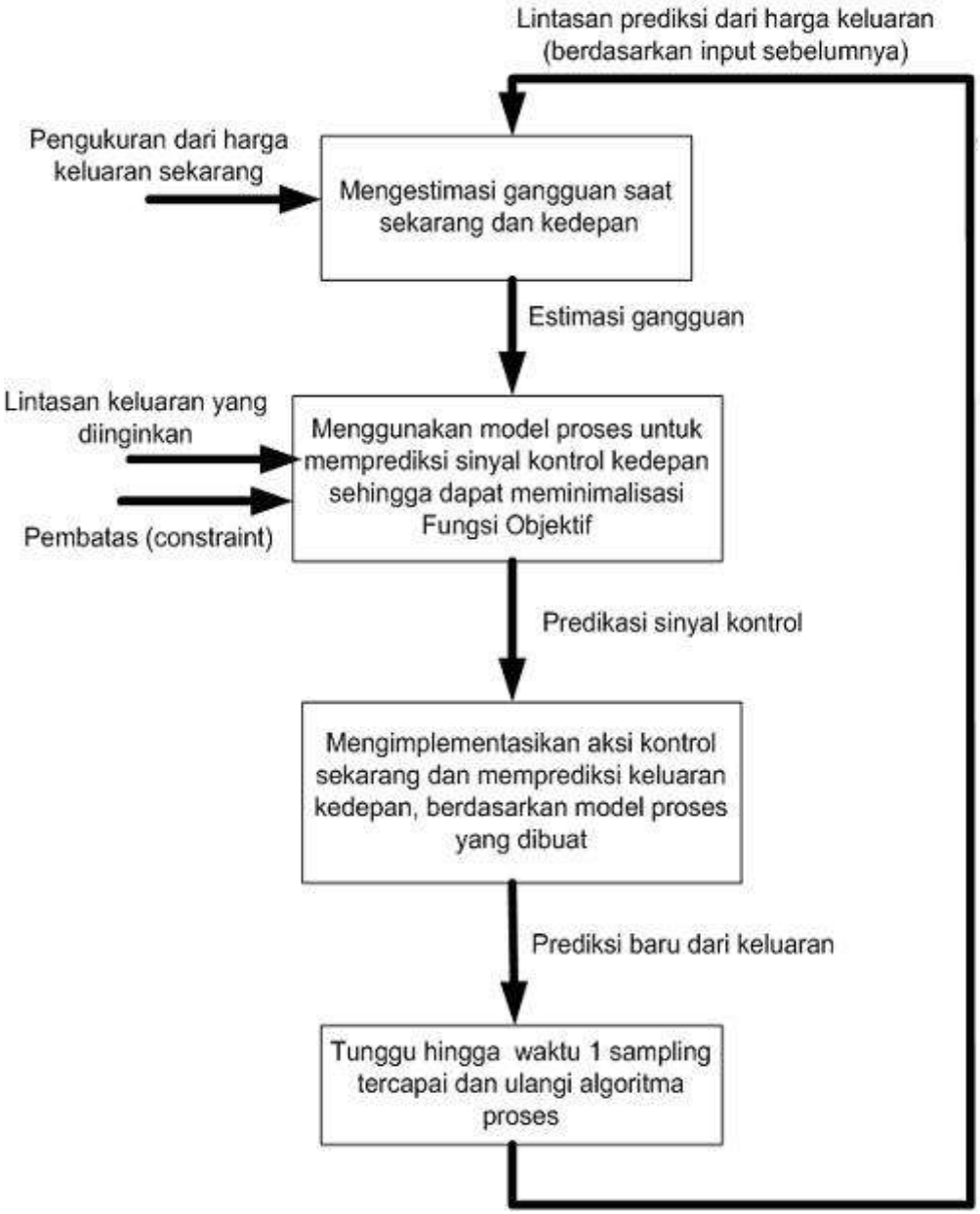
a. *Manipulated Variabel* (MV) – input proses yang diatur untuk menjaga output proses tetap pada setpoint.
b. *Controller Variabel* (CV) -- output proses yang ingin dijaga pada harga setpoint.
c. *Disturbance Variabel* (DV) – input proses yang memberi efek terhadap proses dan output variabel tertentu.
d. *Constraint* (AV) – output proses yang harus dijaga dalam suatu definisi batas jangkauan operasi yang berlawanan dengan setpoint.Proses input dan output dapat juga didefinisikan sebagai *teroptimisasi* jika pada proses input atau output tersebut terdapat sebuah pertimbangan ekonomis atau kinerja untuk menekan proses variabel dalam arah yang diinginkan hingga beberapa *constraint* (batasan) proses menjadi aktif. Pada langkah ini juga akan didefinisikan variabel *Manipulated (MV) , Disturbance (DV),*

