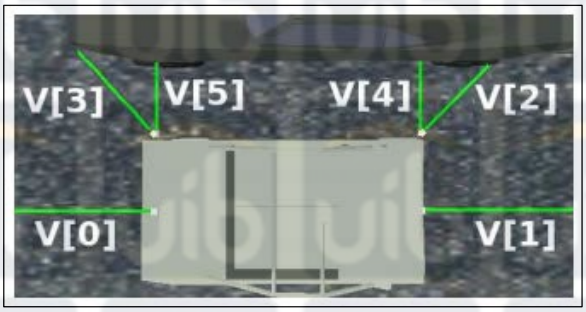
BAB II
KAJIAN PUSTAKA DAN LANDASAN TEORI

2.1. Penelitian Sebelumnya

Penelitian sebelumnya terkait dengan *self-parking* pernah diteliti oleh Milton Roberto Heinen, dkk. Judul yang diambil dalam penelitian tersebut adalah *"Autonomous Vehicle Parking and Pull Out Using Artificial Neural Network"*. Penelitian yang dilakukan diimplementasikan pada simulasi komputer SEVA3D-A (*control base on a FSA*) dan SEVA3D-N (*control base on an ANN*) [4]. Pada penelitian ini design mekanik yang digunakan adalah *non-holonomic kinematic* dengan roda penggerak pada roda bagian belakang dan roda *steering* pada bagian depan. Sebagai batasan masalah yang digunakan dalam penelitian ini adalah peletakan sensor sonar yang digunakan hanya dibatasi pada sisi kanan bagian mobil. Hal ini dikarenakan sisi tersebut sudah dapat mewakili kondisi untuk dua ruas jalan yang berlawanan.



Gambar 2.1. Peletakan Sensor, sumber : Milton, Roberto Heinen

Metode yang digunakan adalah ANN dengan metode pembelajaran *supervised learning*. ANN dapat belajar untuk dapat memarkirkan mobil dan mengeluarkan mobil dengan sendirinya. Langkah awal yang harus dilakukan

Muhammad Thoriqul Ullum, Implementasi Metode ANN (Artificial Neural Network) Pada Robot Self-Parking (Nonholonomic Robot) Menggunakan Prototype Robot Lego® Mindstorm®, 2015, UIB Repository©2015

adalah menyiapkan data parkir yang akan digunakan sebagai data pembelajaran pada ANN. Data tersebut berupa data kondisi sensor dari masing-masing titik, data aktuator dari setiap pergerakan *steering* yang nantinya akan digunakan pada komputasi ANN. Jaringan ANN ini digunakan dengan cara berikut ini : data *input* digunakan untuk menunjukkan kondisi saat ini dan data *output* digunakan sebagai penentu langkah selanjutnya.

Database yang digunakan adalah sebanyak 5000 contoh dengan 2500 contoh untuk pembelajaran dan 2500 contoh untuk penyamarataan, masing-masing dengan 13 *input* dan 12 *output*. Hasil dari pembelajaran JST menghitung rata-rata 10 simulasi dengan inisialisasi yang berbeda memiliki kinerja belajar dengan nilai rata-rata 96,83% dengan hasil jawaban benar[4]. Jawaban yang benar adalah ketika semua sinyal *output* dikoreksi, mempertimbangkan *thereshold* 0,4. Tiga buah neuron yang digunakan dalam rata-rata *epoch* adalah 192 kali.

Tujuan utama dari penelitian ini adalah untuk mengembangkan simulator untuk kontrol otonom kendaraan *autonomous parallel parking*. Sistem ini meliputi model 3D dari sensor sonar, hasil penelitian menyatakan bahwa sistem kontrol dapat mengontrol kendaraan. SEVA3D jaringan saraf akan disesuikan (dilatih) dengan menggunakan perangkat keras (sensor sonar yang asli) dalam artian sistem harus dapat diimplementasikan pada perangkat yang sebenarnya.

Kemudian penelitian yang berbeda mengenai *Sequential Learning* dengan judul *"A Fast and Accurate Online Sequential Learning Algorithm for Feedforward Networks"* yang dilakukan oleh Nan-Ying Liang, dkk [17]. Dalam jurnal tersebut, dijelaskan mengenai *Sequential Learning Algorithm Feedforward*

Muhammad Thoriqul Ullum, Implementasi Metode ANN (Artificial Neural Network) Pada Robot Self-Parking (Nonholonomic Robot) Menggunakan Prototype Robot Lego® Mindstorm®, 2015, UIB Repository©2015

Network (SLFNs) untuk jaringan tunggal tersembunyi lapisan *feedforward* dengan tambahan node tersembunyi dalam kerangka terpadu. Algoritma ini memungkinkan untuk dapat mempelajari data satu – per – satu atau potong per potong dengan nilai yang bervariasi. Fungsi aktivasi untuk node aditif berupa bagian demi bagian *nonconstant* fungsi kontinu yang dibatasi.

Untuk beberapa kebutuhan kita dapat menyederhanakannya menjadi beberapa kategori dasar, yaitu sebagai berikut ini [18].

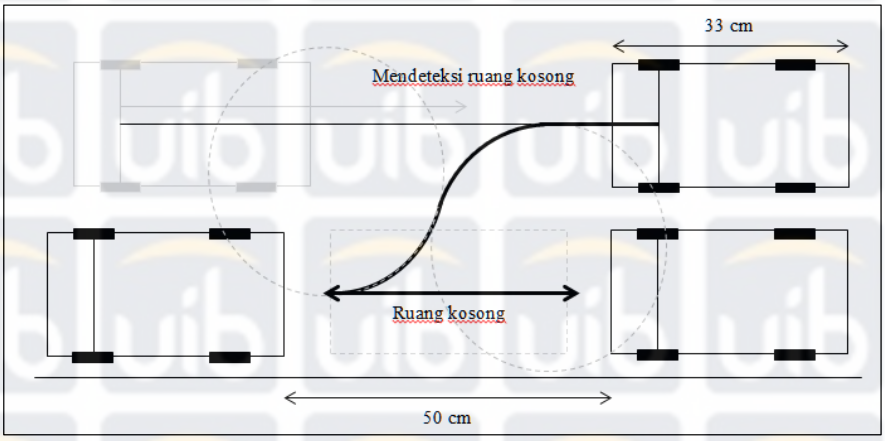
- *Sequence prediction* yaitu mencoba untuk memprediksi unsur urutan atas dasar unsur-unsur sebelumnya.
- *Sequence generation* adalah upaya untuk menghasilkan unsur urutan satu per satu secara alami.
- *Sequence recognition* adalah sebuah cara yang digunakan untuk menentukan apakah urutan sah menurut beberapa kriteria.
- *Sequence decision making* merupakan pengambil keputusan yang berurutan yaitu pemilihan urutan tindakan untuk mencapai tujuan, contoh untuk mengikuti lintasan (*trajectory*).

