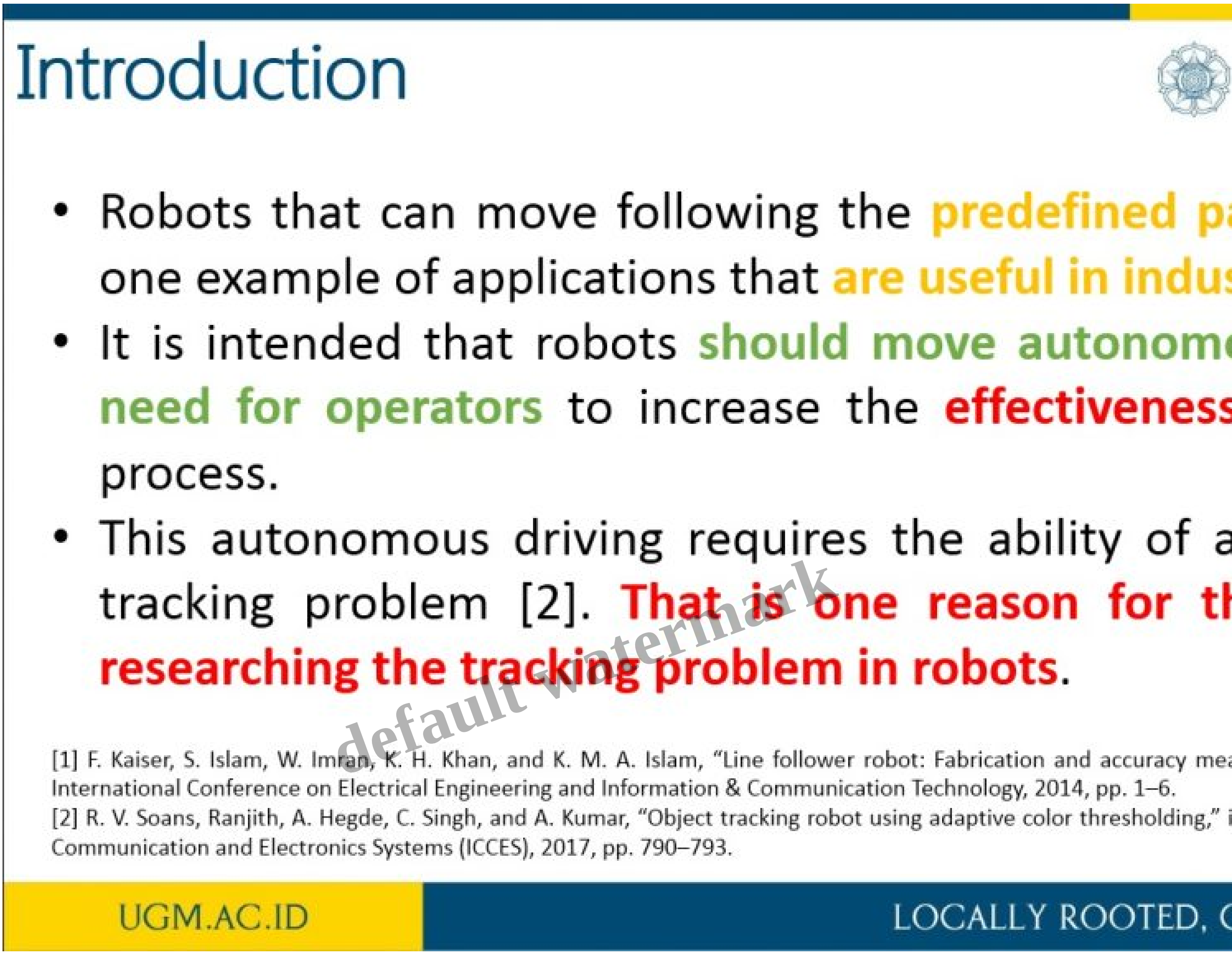
Mobile Robot

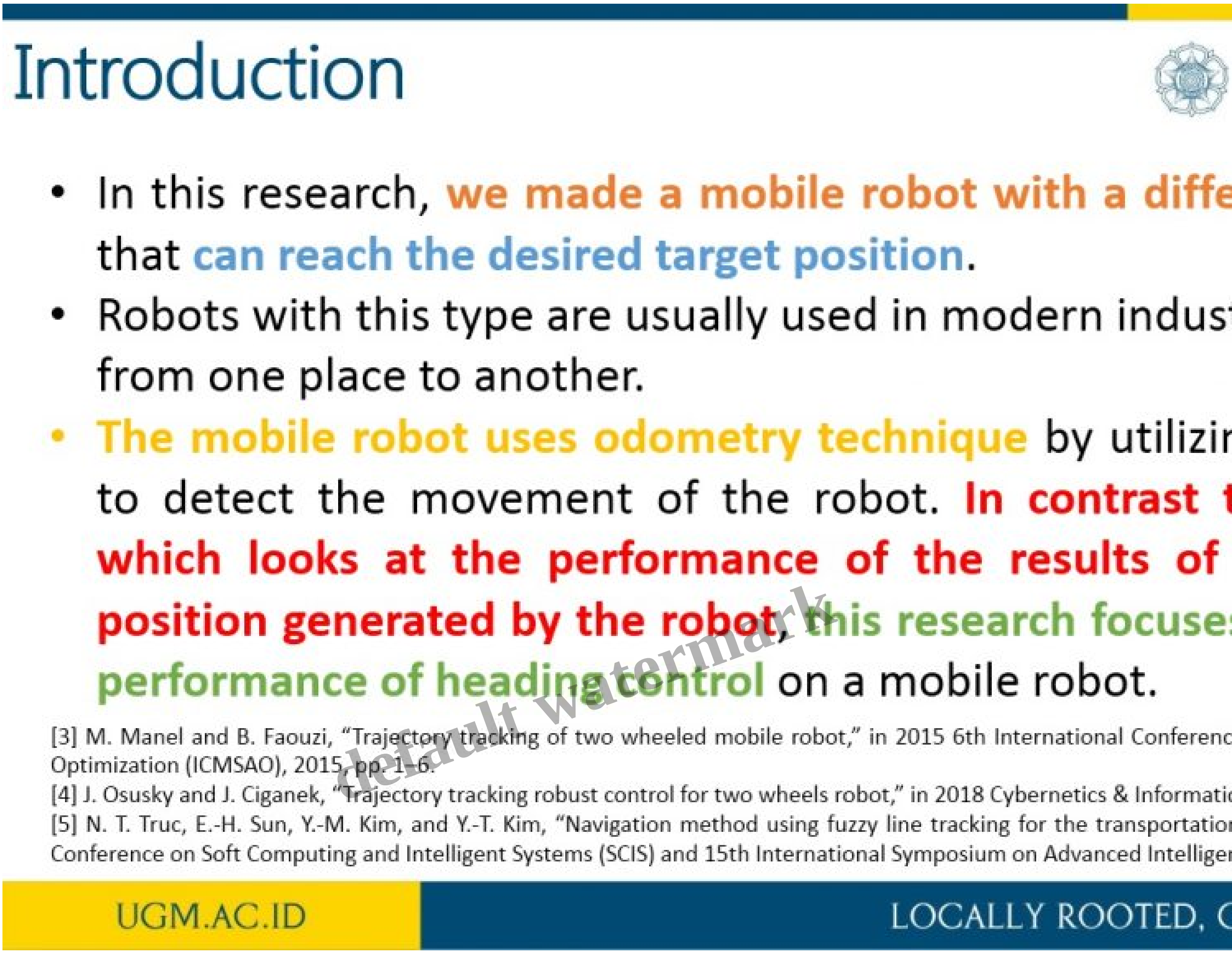
## Description

# Heading Control on Differential Drive Wheeled Mobile Robot with Odometry for Tracking Problem

**Abstract**

This research describes the process of designing odometry and heading controls from a differential steering wheeled mobile robot (DSWMR). The odometry system aims to estimate the position relative to the initial position of the robot to estimate changes in position from time to time in the trajectory tracking process. The problem that often arises in the tracking problem is the heading error that can be caused by a slip on the robot wheel or an irregularity between the speed of the DC motor on the robot wheel. Heading errors in DSWR can be obtained with the help of a rotary encoder located on a DC motor. This work applied PID control to obtain the heading error close to 0 degrees on the odometry system for trajectory tracking. It works by controlling the rotating speed of each DC motor on the robot wheel. The results of the PID control parameters implemented on the DSWMR were obtained from the results of tuning experiments with Kp = 3.0, Ki = 0.0003 and Kd = 2.5.


Introduction
• Robots that can move following the predefined pa
one example of applications that are useful in indus
• It is intended that robots should move autonomo
need for operators to increase the effectiveness
process.
• This autonomous driving requires the ability of a
tracking problem [2]. That is one reason for th
researching the tracking problem in robots.
[1] F. Kaiser, S. Islam, W. Imran, K. H. Khan, and K. M. A. Islam, “Line follower robot: Fabrication and accuracy mea
International Conference on Electrical Engineering and Information & Communication Technology, 2014, pp. 1–6.