Controlled (CV) dan *Constraint (AV)* yang berkaitan dengan proses yang ingin dikontrol.

2. Membuat modul program MPC. Modul dapat berisikan informasi algoritma, kondisi, *alarm, display, historical information*, dan karakteristik lainnya yang menyatakan perlengkapan proses.
3. Setelah modul program dirancang dalam suatu area *project,* langkah selanjutnya adalah *asignment* program ke data *historian* yang terdapat dalam memori sebuah mikro yang disimulasikan menjadi perangkat lunak.
4. Melakukan tes terhadap proses dan data yang didapat akan disimpan secara otomatis dalam *data historian*. Berdasarkan data-data yang tersimpan dalam *data historian,* selanjutnya akan dibangkitkan model dari proses dan sinyal kontrol prediksi.
5. Setelah identifikasi model proses dan membangkitkan sinyal kontrol prediksi, dilakukan perbaikan model sehingga lebih mendekati model proses yang sebenarnya.
6. Melakukan tes simulasi untuk memastikan algoritma program sudah berjalan dengan benar. Jika unjuk kerja respon pengontrolan kurang baik, dapat dilakukan *tuning* parameter pengontrol sehingga didapatkan respon terbaik.

Langkah ketujuh ditandai dengan garis terputus-putus, dimaksudkan jika dikehendaki adanya *commisioning* pada proses sebenarnya (*real plant*) maka dilakukan proses *download* dan menjalankan *controller*. Pada penerapan ini, langkah yang dilakukan hanya sebatas pada *point* 6 saja (simulasi) dan setelahnya dilakukan evaluasi terhadap perancangan pengontrolan dan simulasi yang dibuat.

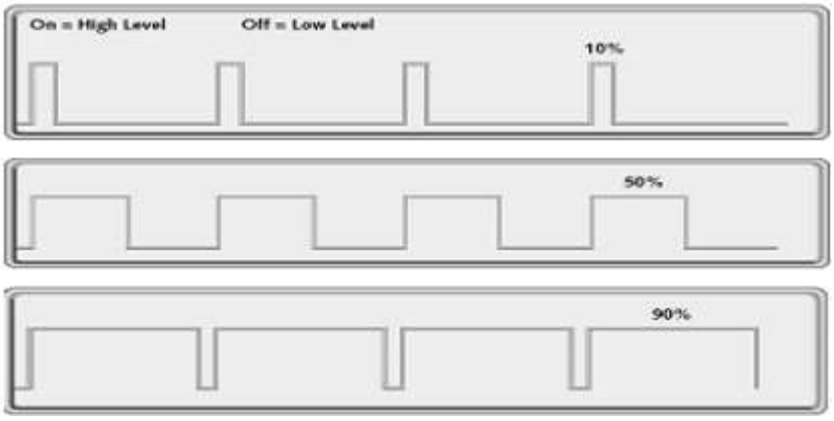
Gambar II.2 adalah bentuk umum algoritma *Model Predictive Control.*



Gambar II.2 Bentuk umum algoritma MPC

II.1.2 Pengertian *Pulse Width Modulation*

Pulse Width Modulation (*PWM*) merupakan suatu cara proses pengaturan kecepatan secara digital yang digunakan pada motor *DC* dengan memberikan pulsa - pulsa untuk waktu on dan *off* atau yaitu sebuah cara pengalihan daya dengan menggunakan sistem lebar pulsa untuk mengemudikan kecepatan putaran motor *DC*, jadi sebenarnya yang diatur adalah rasio waktu pemberian tegangan kepada motor *DC*. Perbandingan panjang waktu on (high) yang lebih lama dari pada waktu *off* (*low*) akan membuat motor *DC* berputar lebih cepat. Waktu periode dapat terjadi pada saat T_{on} dan T_{off} mempunyai frekuensi yang sama pada kecepatan yang berbeda. Gambar II.3 adalah gambar perbandingan pulsa high dan low pada gelombang digital.