Pentingnya perilaku sekuensial dan urutan pembelajaran dalam kecerdasan buatan dan pengartian tidak bisa berdasarkan taksiran. Sebelumnya *artificial intelligence* dan mesin pembelajaran penelitian secara keliru mengabaikkan peranan perilaku sekuensial dan pembelajaran berurutan. Ini adalah langkah pertama menuju pembentukan kerangka kerja konseptual yang lebih kesatuan dan koheren untuk mempelajari berbagai jenis *sequential learning*.

Muhammad Thoriqul Ullum, Implementasi Metode ANN (Artificial Neural Network) Pada Robot Self-Parking (Nonholonomic Robot) Menggunakan Prototype Robot Lego® Mindstorm®, 2015, UIB Repository©2015

2.2. *Autonomous Self-Parking*

Robot *Self-parking* adalah sebuah robot yang memungkinkan dirinya untuk dapat mencari hingga menempatkan diri pada area parkir yang kosong. Kinematik robot yang digunakan sama seperti halnya sebuah mobil pada skala sebenarnya yaitu menggunakan *non-holonomic kinematic.* Kinematik ini mempunyai karakteristik yang memiliki empat roda dengan dua roda di bagian belakang sebagai penggerak dan dua roda pada bagian depan sebagai *steering.*



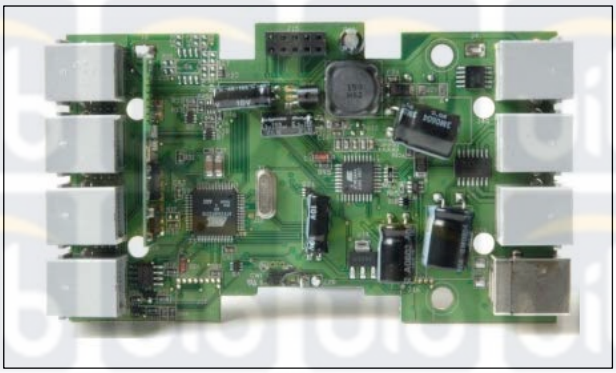
*Gambar 2.2.*Skema prosedur parkir

Sistem parkir otomatis bertujuan untuk meningkatkan kenyaman dan keamanan pada saat melakukan *maneuver* parkir, terlebih jika yang melakukan parkir tersebut adalah pemula dengan kondisi lingkungan parkir yang sulit. *Maneuver* parkir ini bisa dilakukan dengan koordinasi kontrol antara sudut kemudi (*steering angle*) dan kecepatan yang memperhitungkan situasi aktual di lingkungan tersebut, untuk dapat memastikan gerak bebas tabrakan dalam ruang yang tersedia [13].

Muhammad Thoriqul Ullum, Implementasi Metode ANN (Artificial Neural Network) Pada Robot Self-Parking (Nonholonomic Robot) Menggunakan Prototype Robot Lego® Mindstorm®, 2015, UIB Repository©2015

2.3. NXT 2.0 Lego® Mindstorm®

NXT 2.0 adalah *platform* buatan Lego Mindstorms dengan spesifikasi yang berjalan pada system 32-bit dibekali dengan mikroprosesor ARM 7 dengan frekuensi 48 MHz selain itu *platform* ini memiliki kapasitas memori 256 Kbytes FLASH, 64 Kbyte RAM. NXT 2.0 ini memiliki 4 port *input* untuk sensor dan memiliki 3 *port output* untuk motor servo (*actuator*). Untuk *interface* NXT ini bisa menggunakan kabel USB atau juga bisa dengan menggunakan *Bluetooth.* Sebagai sumber *sources power* NXT 2.0 menggunakan batrei sebanyak 6 buah dengan tegangan masing-masing batrei 1.5 volt atau dengan menggunakan *power source rechargeable lithium battery* [5].



Gambar 2.3. NXT 2.0 Lego Mindstorms®, sumber :http://education.lego.com

Sedangkan untuk bahasa pemrogramman secara standar NXT 2.0 menggunakan bahasa yang sering di sebut bahasa *drag and drop programming* karena memang *platform* ini memang diciptakan untuk anak-anak sehingga mempermudah dalam pemrogramannya. Selain bahasa *drag and drop platform* ini juga dapat di program dengan menggunakan bahasa C dengan menggunakan *software* ROBOTC.

Muhammad Thoriqul Ullum, Implementasi Metode ANN (Artificial Neural Network) Pada Robot Self-Parking (Nonholonomic Robot) Menggunakan Prototype Robot Lego® Mindstorm®, 2015, UIB Repository©2015

2.4. Ultrasonic Sensor

Ultrasonic adalah suara atau getaran dengan frekuensi yang terlalu tinggi untuk bisa didengar oleh telinga manusia. Sensor ultrasonic secara sederhana adalah sebagai penangkap gelombang ultrasonic yang mempunyai rentang lebih besar dari 20 KHz. Gelombang *ultrasonic* ini digunakan oleh lumba-lumba dan kelelawar.

Pada LEGO® MINDSTORM® 2.0, sensor ini merupakan sensor utama yang dijadikan sebagai navigasi untuk menghindari halangan. Bisa dikatakan pemanfaatan gelombang ultrasonic ini sama dengan teknik navigasi pada kelelawar maupun kapal selam. Sensor ini memiliki bentuk seperti mata yang terdiri dari dua buah bagian yaitu bagian sebelah kanan adalah *transmitter* dan yang sebelah kiri adalah *receiver*.

Gambar 2.4. Ultrasonic Sensor, sumber :http://education.lego.com

Cara kerja sensor tersebut sama seperti konsep penggunaan gelombang ultrasonic untuk menentukan jarak. Awalnya mata pada bagian kanan akan memancarkan gelombang ultrasonic kemudian mata bagian kirinya akan menerima pantulan dari gelombang tersebut. Selisih antara waktu gelombang

Muhammad Thoriqul Ullum, Implementasi Metode ANN (Artificial Neural Network) Pada Robot Self-Parking (Nonholonomic Robot) Menggunakan Prototype Robot Lego® Mindstorm®, 2015, UIB Repository©2015

yang dikirimkan dan pantulannya diterima akan digunakan untuk menentukan posisi benda yang dideteksi.