[2] R. V. Soans, Ranjith, A. Hegde, C. Singh, and A. Kumar, “Object tracking robot using adaptive color thresholding,” i
Communication and Electronics Systems (ICCES), 2017, pp. 790–793.
UGM.AC.ID
LOCALLY ROOTED, G

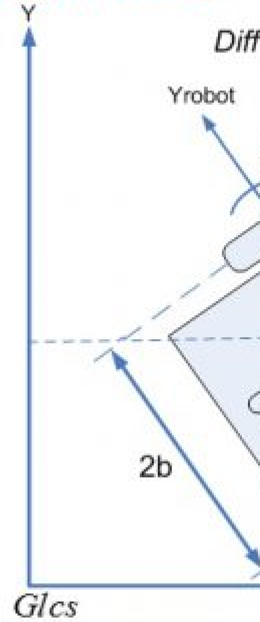

Introduction
• In this research, we made a mobile robot with a diffe
that can reach the desired target position.
• Robots with this type are usually used in modern indus
from one place to another.
• The mobile robot uses odometry technique by utilizi
to detect the movement of the robot. In contrast t
which looks at the performance of the results of
position generated by the robot, this research focuse
performance of heading control on a mobile robot.
[3] M. Manel and B. Faouzi, "Trajectory tracking of two wheeled mobile robot," in 2015 6th International Conferenc
Optimization (ICMSAO), 2015, pp. 1–6.
[4] J. Osusky and J. Ciganek, "Trajectory tracking robust control for two wheels robot," in 2018 Cybernetics & Informati
[5] N. T. Truc, E.-H. Sun, Y.-M. Kim, and Y.-T. Kim, "Navigation method using fuzzy line tracking for the transportatio
Conference on Soft Computing and Intelligent Systems (SCIS) and 15th International Symposium on Advanced Intellige
UGM.AC.ID
LOCALLY ROOTED, G

# Kinematics of Differential Steering Mobile Robot

The **position and heading** of the robot are explained by vector :

$$q = \left[ x_Q, y_Q, \varphi \right]^T$$

The relationship between the **linear velocity of the robot** $v$ and **its angular velocity** $\omega$
position and heading :




$$\begin{bmatrix} \dot{x}_Q \\ \dot{y}_Q \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \cos\varphi & 0 \\ \sin\varphi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