Gambar II.3 Perbandingan pulsa *high* dan *low* pada gelombang digital

Semakin rapat periode antar pulsa, maka frekuensi yang dihasilkan akan semakin tinggi, dengan ini berarti kecepatan akan dapat bertambah. Semakin lebar jarak antar pulsa, maka frekuensi semakin rendah ini berarti kecepatan berkurang atau menurun. Kondisi pemberian kecepatan harus disesuaikan dengan

kondisi lintasan sekitar yang akan dilewati. Misal pada saat jalan lurus, naik atau turun tentunya harus mendapatkan nilai PWM yang cukup tepat.

II.1.2.1 Pengenalan Optocoupler

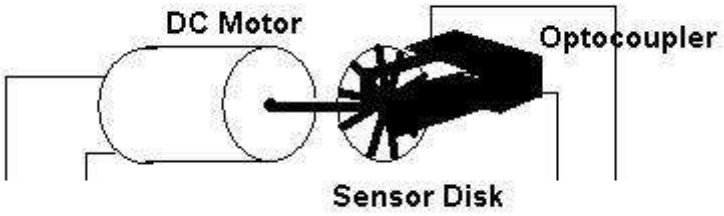
Opto berarti optik dan *coupler* berarti pemicu. Jadi, dapat diartikan bahwa *optocoupler* adalah komponen yang bekerja berdasarkan picu cahaya optik. *Optocoupler* termasuk dalam sensor yang terdiri dari dua bagian, yaitu pemancar dan penerima. *Optocoupler* adalah kombinasi dari LED inframerah dengan fototransistor yang dibungkus ke dalam satu kemasan tunggal. Gambar II.4 adalah contoh bentuk fisik *optocoupler*.



Gambar II.4 *Optocoupler*

Bagian pemancar *(transmitter)* dibangun dari sebuah *LED* inframerah untuk mendapatkan ketahanan yang lebih baik daripada menggunakan *LED* biasa. Sinar inframerah termasuk pada gelombang elektromagnetik yang tidak terlihat oleh mata. Cahaya ini tidak muncul ke mata karena panjang gelombang cahaya ini terlalu panjang untuk respon mata manusia. Sinar infra merah memiliki wilayah frekuensi 1 x 1012 Hz sampai dengan 1 GHz atau 1014 x frekuensi wilayah dengan panjang gelombang 1μm-1mm. Cahaya inframerah yang terkandung dalam *optocoupler* tidak perlu lensa untuk memfokuskan cahaya karena pemancarnya memiliki jarak yang dekat ke penerima.

Bagian *optocoupler* yang berfungsi sebagai penerima *(receiver)* adalah fototransistor. Fototransistor adalah komponen elektronik yang berfungsi sebagai detektor cahaya inframerah. Fototransistor mengubah cahaya menjadi sinyal listrik. Sinyal listrik (arus) pada input menjadi sinyal optik dengan menggunakan sumber cahaya yaitu LED dan sinyal optik tersebut dapat diterima detektor untuk diubah menjadi sinyal listrik kembali. Gambar II.5 adalah gambar blok *optocoupler* untuk kontrol motor *DC*



Gambar II.5 Blok *Optocoupler* untuk kontrol Motor *DC*

Penggunaan *optocoupler* dapat diterapkan untuk mendeteksi perputaran motor. Dengan meletakkan piringan sensor pada celah *optocoupler*, maka kita akan mendapatkan data kecepatan putaran dari setiap roda. Agar semakin presisi pembacaan datanya, piringan sensor tersebut dibuat selang-seling berwarna hitam dan transparan. Prinsip kerjanya adalah jika antara transistor dan LED dihalangi atau mendapatkan warna hitam pada piringan, maka fototransistor akan ON sehingga *output* dari kolektor akan berlogika *high*. Sebaliknya jika antara transistor dan LED tidak dihalangi atau mendapatkan warna putih pada piringan, maka transistor akan OFF sehingga *output* akan berlogika *low*.