Berdasarkan penelitian ada dua kelemahan utama pada sensor ini.Kelemahan pertama adalah terjadinya kekacauan untuk perhitungan pada jarak lebih dari 255cm. Hal ini disebabkan karena nilai yang didaptkan sensor sudah berada diluar rentang nilai yang dimiliki oleh sensor tersebut. Kelemahan kedua adalah untuk benda yang berada pada jarak antara 25 – 50 cm, sensor mempunyai probabilitas besar untuk membaca jarak tersebut menjadi 48cm. [6]

2.5. Motor Servo

Fungsi servo motor adalah sebagai motor untuk menggerakan komponen lain dalam Lego mindstorm. Kecepatan sudut maksimum motor adalah 2π / detik atau sama dengan satu putaran/detik. Motor ini dilengkapi dengan kemampuan untuk dapat menghitung perubahan sudut yang dialami oleh motornya. Servo motor dapat menghitung berapa derajat rotasi yang telah dilakukannya. Akurasi dari motor ini kurang lebih 1derajat.

Gambar 2.5. Motor DC, Sumber :http://www.amazon.ca

Muhammad Thoriqul Ullum, Implementasi Metode ANN (Artificial Neural Network) Pada Robot Self-Parking (Nonholonomic Robot) Menggunakan Prototype Robot Lego® Mindstorm®, 2015, UIB Repository©2015

2.6. Karakteristik Geometri Mobil

Variasi sudut belok yang dapat dilakukan mobil adalah $-\beta_{min} < \beta < \beta_{max}$. saat kecepatan mobil konstan dan *steering angle* (sudut belok) fiks, maka pergerakan mobil akan membentuk lintasan dengan radius yang bervariasi sesuai dengan variasi sudut belok yang diberikan. Kondisi inilah yang digunakan untuk merencanakan lintasan yang sederhana dengan gerakan melingkar.

Radius putar merupakan radius sebuah lingkaran yang dibentuk oleh mobil saat memutar dengan sudut belok tetap.Dengan menggunakan sudut belok β dan *wheelbase* e, atas dasar aturan *trigonometry*dapat ditulis sebagai berikut.

$$R = \frac{e}{sin\beta} \quad \text{..... (1)}$$

Dua radius putar lainnya dapat didefinisikan sebagai R_i merupakan radius terkecil yang dibentuk oleh roda belakang bagian dalam. Sedangkan radius terluar R_e merupakan radius terbesar yang dibentuk oleh sudut depan bagian luar dari *prototype* mobil itu sendiri. Semakin besar sudut belok yang terbentuk maka semakin kecil radius lingkaran yang terbentuk.

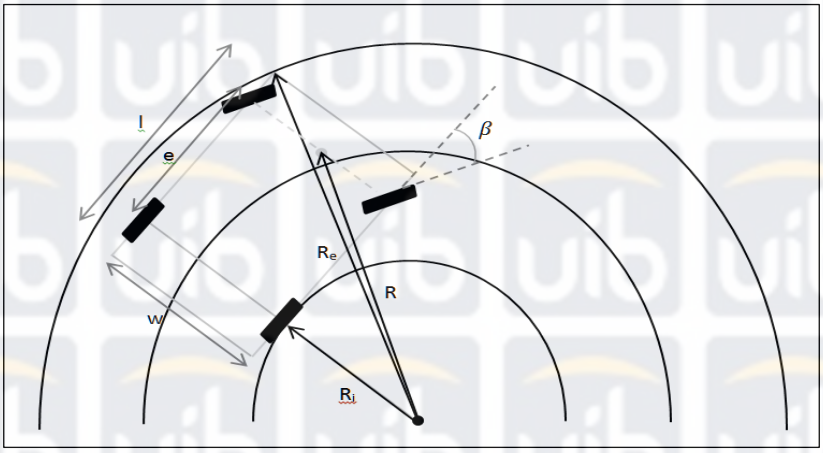
Radius $R_i$dapat dihitung dari R dengan teorema *phytagoras* yang ditujukan pada persamaan (2.4) begitu juga dengan R_e terdapat pada persamaan (2.5). Ketiga lingkaran yang terbentuk ini digunakan sebagai acuan perhitungan *parking space* minimal yang digunakan[8].

$$R_i = \sqrt{R^2 - e^2} - \frac{w}{2} = \sqrt{\frac{e^2}{sin^2\beta} - e^2} - \frac{w}{2}$$

$$R_i = \frac{e}{tan\beta} - \frac{w}{2} \quad \text{..... (2)}$$

$$R_e = \sqrt{(R_i - w)^2 + (e + p)^2}$$

$$R_e = \sqrt{(\sqrt{R^2 - e^2} + \frac{w}{2})^2 + (e + p)^2} \quad \text{..... (3)}$$



Gambar 2.6. Radius Belok Robot, sumber : Habiburrahman, 2012

Dimana :

l = panjang dari *prototype* mobil (cm)

e = *wheelbase* (jarak antara poros roda belakang dan poros roda depan)

p = jarak antara poros depan ke bagian paling belakang dari *prototype* mobil atau poros roda belakang ke bagian paling belakang dari *prototype*mobil tersebut. (cm)

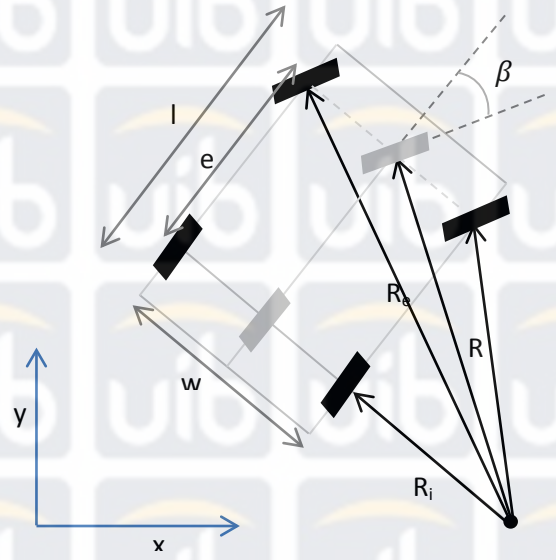
W = lebar mobil (cm)

β = sudut *steering* (^{o})

2.7. *Non-Holonomic Kinematic*

Model *autonomous car* yang digunakan adalah model kinematic *non-holonomic* (*ackerman model*) dengan penggerak utama berada di roda belakang

yang hanya bisa bergerak maju mundur. Sedangkan untuk kemudinya berada pada roda depan yang bisa bergerak ke kanan dan kiri. Model matematika didapatkan dengan cara menggambarkannya pada bidang horizontal dengan asumsi pengendali sistem kinematic selalu benar maka semua roda akan berputar di sekitar titik yang sama dilambangkan dengan notasi P yang terletak pada garis poros roda belakang.