$$v = \frac{r(\dot{\theta}_r + \dot{\theta}_l)}{2}$$

$$\omega = \frac{r(\dot{\theta}_r - \dot{\theta}_l)}{2b}$$

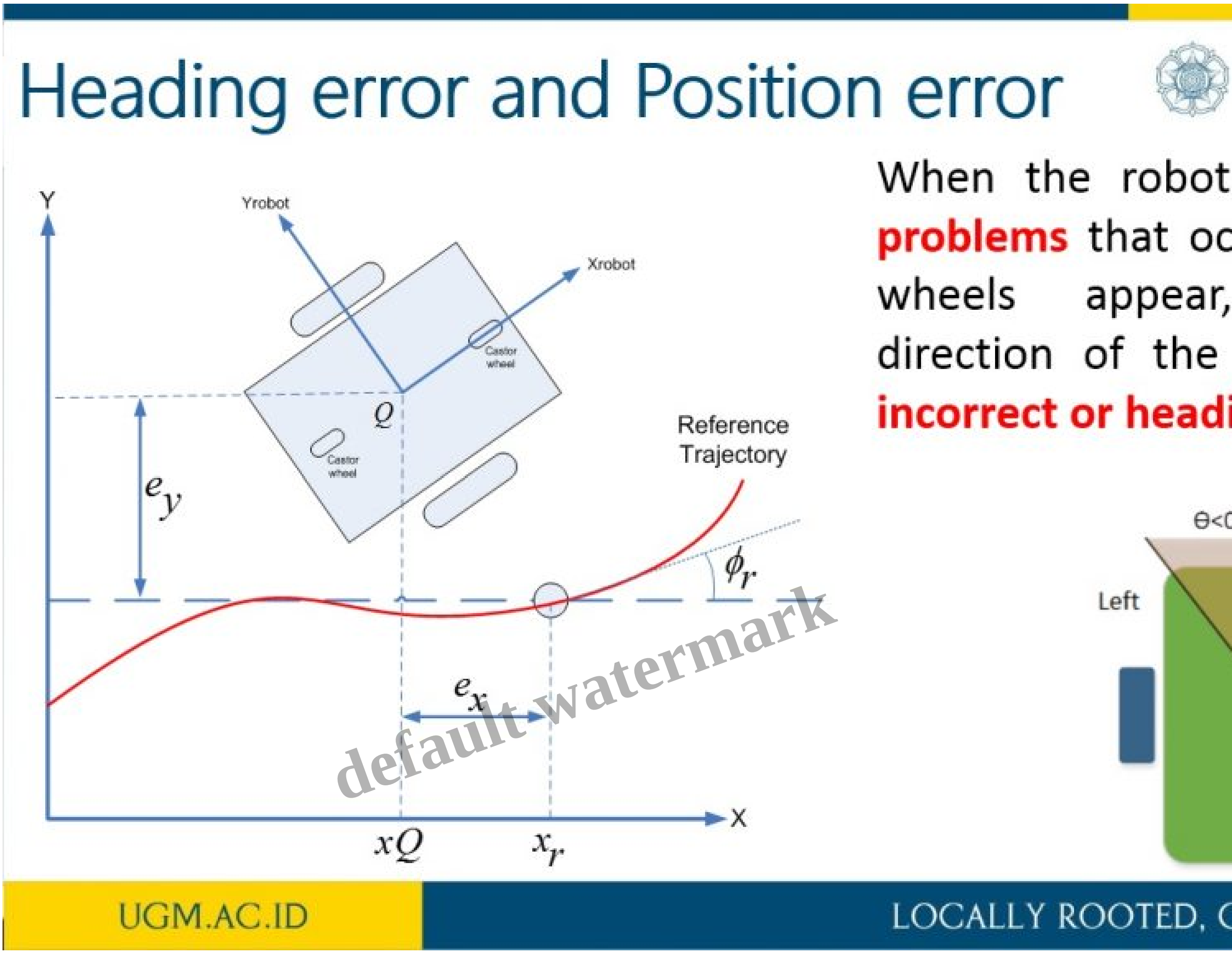
$$\dot{x}_Q = \frac{r}{2}(\dot{\theta}_r \cos\varphi + \dot{\theta}_l \cos\varphi)$$

$$\dot{y}_Q = \frac{r}{2}(\dot{\theta}_r \sin\varphi + \dot{\theta}_l \sin\varphi)$$

$$\dot{\varphi} = \frac{r(\dot{\theta}_r - \dot{\theta}_l)}{2b}$$

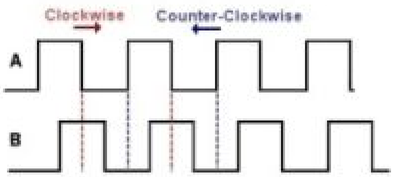


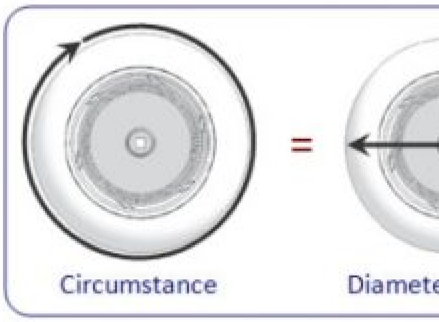
UGM.AC.ID LOCALLY ROOTED, G


Heading error and Position error
When the robot
problems that oc
wheels appear,
direction of the
incorrect or headi
Y
Yrobot
Xrobot
Castor wheel
Castor wheel
Q
Reference Trajectory
$e_y$
$\phi_r$
$e_x$
X
$xQ$
$x_r$
Left
UGM.AC.ID
LOCALLY ROOTED,

# Odometry

is used to calculate heading robots and the coordinates of the initial position.




$$Encoder\ Pulse\ /\ mm = \frac{encoder\ rotary\ resolution}{wheel\ circumference}$$

$$distance = \frac{(enc\_right + enc\_left)}{2}\ (mm)$$

$$heading = \frac{(enc\_right - enc\_left)}{l}\ (radian)$$

$$heading = \frac{(enc\_right - enc\_left)}{l}\frac{180}{\pi}\ (degree)$$

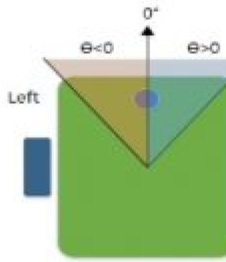


UGM.AC.ID

LOCALLY ROOTED, G

# Example

- It is known that a robot has the following details:
- Encoder resolution = 130
- Wheel diameter = 50 mm
- Distance between wheels = 120 mm
- Then: $Encoder\ Pulse\ /\ \ mm = \frac{130}{\pi d} = \frac{130}{157.0796} = 0.827606$




If the **left** encoder reads **750** and the **right** encoder rea
distance and heading of the robot.

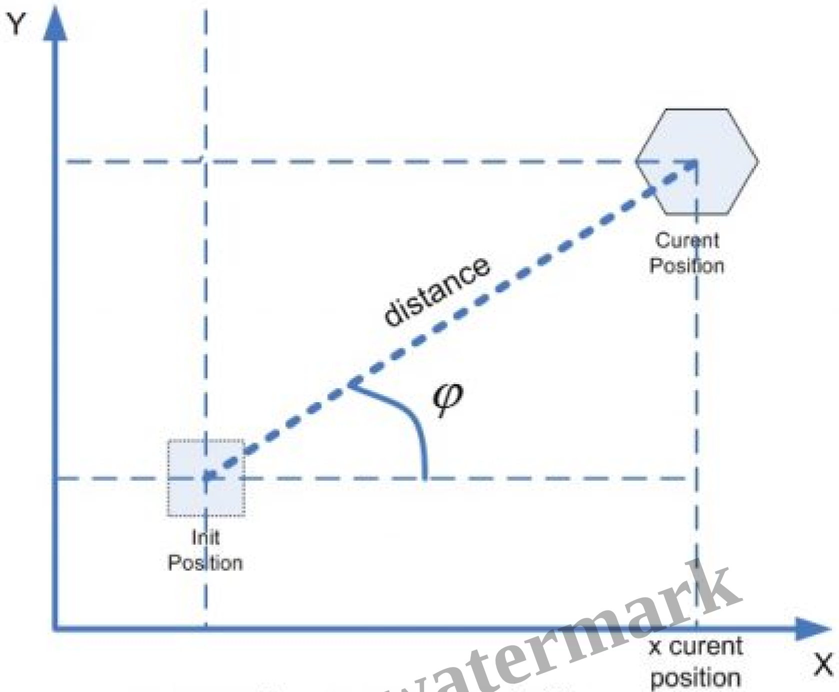
$distance = \frac{\left(\frac{750}{0.827606} + \frac{800}{0.827606}\right)}{2}$ = 936.436 mm = **93.64 cm**