II.1.3 Tool Analisis

Pada tingkat teknik, rekayasa perangkat lunak dimulai dengan serangkaian tugas pemodelan yang membawanya kepada suatu spesifikasi lengkap dari persyaratan representasi desain yang komprehensif bagi perangkat lunak yang akan dibangun. Model analisis, yang sebenarnya merupakan serangkaian model representasi teknis dari sistem. Saat ini ada dua yang mendominasi *landscape* pemodelan analisis. Yang pertama analisis terstruktur, adalah pemodelan klasik dan yang kedua adalah analisis berorientasi objek.

II.1.3.1 *DFD*

DFD adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut.

DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau dimana data tersebut akan disimpan.

DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur. Kelebihan utama pendekatan alir data, yaitu :

1. Kejelasan dari menjalankan implementasi teknis sistem.
2. Pemahaman lebih jauh mengenai keterkaitan satu sama lain dalam sistem dan subsistem.
3. Mengkomunikasikan pengetahuan sistem yang ada dengan pengguna melalui diagram alir data.
4. Menganalisis sistem yang diajukan untuk menentukan apakah data-data dan proses yang diperlukan sudah ditetapkan.

DFD terdiri dari *context diagram* dan diagram rinci (*DFD leveled*), *context diagram* berfungsi memetakan model lingkungan (menggambarkan hubungan antara entitas luar, masukan dan keluaran sistem), yang direpresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem. *DFD leveled* menggambarkan sistem jaringan kerja antara fungsi yang berhubungan satu sama lain dengan aliran data penyimpanan data, model ini hanya memodelkan sistem dari sudut pandang fungsi.

Beberapa simbol yang akan digunakan di dalam *DFD* antara lain adalah sebagai berikut :

1. Kesatuan luar (*External Entity*)

   Setiap sistem pasti mempunyai batas sistem yang memisahkan suatu sistem dengan lingkungan luarnya. Sistem akan menerima *input* dan menghasilkan *output* kepada lingkungan luarnya. Kesatuan luar (*external entity*) merupakan kesatuan dilingkungan luar sistem dapat berupa orang, organisasi atau sistem

lainnya yang berada dilingkungan luarnya yang akan memberikan *input* atau menerima *output* dari sistem. Kesatuan luar ini kebanyakan adalah salah satu dari berikut ini :

a. Suatu kantor, departemen atau divisi dalam perusahaan tetapi di luar sistem yang sedang dikembangkan.
b. Orang atau sekelompok orang di organisasi tetapi di luar sistem yang sedang dikembangkan.
c. Suatu organisasi atau orang di luar organisasi.
d. Sistem informasi yang lain di luar sistem yang sedang dikembangkan.
e. Sumber asli dari suatu transaksi.
f. Penerimaan akhir dari suatu laporan yang dihasilakn oleh sistem.

2. Aliran Data (*Data Flow*)

   Aliran data di *DFD* diberi simbol suatu panah. Aliran data ini mengalir diantara proses (*process*), simpan data (*data store*) dan kesatuan luar (*external entity*). Aliran data ini menunjukkan aliran dari data yang dapat berupa masukkan untuk sistem atau hasil dari proses sistem.

3. Proses

   Proses adalah kegiatan atau kerja yang dilakukan oleh orang, mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses yang digambarkan secara umum. Suatu proses dapat ditunjukkan dengan simbol lingkaran atau dengan simbol empat persegi panjang tegak dengan sudut – sudutnya tumpul.

4. Berkas atau Simpanan Data (*Data Store*)

   Berkas atau simpanan data merupakan simpanan dari data yang dapat berupa :

   a. Suatu file atau database di sistem komputer.

   b. Suatu arsip atau catatan manual.

   c. Suatu kotak tempat data di meja seseorang.

   d. Suatu tabel acuan manual.