Gambar 2.7. Notifikasi Variable Robot, sumber :Habiburrahman, 2012

Kinematika robot dapat direpresentasikan dengan kinematika sepeda dengan memberikan roda virtual di tengah-tengah poros roda robot. Koordinat titik tengah garis poros roda belakang dilambangkan dengan *x,y*. Orientasi robot dilihat dari sumbu x dilambangkan dengan Θ. Sudut roda depan sepeda yang berhubungan dengan sumbu simetri longitudinal robot dlambangkan dengan β, sehingga dapat di ambil β atau turunannya $u_1 = d\beta/dt$ sebagai masukan. Kecepatan longitudinal dari titik tengah garis poros roda belakang dinotasikan sebagai u_2 [14].

Muhammad Thoriqul Ullum, Implementasi Metode ANN (Artificial Neural Network) Pada Robot Self-Parking (Nonholonomic Robot) Menggunakan Prototype Robot Lego® Mindstorm®, 2015, UIB Repository©2015

Dalam perancangan kontroler dan perancangan trayektori, robot direpresentasikan dalam sebuah persamaan dinamika dan kinematika. Dalam kondisi ideal dimana robot bergerak dengan kecepatan rendah, gaya-gaya lawan yang mempengaruhi pergerakan robot seperti kecepatan angin, konstanta slip roda dengan permukaan jalan, serta pengaruh suspense dianggap tidak ada. Sehingga persamaan dinamika robot diperoleh dengan pendekatan geometri dan dapat ditulis sebagai berikut ini.

$$\frac{d}{dt}\begin{bmatrix} x \\ y \\ \theta \\ \beta \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_1 + \begin{bmatrix} cos\theta \\ sin\theta \\ \frac{tan\beta}{e} \\ 0 \end{bmatrix} u_2 \quad \text{..... (4)}$$

Dengan asumsi keadaan ideal makan persamaan kinematika robot diperoleh dari persamaan dinamikanya dengan mengubah parameter kecepatan untuk setiap sudut pandang menjadi posisi, sehingga posisi robot pada setiap saat (waktu) dapat dihitung sebagai berikut:

$$x = \int_{t_o}^{t} cos\theta . u_2 dt \quad \text{..... (5)}$$

$$y = \int_{t_o}^{t} sin\theta . u_2 dt \quad \text{..... (6)}$$

$$\theta = \int_{t_o}^{t} \frac{tan\beta}{2.7} . u_2 dt \quad \text{..... (7)}$$

$$\beta = \int_{t_o}^{t} u_1 dt \quad \text{..... (8)}$$

Pada penelitian sebelumnya yang dilakukan oleh Habiburrahman (2012). Model *autonomous car* yang digunakan adalah model *ackerman* (sama dengan *nonholonomic kinematic*) dengan penggerak utama berada pada roda belakang

Muhammad Thoriqul Ullum, Implementasi Metode ANN (Artificial Neural Network) Pada Robot Self-Parking (Nonholonomic Robot) Menggunakan Prototype Robot Lego® Mindstorm®, 2015, UIB Repository©2015

yang hanya bisa bergerak maju dan mundur sedangkan kemudi diletakkan pada roda depan[8].

Pada percobaan yang telah dilakukan oleh Habiburahman didapatkan kesimpulan bahwa perencanaan sistem kendali yang saling independen antara sudut belok dan kecepatan mobil memiliki kelemahan dalam penerjemahan *error* yang terjadi pada lintasan[8].

2.8. *Artificial Neural Network*

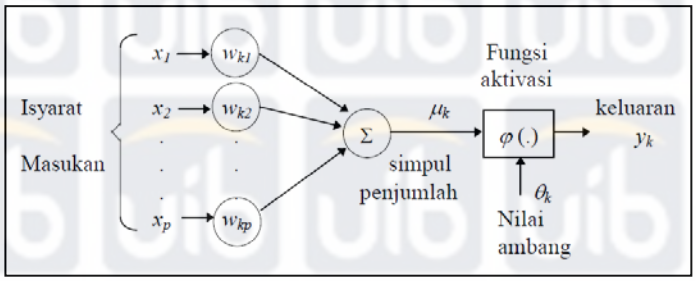
Artificial Neural Network atau Jaringan Syaraf Tiruan (JST) dibuat pertama kali oleh Mc Culloch dan Pitts pada tahun1943. Mc Culloch dan Pitts menyimpulkan bahwa kombinasi beberapa neuron sederhana menjadi sebuahsistem *neural* akan meningkatkan kemampuan komputasinya. Bobot dalam jaringan yang diusulkan oleh Mc Culloch dan Pitts diatur untuk melakukan fungsi logika sederhana. Fungsi aktivasi yang dipakai adalah fungsi *threshold*. Selanjutnya pada tahun 1958, Rosenblatt memperkenalkan dan mulai mengembangkan model jaringan baru yang terdiri dari beberapa lapisan yangdisebut *perceptron*.

Metode pelatihan diperkenalkan untuk mengoptimalkan hasil iterasi-nya. Widrow dan Hoff pada tahun 1960 mengembangkan *perceptron* dengan memperkenalkan aturan pelatihan jaringan, yang dikenal sebagai aturan delta (kuadrat rata-rata terkecil). Aturan ini akan mengubah bobot *perceptron* apabila keluaran yang dihasilkan tidak sesuai dengan target yang diinginkan. Apa yang dilakukan peneliti terdahulu hanya menggunakan jaringan dengan layer tunggal

Muhammad Thoriqul Ullum, Implementasi Metode ANN (Artificial Neural Network) Pada Robot Self-Parking (Nonholonomic Robot) Menggunakan Prototype Robot Lego® Mindstorm®, 2015, UIB Repository©2015

(*single layer*). Rumelhart bersama Mc Clelland pada tahun 1986 mengembangkan *perceptron* menjadi *backpropagation*, yang memungkinkan jaringan diproses melalui beberapa layer [9]. Secara umum JST terbentuk dari jutaan struktur dasar neutron yang terinterkoneksi dan terintegrasi antara satu dengan yang lain sehingga dapat melaksanakan aktifitas secara teratur dan terus menerus sesuai dengan kebutuhan.