$heading\ (\theta) = \frac{\left(\frac{750}{0.827606} - \right.}{12}$



UGM.AC.ID LOCALLY ROOTED, C

# Coordinates of location relative to the initial position






$$\varphi = \text{ta}$$

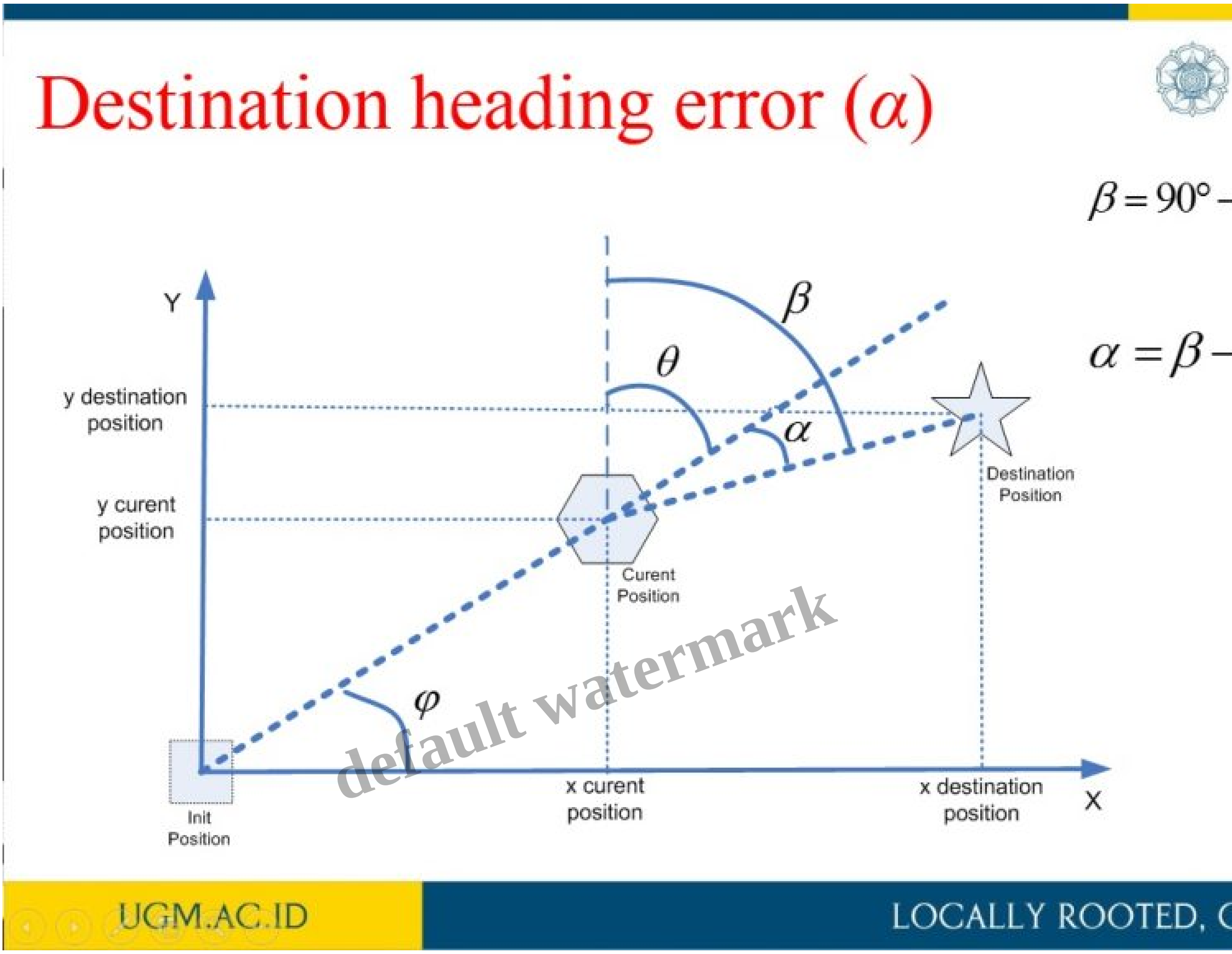