II.1.3.2 *ERD*

ERD merupakan notasi grafis dalam pemodelan data konseptual yang mendeskripsikan hubungan antara penyimpanan. *ERD* digunakan untuk memodelkan struktur data dan hubungan antar data, karena hal ini relatif kompleks. Dengan *ERD* kita dapat menguji model dengan mengabaikan proses yang harus dilakukan. *ERD* menggunakan sejumlah notasi dan simbol untuk menggambarkan struktur dan hubungan antar data, pada dasarnya ada 3 macam simbol yang digunakan yaitu :

1. *Entity*

   Entity adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai, sesuatu yang penting bagi pemakai dalam konteks sistem yang akan dibuat.

2. Atribut

   Entiti mempunyai elemen yang disebut atribut, dan berfungsi mendeskripsikan karakter entiti.

3. Hubungan

Relationship sebagaimana halnya entiti maka dalam hubungan pun harus dibedakan antara hubungan atau bentuk hubungan antar entiti dengan isis dari hubungan itu sendiri. Relasi antara dua *file* atau dua tabel dapat dikategorikan menjadi tiga macam, yaitu :

a. One to One Relationship

Yang berarti entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas B, dan begitu juga sebaliknya setiap entitas pada himpunan entitas B berhubungan paling banyak dengan satu entitas pada himpunan entitas A.

b. One to Many Relationship

Yang berarti entitas pada himpunan entitas A berhubungan dengan banyak entitas pada satu himpunan entitas B, tetapi tidak sebaliknya setiap entitas pada himpunan entitas B berhubungan paling banyak dengan satu entitas pada himpunan entitas A.

c. Many to Many Relationship

Yang berarti entitas pada himpunan entitas A berhubungan dengan banyak entitas pada satu himpunan entitas B dan begitu juga sebaliknya setiap entitas pada himpunan entitas B berhubungan dengan banyak entitas pada himpunan entitas A.

II.1.4 Pengertian robot

Robot adalah sebuah alat mekanik yang dapat melakukan tugas fisik, baik menggunakan pengawasan dan kontrol manusia, ataupun menggunakan program yang telah didefinisikan terlebih dulu (kecerdasan buatan). Robot biasanya digunakan untuk tugas yang berat, berbahaya, pekerjaan yang berulang dan kotor. Biasanya kebanyakan robot industri digunakan dalam bidang produksi. Penggunaan robot lainnya termasuk untuk pembersihan limbah beracun, penjelajahan bawah air dan luar angkasa, pertambangan, pekerjaan "cari dan tolong" (*search and rescue*), dan untuk pencarian tambang. Dewasa ini robot mulai memasuki pasaran konsumen di bidang hiburan, dan alat pembantu rumah tangga, seperti penggunaan dalam alat yang membantu untuk penyedot debu, dan alat yang membantu untuk pemotong rumput.

II.1.4.1 Macam – macam robot

Secara fisik umumnya robot dalam dunia pendidikan digunakan sebagai alat bantu menerangkan prinsip ilmu alam atau *engineering*. Tetapi sebenarnya robot dapat berfungsi selain dari itu. Robot dapat membantu untuk membuat perencanaan, pengerjaan, pencapaian sesuatu, keyakinan, serta mengekspresikan kreativitas.

Berikut ini akan dibahas macam – macam bentuk robot pada umumnya :

1. Robot *Mobile*



Gambar II.6 Robot Beroda

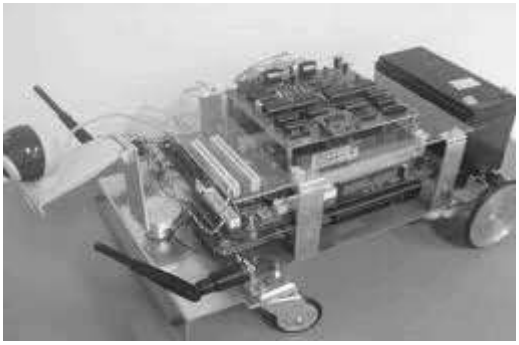
Gambar II.6 adalah gambar robot beroda. Robot Mobil atau *Mobile* Robot adalah konstruksi robot yang ciri khasnya adalah mempunyai aktuator berupa roda untuk menggerakkan keseluruhan badan robot tersebut, sehingga robot tersebut dapat melakukan perpindahan posisi dari satu titik ke titik yang lain.