Dalam merancang suatu jaringan syaraf tiruan, selain memperhatikan struktur hubunganantara masukan dengan keluaran, perlu ditentukan juga cara atau metode pembelajarannya. Belajar bagi jaringan syaraf adalah cara memperbaharui bobot sinapsis yang disesuaikan dengan isyarat masukan dan keluran yang diharapkan. Secara umum suatu jaringan syaraf dibentuk atas sejumlah *neuron* sebagai unit pengolah informasi sebagai dasar operasi untuk menjalankan fungsi atau tugasnya. Model susunan *neuron* dapat ditunjukkan pada gambar berikut:



Gambar 2.8. Model *neuron k*, sumber :Saludin Muis, 2006

Secara matematis, untuk *neuron k* berlaku persamaan:

$$u_k = \sum_{j=1}^{p} w_{kj}\, x_j \quad \text{..... (9)}$$

$y_k = \varphi(\mu_k - \theta_k)$ ..... (10)

Dengan $x_1, x_2, \dots, x_p$ adalah isyarat masukan, $w_{k1}, w_{k2}, \dots, w_{kp}$ adalah bobot sinapsis untuk *neuronk*, μ_k adalah keluaran kombinasi linier, θ_k adalah nilai ambang, $\varphi(\,)$ adalah fungsi aktivasinya, dan $y_k$adalah isyarat keluaran *neuronk* [9]. Dari model *neuron* untuk jaringan syaraf seperti terlihat pada Gambar 1 terdapat tiga bagian utama berikut:

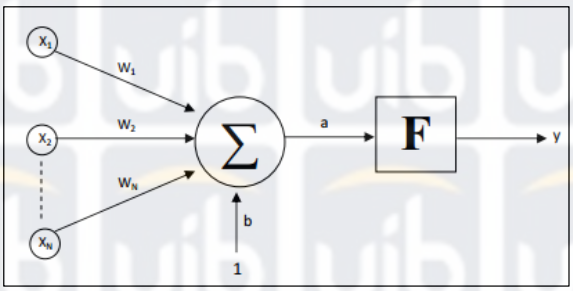
1. Sekumpulan sinapsis atau jalur penghubung, yang mempunyai bobot atau kekuatan. Indeks yang tertulis pada notasi bobot sinapsis menunjukkan posisi masukan dan keluaran yang dihubungkan.
2. Unit penjumlah untuk menjumlahkan isyarat masukan.
3. Fungsi aktivasi untuk membatasi amplitude keluaran.

2.9. Fungsi Aktivasi

Fungsi aktivasi merupakan fungsi yang biasa digunakan pada sebuah jaringan syaraf tiruan untuk mengaktifkan atau tidak mengaktifkan neuron.seperti pada gambar 2.7 sebuah neuron akan mengolah N *input* (x1, x2, …x_n) yang masing-masing memiliki bobot w1, w2, w3, … w_n dan bobot bias b, dengan rumus sebagai berikut:

$a = \sum_{i=1}^{N} x_i w_i$ .....(11)

Kemudian fungsi aktivasi F akan mengativasi a menjadi *output* jaringan y.



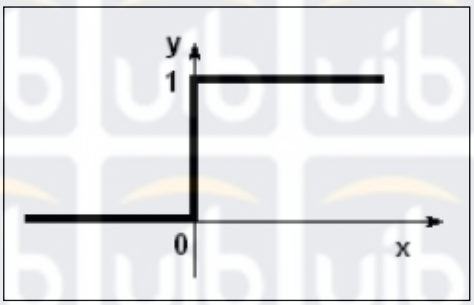
Gambar 2.9. Fungsi aktivasi pada JST sederhana, sumber : Kusumadewi, Sri 2003

Ada beberapa fungsi aktivasi yang sering digunakan dalam jaringan syaraf tiruan.Berikut adalah fungsi aktivasi yang biasa digunakan dalam ANN.

1. Fungsi Undak Biner (*Hardlim*)

   Fungsi undak biner (step function) yang sering digunakan pada jaringan dengan lapisan tunggal untuk mengkonversi *input* dari suatu variable yang bernilai kontinu ke suatu *output* biner (0 atau 1) dengan syntax Y = hardlim(a).

$$Y = \begin{cases} 0, jika\ x \leq 0 \\ 1, jika\ x > 0 \end{cases} \quad \text{(12)}$$



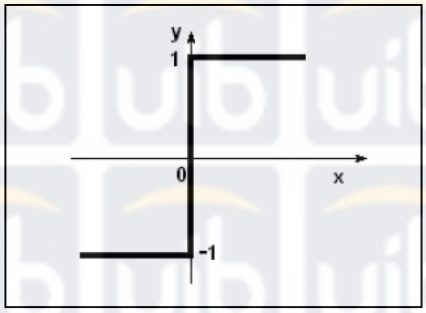
Gambar 2.10. Fungsi Aktivasi *Biner Hard Limit*, sumber : Kusumadewi, 2003

Muhammad Thoriqul Ullum, Implementasi Metode ANN (Artificial Neural Network) Pada Robot Self-Parking (Nonholonomic Robot) Menggunakan Prototype Robot Lego® Mindstorm®, 2015, UIB Repository©2015

2. Fungsi Bipolar (*Hardlims*)

Fungsi bipolar sebenarnya hampir sama dengan fungsi undak biner, hanya saja *output* yang dihasilkan berupa 1 atau -1, dengan syntax : Y=hardlim(a)

$$Y = \begin{cases} 1, jika\ x > 0 \\ 0, jika\ x = 0 \\ -1, jika\ x < 0 \end{cases} \quad \text{..... (13)}$$