$$x\_curent = x\_pos - x\_init$$

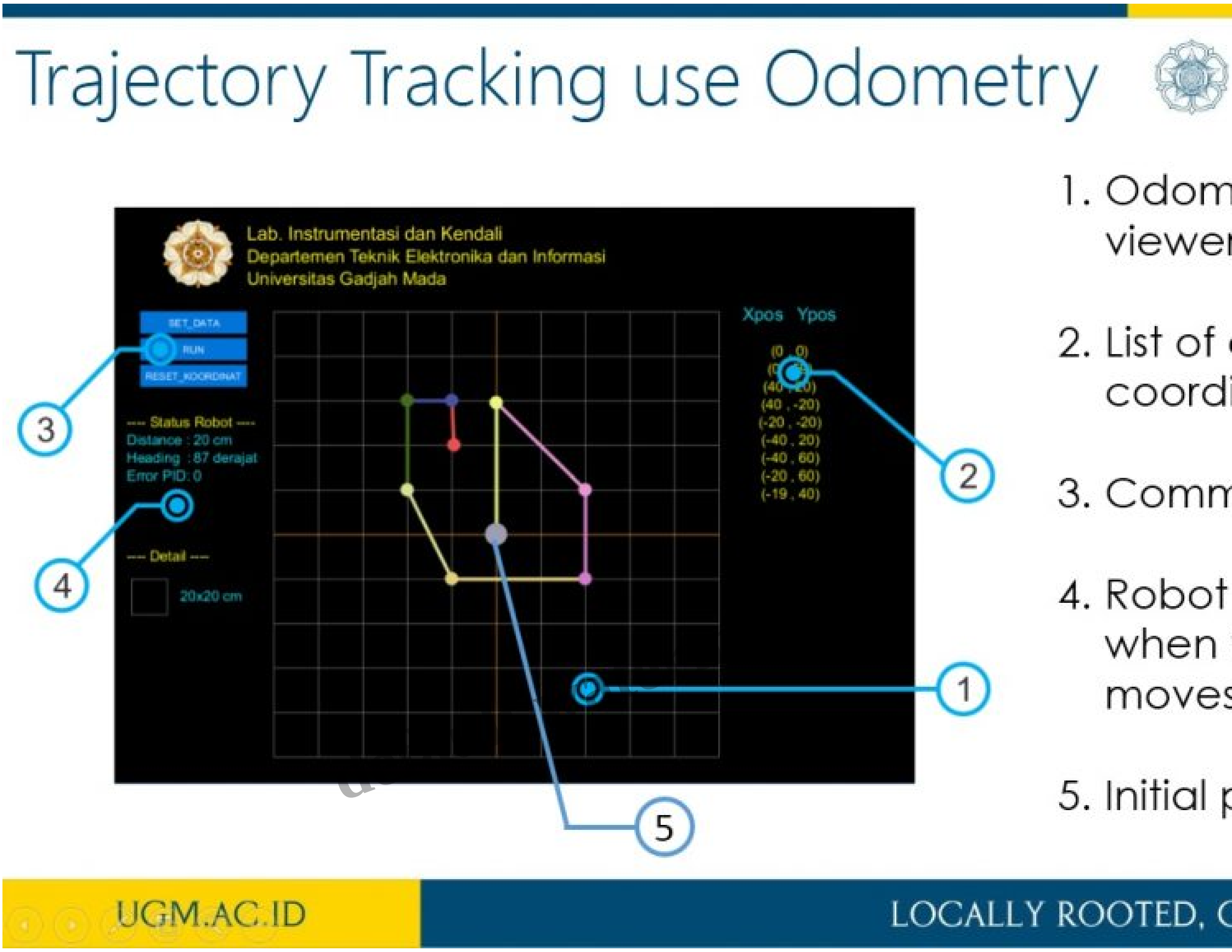
$$y\_curent = y\_pos - y\_init$$

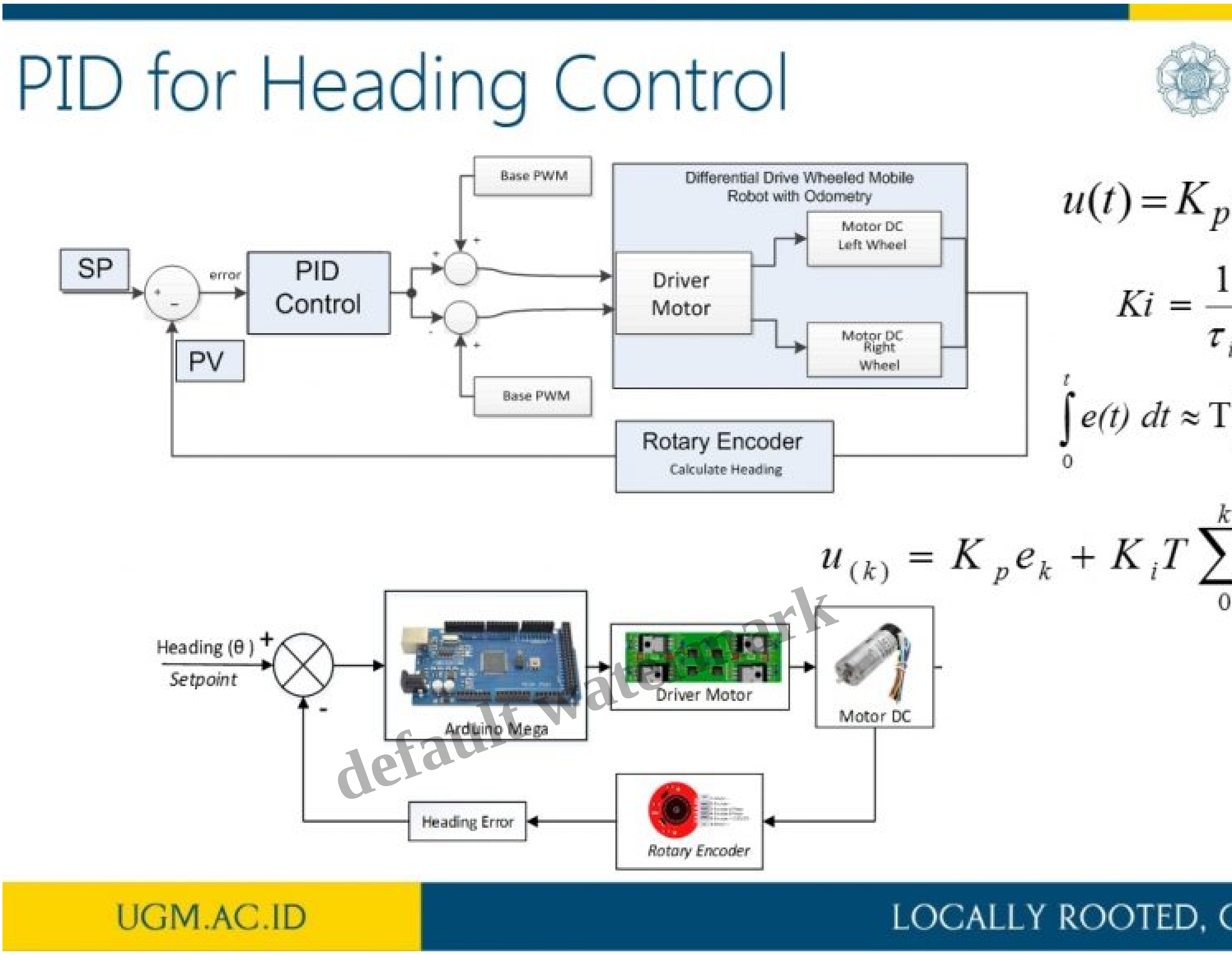
$$distance = \sqrt{x\_curent^2 + y\_curent^2}$$