Robot mobil ini sangat disukai bagi orang yang mulai mempelajari robot. Hal ini karena membuat robot mobil tidak memerlukan kerja fisik yang berat. Untuk dapat membuat sebuah robot mobile minimal diperlukan pengetahuan tentang mikrokontroler dan sensor-sensor elektronik.

Base robot mobil dapat dengan mudah dibuat dengan menggunakan *plywood* atau *triplek*, akrilik hingga menggunakan logam (aluminium). Robot mobil dapat dibuat sebagai pengikut garis (*Line Follower*) atau pengikut dinding (*Wall Follower*) ataupun pengikut dinding.

2. Robot jaringan

Robot jaringan adalah pendekatan baru untuk melakukan kontrol robot menggunakan jaringan internet dengan protokol TCP/IP. Perkembangan robot jaringan dipicu oleh kemajuan jaringan dan internet yang pesat. Dengan koneksi jaringan, proses kontrol dan monitoring, termasuk akuisisi data bila ada, seluruhnya dilakukan melalui jaringan. Keuntungan lain, koneksi ini bisa dilakukan secara nirkabel. Gambar II.7 adalah gambar robot jaringan.



Gambar II.7 Robot Jaringan

Di Indonesia, pengembang robot jaringan belum banyak, meski pengembang dan komunitas robot secara umum sudah banyak. Hal ini disebabkan tuntutan teknis yang jauh lebih kompleks. Salah satu robot jaringan yang sudah berhasil dikembangkan adalah LIPI *Wireless* Robot yang dikembangkan oleh Grup Fisika Teoritik dan Komputasi–GFTK LIPI.

Seperti ditunjukkan di LIPI *Wireless* Robot, seluruh proses kontrol dan monitoring bisa dilakukan melalui perambah internet. Lebih jauh, seluruh sistem dan protokol yang dikembangkan untuk LIPI *Wireless* Robot ini telah

dibuka sebagai open-source dengan lisensi GNU *Public License* (GPL) oleh *SourceForge* dengan nama openNR.

3. Robot Manipulator (tangan)



Gambar II.8 Robot Tangan

Gambar II.8 adalah gambar robot tangan. Robot bertangan hanya memiliki satu tangan namun fungsinya tidak jauh seperti tangan manusia yang fungsinya untuk memegang atau memindahkan barang, contoh robot ini adalah robot las di Industri mobil, robot merakit elektronik.

4. Robot Humanoid



Gambar II.9 Robot Humanoid

Gambar II.9 adalah gambar robot humanoid. Robot yang memiliki kemampuan menyerupai manusia, baik fungsi maupun cara bertindak, contoh

robot ini adalah *Ashimo* yang dikembangkan oleh *Honda*. Robot adalah sebuah alat mekanik yang dapat melakukan tugas fisik, baik menggunakan pengawasan dan kontrol manusia, ataupun menggunakan program yang telah didefinisikan terlebih dulu (kecerdasan buatan). Robot biasanya digunakan untuk tugas yang berat, berbahaya, pekerjaan yang berulang dan kotor. Biasanya kebanyakan robot industri digunakan dalam bidang produksi. Penggunaan robot lainnya termasuk untuk pembersihan limbah beracun, penjelajahan bawah air dan luar angkasa, pertambangan, pekerjaan "cari dan tolong" (search and rescue), dan untuk pencarian tambang. Belakangan ini robot mulai memasuki pasaran konsumen di bidang hiburan, dan alat pembantu rumah tangga, seperti penyedot debu, dan pemotong rumput.

5. Robot Berkaki



Gambar II.10 Robot Berkaki

Gambar II.10 adalah gambar robot berkaki. Robot berkaki memiliki kaki seperti hewan atau manusia, yang mampu melangkah, seperti robot serangga, robot kepiting, baik untuk berkaki enam maupun delapan.

6. *Flying* Robot (Robot Terbang)



Gambar II.11 Robot Terbang

Gambar II.11 adalah gambar robot yang memiliki kemampuan dapat terbang diudara. Robot yang mampu terbang ini menyerupai pesawat model yang di program khusus untuk memonitor keadaan di tanah dari atas, dan juga untuk meneruskan komunikasi.