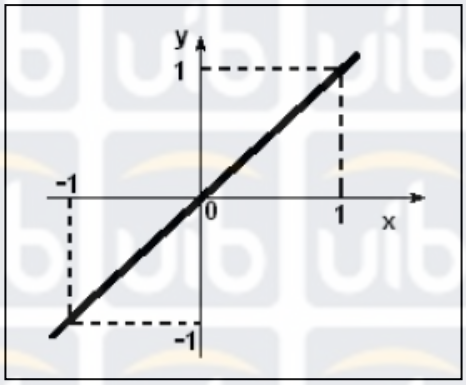


Gambar 2.11. Fungsi Aktivasi *Bipolar Sigmoid*, sumber : Kusumadewi, 2003

3. Fungsi linier (*purelin*)

Fungsi linier memiliki nilai *output* sama dengan nilai *input*nya. Fungsi linier tersebut dirumuskan sebagai berikut :

$$y = x \quad \text{..... (14)}$$

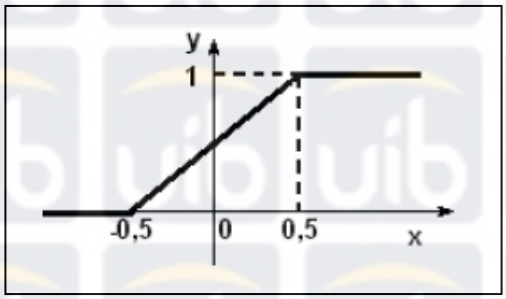


Gambar 2.12. Fungsi Aktivasi *Linear*, sumber : Kusumadewi, 2003

Muhammad Thoriqul Ullum, Implementasi Metode ANN (Artificial Neural Network) Pada Robot Self-Parking (Nonholonomic Robot) Menggunakan Prototype Robot Lego® Mindstorm®, 2015, UIB Repository©2015

4. Fungsi *Saturating Linier*

Fungsi ini akan bernilai 0 jika *input*nya kurang dari ½ dan akan bernilai 1 jika *input*nya lebih dari ½. Sedangkan jika nilai *input* terletak antara -½ dan ½, maka *output*nya akan bernilai sama dengan nilai *input* ditambah ½.



Gambar 2.13. Fungsi Aktivasi *Saturating Linear*, sumber : Kusumadewi, 2003

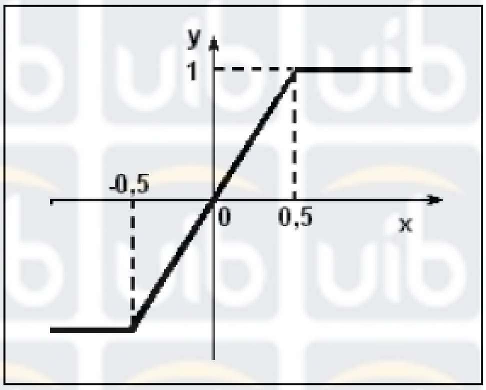
Fungsi *Saturating Linear*ini dapat dirumuskan sebagai berikut.

$$Y = \begin{cases} 1; jika\ x \geq 0 \\ x + 0.5; jika - 0.5 \leq x \leq 0.5 \\ 0; jika\ x \leq 0 \end{cases} \quad \text{(15)}$$

5. Fungsi *Symetric Saturation Linier*

Fungsi ini akan bernilai -1 jika *input*nya kurang dari -1, dan akan bernilai 1 jika *input*nya lebih dari 1. Sedangkan jika nilai *input*nya terletak antara -1 dan 1, maka *output*nya akan bernilai sama dengan nilai *input*nya. Berikut adalah persamaan dari fungsi tersebut.

$$Y = \begin{cases} 1; jika\ x \geq 1 \\ x; jika - 1 \leq x \leq 1 \\ -1; jika\ x \leq -1 \end{cases} \quad \text{(16)}$$

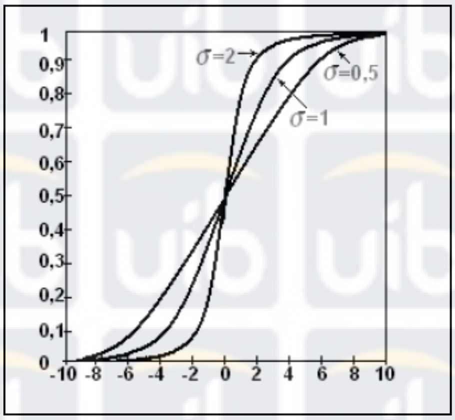


Gambar 2.14. Fungsi *Symetric Saturating Linear*, sumber : Kusumadewi, 2003

6. Fungsi *Sigmoid Biner*

Fungsi ini digunakan untuk JST yang dilatih dengan menggunakan metode *backpropagation.* Fungsi sigmoid biner memiliki nilai pada range 0 sampai 1. Oleh karena itu, fungsi ini sering digunakan untuk jaringan syaraf yang membutuhkan nilai *output* yang terletak pada interval 0 sampai 1, namun fungsi ini juga dapat digunakan oleh JST yang nilai *output*nya 0 atau 1. Berikut ini adalah rumus yang digunakan untuk fungsi sigmoid biner.

$y = f(x) = \frac{1}{1+e^{-\propto x}}$ .... (17)

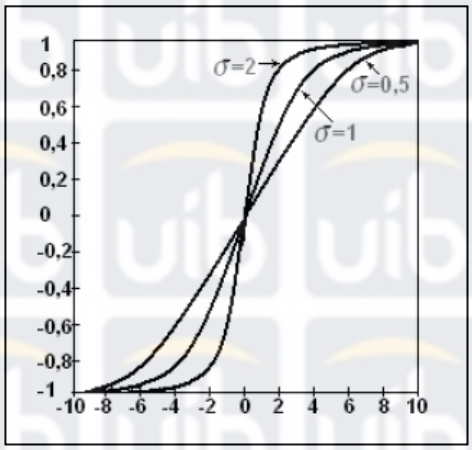


Gambar 2.15. Fungsi Aktivasi *Sigmoid Biner*, sumber : Kusumadewi, 2003

Muhammad Thoriqul Ullum, Implementasi Metode ANN (Artificial Neural Network) Pada Robot Self-Parking (Nonholonomic Robot) Menggunakan Prototype Robot Lego® Mindstorm®, 2015, UIB Repository©2015

7. Fungsi *Sigmoid Bipolar*

Fungsi *sigmoid bipolar* hampir sama dengan fungsi *sigmoid biner*, hanya saja *output* dari fungsi ini memiliki *range* antara 1 sampai -1.