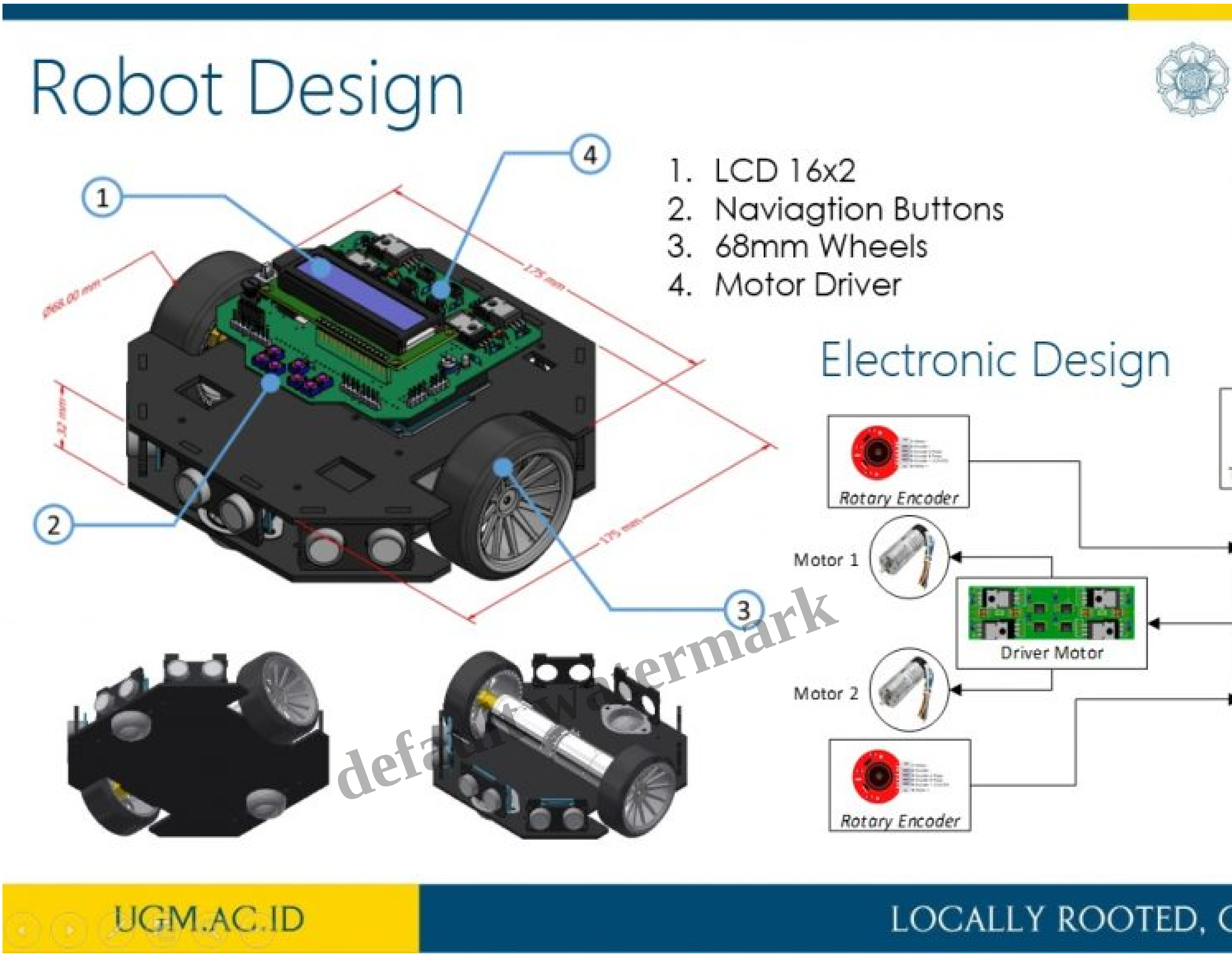
UGM.AC.ID

LOCALLY ROOTED, G


Destination heading error (α)
β = 90° –
α = β –
Y
β
θ
α
φ
y destination position
y curent position
Destination Position
Curent Position
Init Position
x curent position
x destination position
X
UGM.AC.ID
LOCALLY ROOTED,


Trajectory Tracking use Odometry
Lab. Instrumentasi dan Kendali
Departemen Teknik Elektronika dan Informasi
Universitas Gadjah Mada
SET_DATA
RUN
RESET_KOORDINAT
--- Status Robot ---
Distance : 20 cm
Heading : 87 derajat
Error PID: 0
--- Detail ---
20x20 cm
Xpos Ypos
(0 , 0)
(40 , -20)
(-20 , -20)
(-40 , 20)
(-40 , 60)
(-20 , 60)
(-19 , 40)
1
2
3
4
5
1. Odom
viewe
2. List of
coordi
3. Comm
4. Robot
when
moves
5. Initial
UGM.AC.ID
LOCALLY ROOTED, C


PID for Heading Control
Base PWM
Differential Drive Wheeled Mobile Robot with Odometry
Motor DC Left Wheel
SP
error
PID Control
Driver Motor
Motor DC Right Wheel
PV
Base PWM
Rotary Encoder
Calculate Heading
Heading (θ)
Setpoint
Arduino Mega
Driver Motor
Motor DC
Heading Error
Rotary Encoder
UGM.AC.ID
LOCALLY ROOTED, G





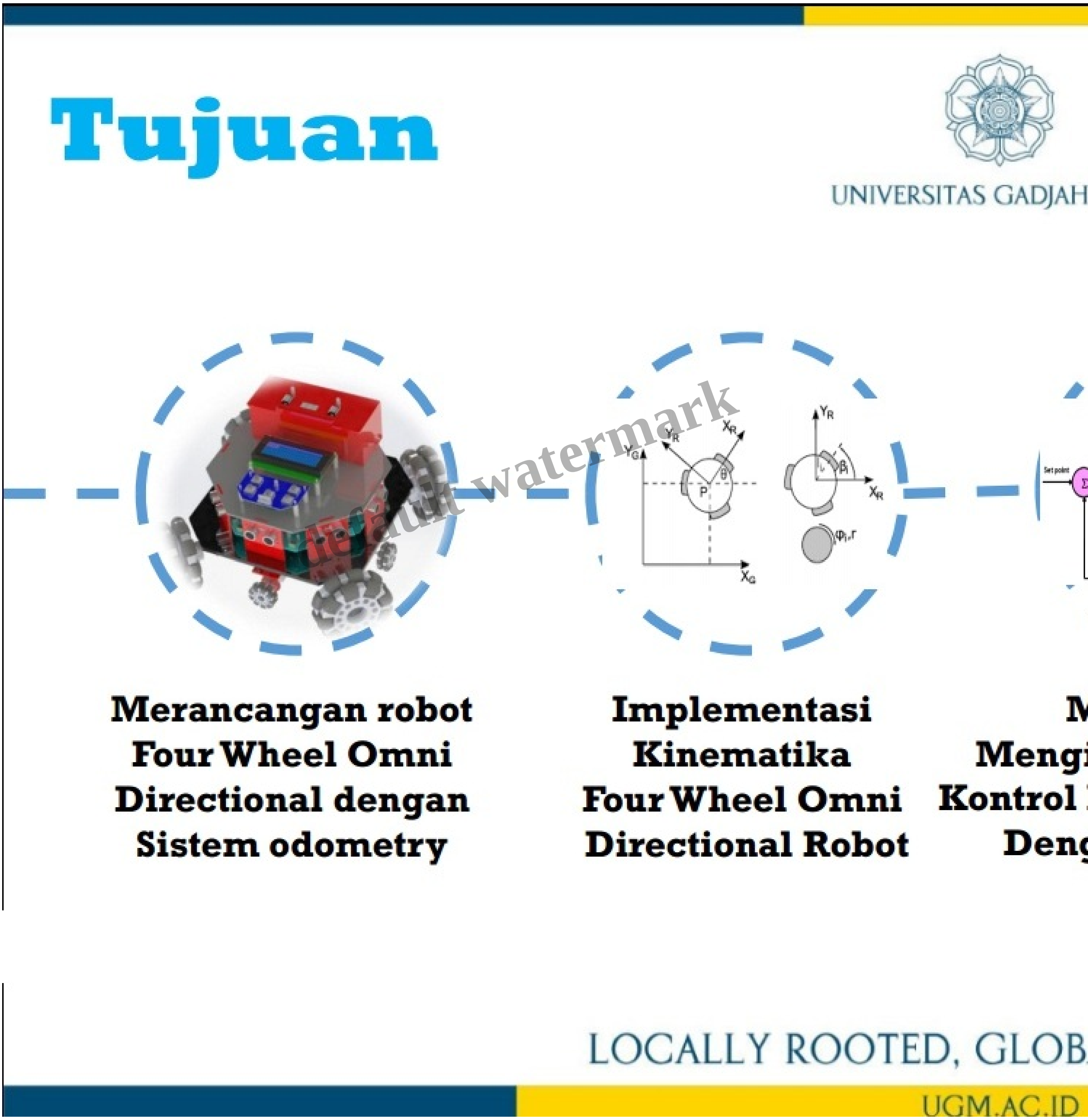

more detail visit our publication [here.]