7. *Under Water* Robot (Robot dalam air)



Gambar II.12 Robot dalam air

Gambar II.12 adalah gambar robot dalam air. Robot ini digunakan di bawah laut untuk memonitor kondisi bawah laut dan juga untuk mengambil sesuatu di bawah laut.

Ada beberapa unjuk kerja robot yang perlu diketahui, antara lain:

a. Resolusi, adalah perubahan gerak terkecil yang dapat diperintahkan oleh sistem kontrol pada lingkup kerja manipulator.

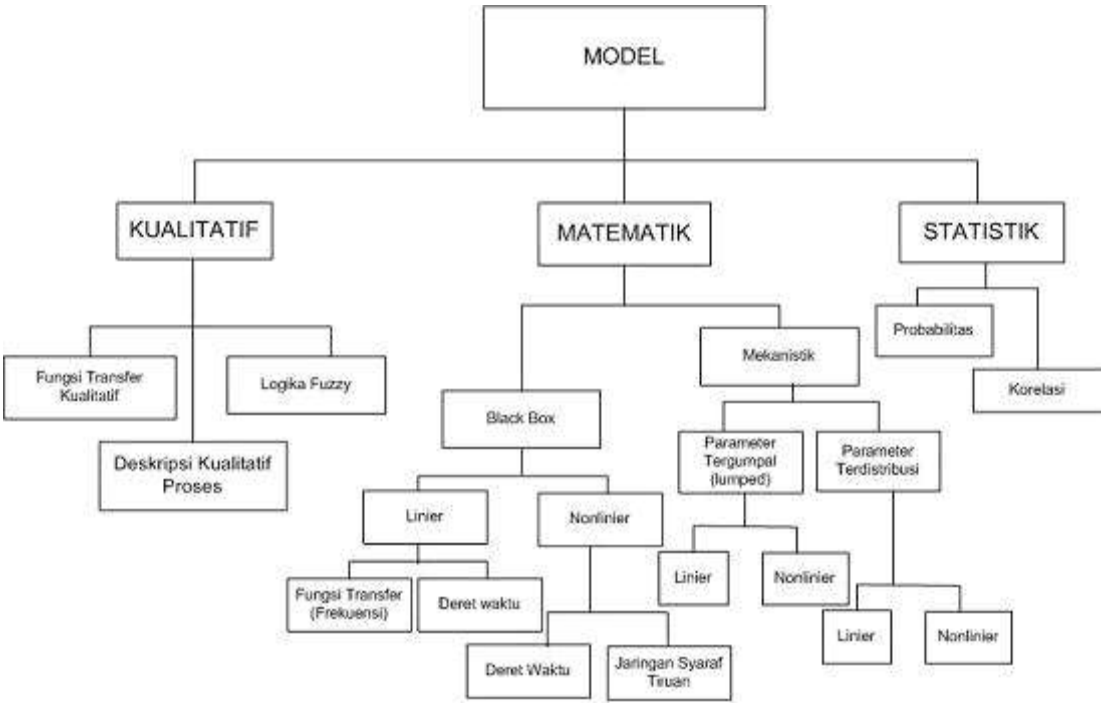
b. Akurasi adalah besarnya penyimpangan/deviasi terhadap masukan yang diketahui

c. *Repeatability* adalah kemampuan robot untuk mengembalikan *end effector* (pemegang/*griper*) pada posisinya semula

d. Fleksibilitas merupakan kelebihan yang dimiliki oleh robot secara umum jika dibandingkan dengan mesin konvensional. Hal inipun tergantung kepada pemrogram dalam merencanakan pola geraknya.