Gambar 2.16. Fungsi Aktivasi *Sigmoid Bipolar*, sumber : Kusumadewi, 2003

Fungsi *sigmoid bipolar* dirumuskan sebagai berikut:

$$y = \frac{1-e^{-x}}{1+e^{-x}} \quad \text{..... (18)}$$

2.10. Metode Pelatihan *Neural Network*

2.10.1. Metode Pelatihan Terbimbing

Metode pelatihan terbimbing (*supervised learning*) adalah metode pelatihan yang memasukkan target keluaran dalam data untuk proses pelatihannya. Ada beberapa metode pelatihan terbimbing yang telah ditentukan oleh para peneliti, diantaranya yang sering diaplikasikan adalah *perceptron* dan *backpropagation*. Metode *backpropagation* tersebut sampai saat ini masih sangat banyak yang menggunakannya, begitu juga dengan yang telah dimodifikasi, sehingga kinerjanya dapat menjadi lebih efektif.

Muhammad Thoriqul Ullum, Implementasi Metode ANN (Artificial Neural Network) Pada Robot Self-Parking (Nonholonomic Robot) Menggunakan Prototype Robot Lego® Mindstorm®, 2015, UIB Repository©2015

2.10.2. Metode Pelatihan Tidak Terbimbing

Metode pelatihan tidak terbimbing (*unsupervised learning*) ini tidak memerlukan target *output*. Tujuan metode ini adalah pengelompokkan unit-unit yang hampir sama dalam suatu area tertentu.

2.11. *Backpropagation Neural Network*

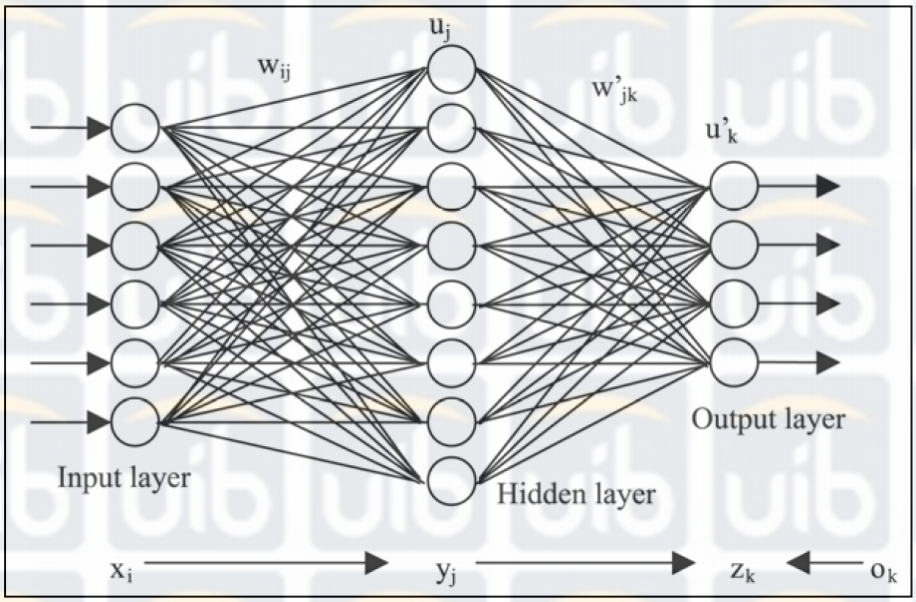
Metode pelatihan *Backpropagation Neural Network* (BP-NN) pertama kali dirumuskan oleh Werbos dan dipopulerkan oleh Rumelhart & Mc Clelland. *Backpropagation Neural Network* merupakan tipe jaringan saraf tiruan yang menggunakan metode pembelajaran terbimbing (*supervised learning*). Pada *supervised learning* terdapat pasangan data *input* dan *output* yang dipakai untuk melatih ANN hingga diperoleh bobot penimbang (*weight*) yang diinginkan. Penimbang itu sendiri adalah sambungan antar lapis dalam ANN. Algoritma ini memiliki proses pelatihan yang didasarkan pada interkoneksi yang sederhana, yaitu apabila keluaran memberikan hasil yang salah, maka penimbang dikoreksi agar *error* dapat diperkecil dan tanggapan ANN selanjutnya diharapkan dapat mendekati nilai yang benar. BP-NN juga berkemampuan untuk memperbaiki penimbang pada lapis tersembunyi (*hidden layer*) [12].

Secara garis besar BP-NN terdiri atas tiga lapis (*layer*), yaitu lapis masukan (*input layer*) x_i, lapis tersembunyi (*hidden layer*) y_j, dan lapis keluaran (*output layer*) z_k. Lapis masukan dan lapis tersembunyi dihubungkan dengan penimbang w_{ij} dan antara lapis tersembunyi dan lapis keluaran dihubungkan oleh penimbang w'_{jk}. Pada pelatihan BP-NN, ketika ANN diberi pola masukan sebagai

Muhammad Thoriqul Ullum, Implementasi Metode ANN (Artificial Neural Network) Pada Robot Self-Parking (Nonholonomic Robot) Menggunakan Prototype Robot Lego® Mindstorm®, 2015, UIB Repository©2015

pola pelatihan maka pola tersebut akan menuju ke unit pada lapis tersembunyi untuk diterusan pada unit yang berada pada lapis keluaran. Keluaran sementara pada lapis tersembunyi u_j akan diteruskan pada lapis keluaran dan lapis keluaran akan memberikan tanggapan yang disebut sebagai keluaran sementara u'_k.

Ketika $u'_k \neq o_k$ dimana o_k adalah keluaran yang diharapkan, maka selisih (*error*) keluaran sementara u'_k akan disebarkan mundur (*backward*) pada lapis tersembunyi dan diteruskan ke unit pada lapis masukan. Oleh karena itu, proses tersebut dinamakan propagasi balik (*backpropagation*) dimana tahap pelatihan dilakukan dengan merubah penimbang yang menghubungkan unit dalam lapis ANN ketika diberi umpan maju dan umpan balik [12].



*Gambar 2.17.Backpropagation*MLP, sumber : Muis, Saludin (2006)

Penemuan NN dengan algoritma *backpropagation* menggunakan *multi layer perceptron* (MLP) menjadi salah satu alternatif penyelesaian permasalahan

yang susah untuk diturunkan dalam model matematiknya [12]. BP-NN ini hanya memiliki koneksi (*synapse*) antara *input, hidden layer* dan *output*.