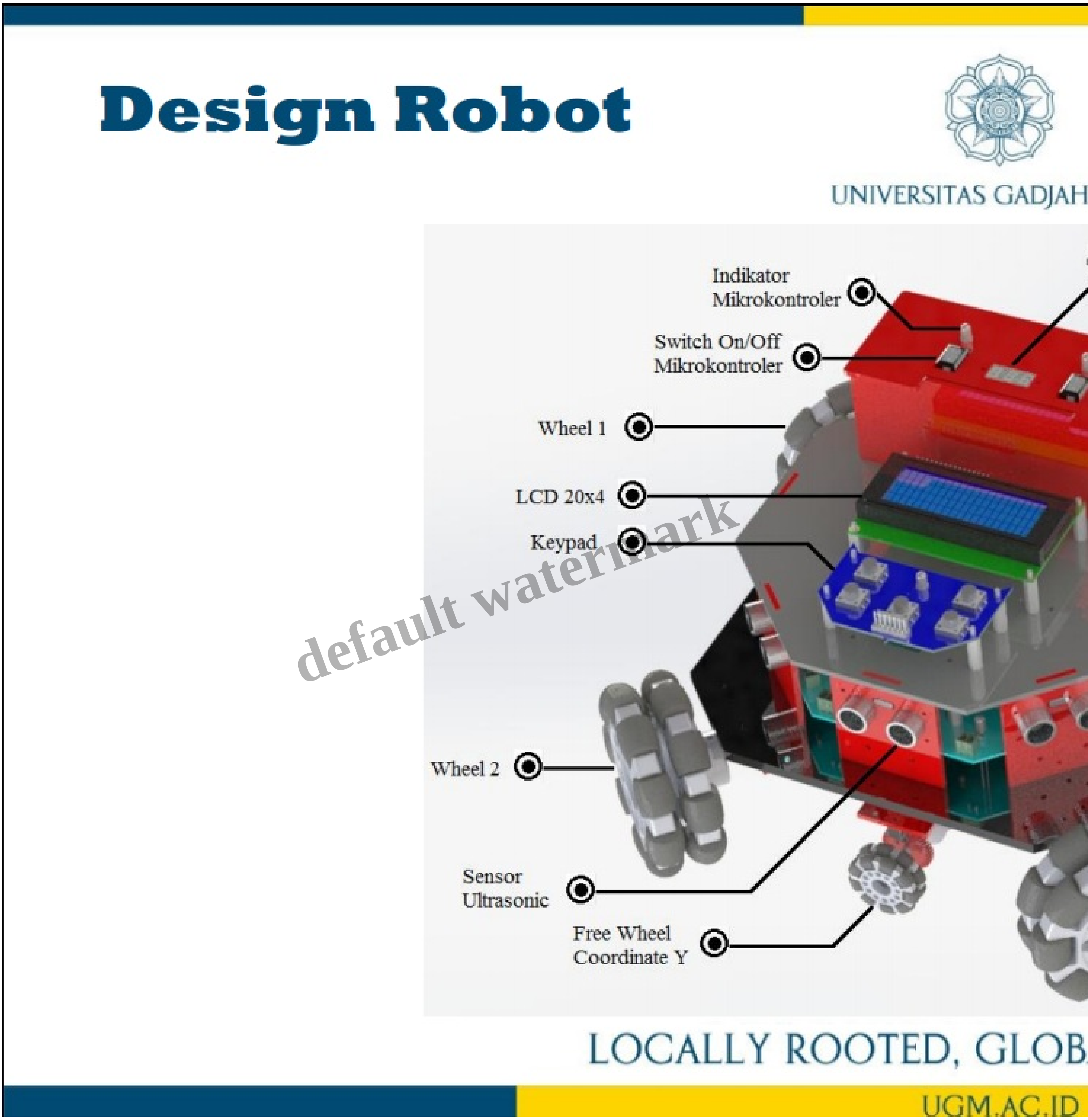
# [Trajectory Tracking pada Robot Omni dengan Metode Odometry]