II.1.5 Pengertian model proses

Model suatu sistem dapat direpresentasikan dalam suatu bentuk ekspresi matematik. Ekspresi matematik tersebut dapat diturunkan dari hukum-hukum fisika maupun dari hubungan output terhadap input yang diberikan.Berdasarkan ekspresi matematik tersebut, maka informasi dinamik serta kondisi *steadystate* (*ajeg*/*mantap*) dari suatu sistem dapat diperkirakan. Informasi-informasi ini berguna bagi seorang sistem enjiner dalam melakukan analisis dan teknik disain suatu sistem.[4]



Gambar II.13 Bagan model Proses

Dalam hubungannya dengan sistem kontrol, sebuah model harus berisi informasi prediksi perilaku suatu sistem sebagai konsekuensi dari perubahan kondisi operasi suatu proses. Keterkaitannya dengan konteks ini,berarti sebuah model dapat berupa fungsi matematika maupun informasi statistik yang dapat menjelaskan aspek khusus suatu proses tertentu. Selain tinjauan matematika dan statistik, sebuah model dapat juga dipandang sebagai deskripsi kualitatif dari perilaku suatu proses. Pengklasifikasian ini ditunjukkan pada gambar II.13 bagan model proses.

II.1.6 Pengertian *Embedded Software*

Embedded software adalah sistem komputer dengan tujuan khusus, yang seluruhnya dimasukkan ke dalam alat yang di kontrol. *Embedded software* menunjukkan bahwa sistem ini merupakan bagian yang tidak dapat berdiri sendiri. Sebuah e*mbedded software* memiliki kebutuhan tertentu dan melakukan tugas yang telah diset sebelumnya, tidak seperti komputer pribadi serba guna. Contoh sistem atau aplikasinya antara lain adalah instrumentasi medik, *process control*, *automated vehicles control*, dan perangkat komunikasi. *Embedded software* biasanya diimplementasikan dengan menggunakan mikrokontroler.

II.1.7 Pengenalan Mikroprocessor

Sebuah mikroprosesor (sering dituliskan: µP) adalah sebuah *central processing unit* (CPU) komputer elektronik yang terbuat dari transistor mini dan sirkuit lainnya di atas sebuah sirkuit terintegrasi semikonduktor.

Berikut adalah karakteristik penting dari mikroprosesor :

1. Ukuran bus data internal (*internal data bus size*): Jumlah saluran yang terdapat dalam mikroprosesor yang menyatakan jumlah *bit* yang dapat ditransfer antar komponen di dalam mikroprosesor.
2. Ukuran bus data eksternal (*external data bus size*): Jumlah saluran yang digunakan untuk *transfer* data antar komponen antara mikroprosesor dan komponen-komponen di luar mikroprosesor.
3. Ukuran alamat memori (*memory address size*): Jumlah alamat memori yang dapat dialamati oleh mikroprosesor secara langsung.
4. Kecepatan clock (*clock speed*): *Rate* atau kecepatan *clock* untuk menuntun kerja mikroprosesor.
5. Fitur-fitur spesial (*special features*): Fitur khusus untuk mendukung aplikasi tertentu seperti fasilitas pemrosesan *floating point*, multimedia dan sebagainya.

II.2 Pengenalan Borland *Delphi* 7

Bahasa pemrograman *Delphi* merupakan pemrograman *visual* (berbasis windows) yang dibuat oleh sebuah Perusahaan *Software Borland .Inc*. Bahasa pemrograman *Delphi* merupakan bahasa pemrograman yang dikembangkan dari bahasa pemrograman *Pascal*. Fungsi dari aplikasi ini sama dengan fungsi aplikasi *visual* lainnya.

Delphi 7.0 dapat menangani pembuatan aplikasi sederhana hingga aplikasi berbasis jaringan. *Delphi* 7.0 dapat dimanfaatkan untuk membuat aplikasi berbasis teks, *grafis*, angka, *database* maupun web.

Bahasa pemrograman visual memiliki dua hal yaitu *object* dan kode program. Objek berbentuk komponen yang dapat dilihat (*visual*), sedangkan kode program merupakan sekumpulan teks yang digunakan sebagai perintah yang telah diatur dengan suatu aturan.

Keunggulan yang dimiliki oleh *Borland Delphi* yaitu :

a. Memiliki banyak fitur berupa *component* yang dapat ditambahkan.
b. Dapat merancang dan membuat tampilan perangkat lunak yang bagus
c. Mudah dalam penulisan *coding*
d. Kompatibel dengan berbagai macam jenis database