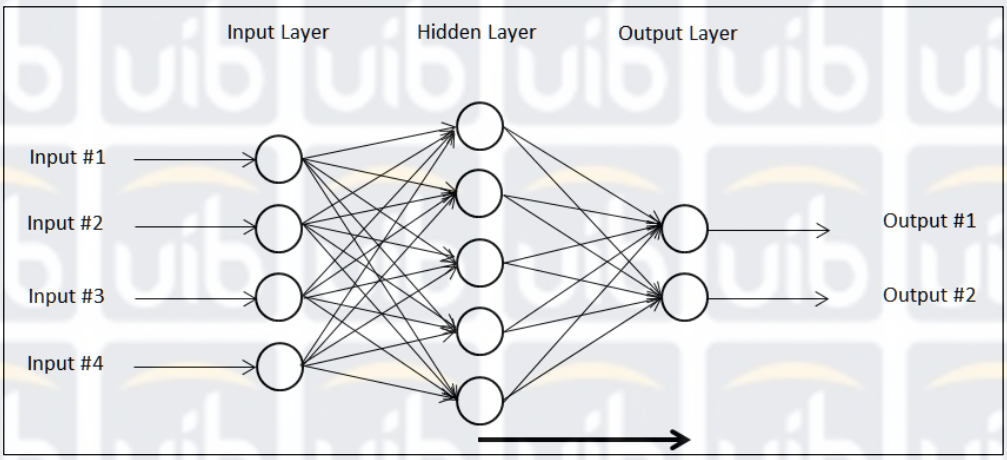
Sedangkan untuk mendeteksi kesamaan pola *output* pembelajaran dan pola *output* yang dihasilkan, maka *error* yang dihasilkan kedua *output* tersebut akan dipropagasikan balik selama proses pelatihan. Agar mendapatkan keseimbangan antara kemampuan untuk mengenali *input* dan pola *output*, maka jaringan akan dilatih. Pelatihan tersebut akan berjalan sampai pola *output* tersebut serupa dengan pola yang digunakan selama pelatihan [12].

Algoritma pelatihan *Backpropagation Neuron Network*merupakan metode pembelajaran terawasi (*supervise*) dimana untuk melakukan pelatihan pada metode tersebut harus melalui 3 fase, yang terdiri dari *feed forwardpropagation*, *feed backwardpropagation* dan *adjust weight*.

2.11.1. *Feed forwardpropagation*

Propogasi maju memiliki *neuron* yang tersusun dari beberapa layer, layer *input* pada awalnya hanya akan memberikan suatu nilai dari suatu variable. Kemudian *hidden layer* dan *output layer* pada *neuron* memiliki hubungan yang saling terintegrasi.

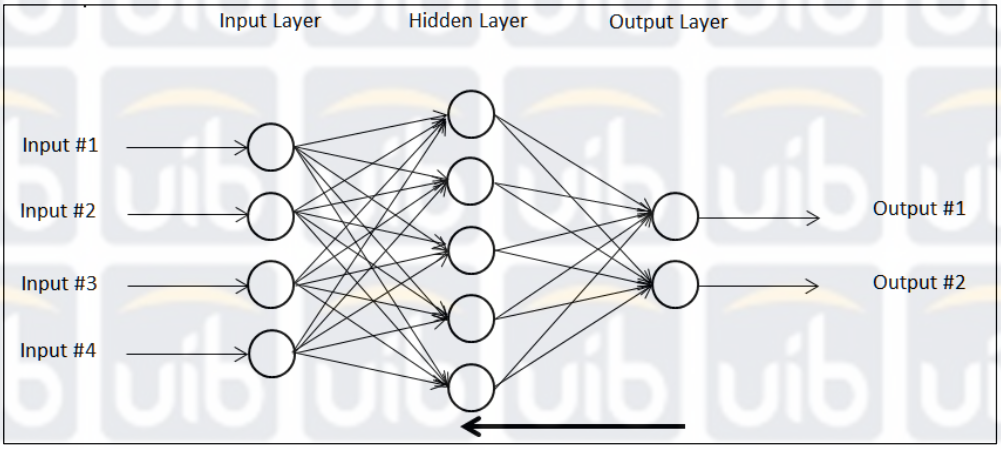
Muhammad Thoriqul Ullum, Implementasi Metode ANN (Artificial Neural Network) Pada Robot Self-Parking (Nonholonomic Robot) Menggunakan Prototype Robot Lego® Mindstorm®, 2015, UIB Repository©2015



Gambar 2.18. Struktur *feedforward*

2.11.2. *Feed backwardpropagation*

Proses *backpropagation* ini adalah proses pembelajaran yang dihasilkan pada *feedback output layer.*



Gambar 2.19. Struktur *feed backward*

2.11.3. *Adjust Weight*

Setelah semua faktor δ dihitung, bobot semua garis dimodifikasi bersamaan. Perubahan bobot suatu garis didasarkan atas faktor δ *neuron* dilayar

Muhammad Thoriqul Ullum, Implementasi Metode ANN (Artificial Neural Network) Pada Robot Self-Parking (Nonholonomic Robot) Menggunakan Prototype Robot Lego® Mindstorm®, 2015, UIB Repository©2015

atasnya. Sebagai contoh, perubahan bobot garis yang menuju ke layar keluaran didasarkan atas δ_k yang ada di unit keluaran. Ketiga fase tersebut diulang secara terus menerus hingga kondisi penghentian dipenuhi. Umumnya kondisi penghentian yang sering dipakai adalah jumlah iterasi atau kesalahan. Iterasi akan dihentikan jika jumlah iterasi yang dilakukan sudah melebihi jumlah maksimum iterasi yang ditetapkan, atau jika kesalahan yang terjadi sudah lebih kecil dari batas toleransi yang diijinkan.

Tahap pelatihan ini merupakan langkah bagaimana suatu jaringan saraf itu berlatih, yaitu dengan cara melakukan perubahan penimbang (sambungan antar lapisan yang membentuk jaringan melalui masing-masing unitnya). Sedangkan pemecahan masalah baru akan dilakukan jika proses pelatihan tersebut selesai, fase tersebut adalah fase *mapping* atau proses pengujian/*testing*.

Muhammad Thoriqul Ullum, Implementasi Metode ANN (Artificial Neural Network) Pada Robot Self-Parking (Nonholonomic Robot) Menggunakan Prototype Robot Lego® Mindstorm®, 2015, UIB Repository©2015