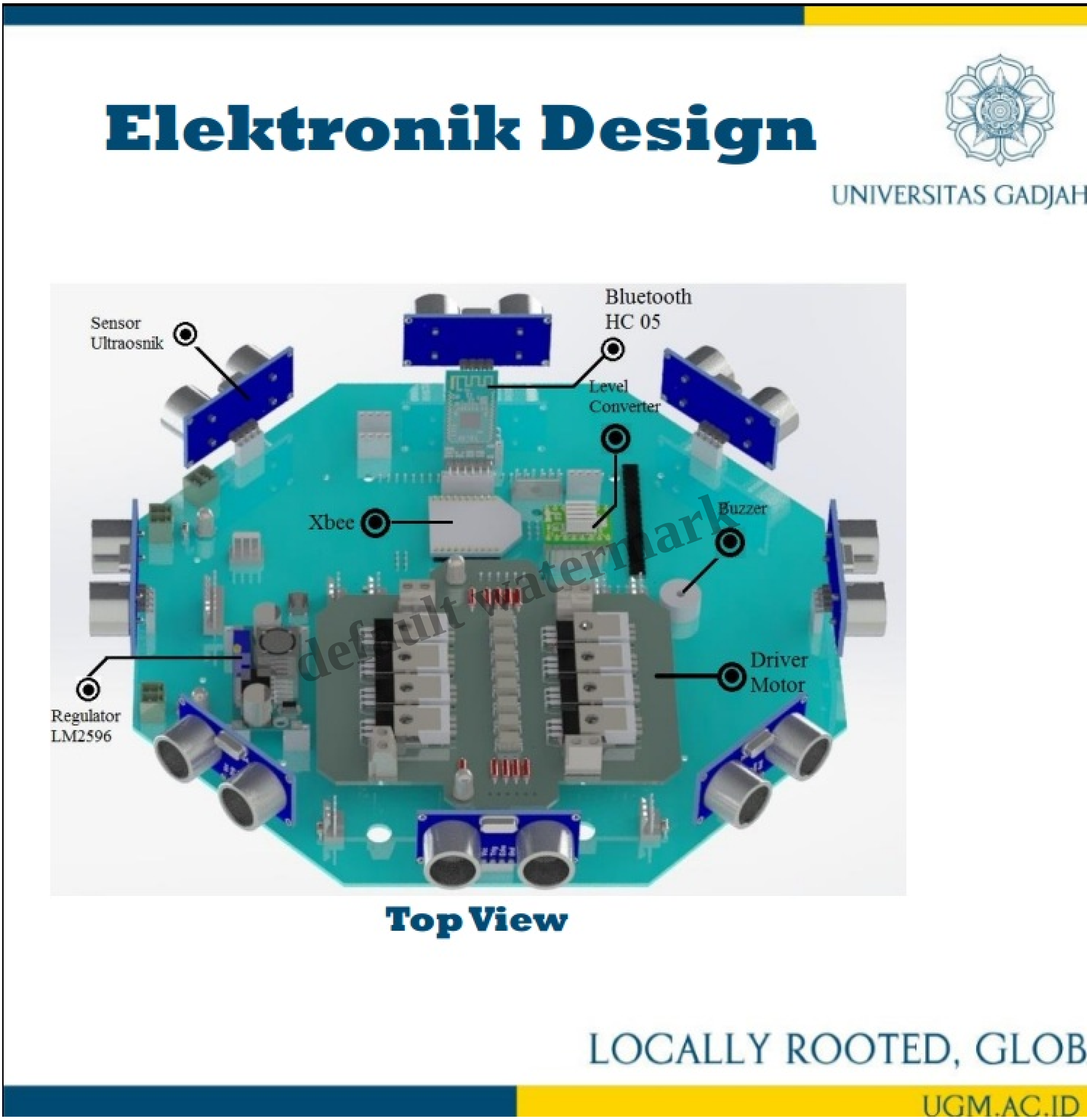
**Abstrak**

Penelitian ini memaparkan trajectory tracking pada robot omni dengan metode odometry. Sistem odometry bertujuan untuk memperkirakan posisi relatif terhadap posisi awal robot omni untuk memperkirakan perubahan posisi dari waktu ke waktu. Sensor rotary encoder digunakan untuk mencacah pergerakan robot omni pada koordinat x dan y pada proses perhitungan odometry. Selanjutnya, dengan menggunakan kinematika balik pada robot omni, nilai kecepatan putar masing-masing motor DC pada roda robot omni diperoleh. Selain itu, untuk memperoleh hasil pergerakan robot yang baik pada sistem odometry, kendali PID diterapkan untuk mengendalikan kecepatan putar masing-masing motor DC pada roda robot omni. Dengan kinematika balik dan sistem odometry,

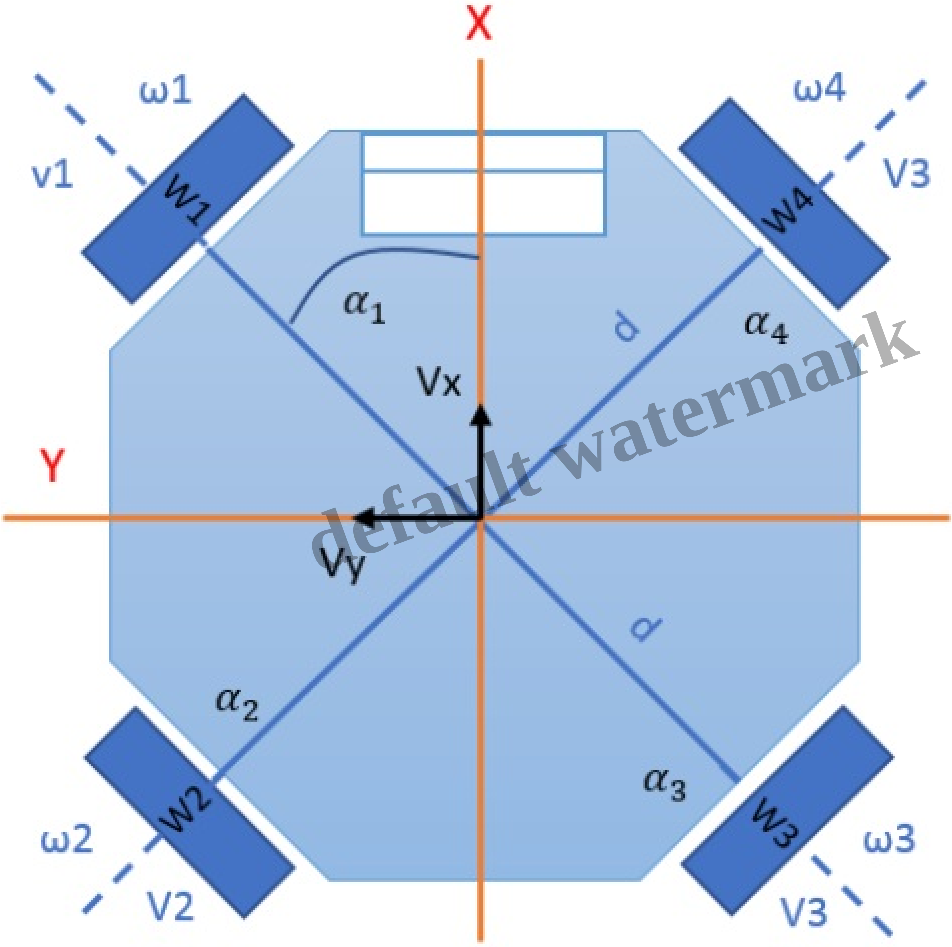
desain trajectory robot omni dapat dengan mudah dibangun. Untuk menguji kinerja metode odometry dalam melakukan proses trajectory tracking, terdapat tiga jenis pola pengujian trajectory, yaitu persegi panjang, segitiga sama sisi, dan segitiga sama kaki. Dari hasil pengujian ini, diperoleh nilai kesalahan di bawah 5%.




Design Robot
UNIVERSITAS GADJAH
Indikator Mikrokontroler
Switch On/Off Mikrokontroler
Wheel 1
LCD 20x4
Keypad
Wheel 2
Sensor Ultrasonic
Free Wheel Coordinate Y
LOCALLY ROOTED, GLOB
UGM.AC.ID


Elektronik Design
UNIVERSITAS GADJAH
Sensor Ultraosnik
Bluetooth HC 05
Level Converter
Xbee
Buzzer
Regulator LM2596
Driver Motor
Top View
LOCALLY ROOTED, GLOB
UGM.AC.ID

# Kinematika *Four Wheel Omni Direc*



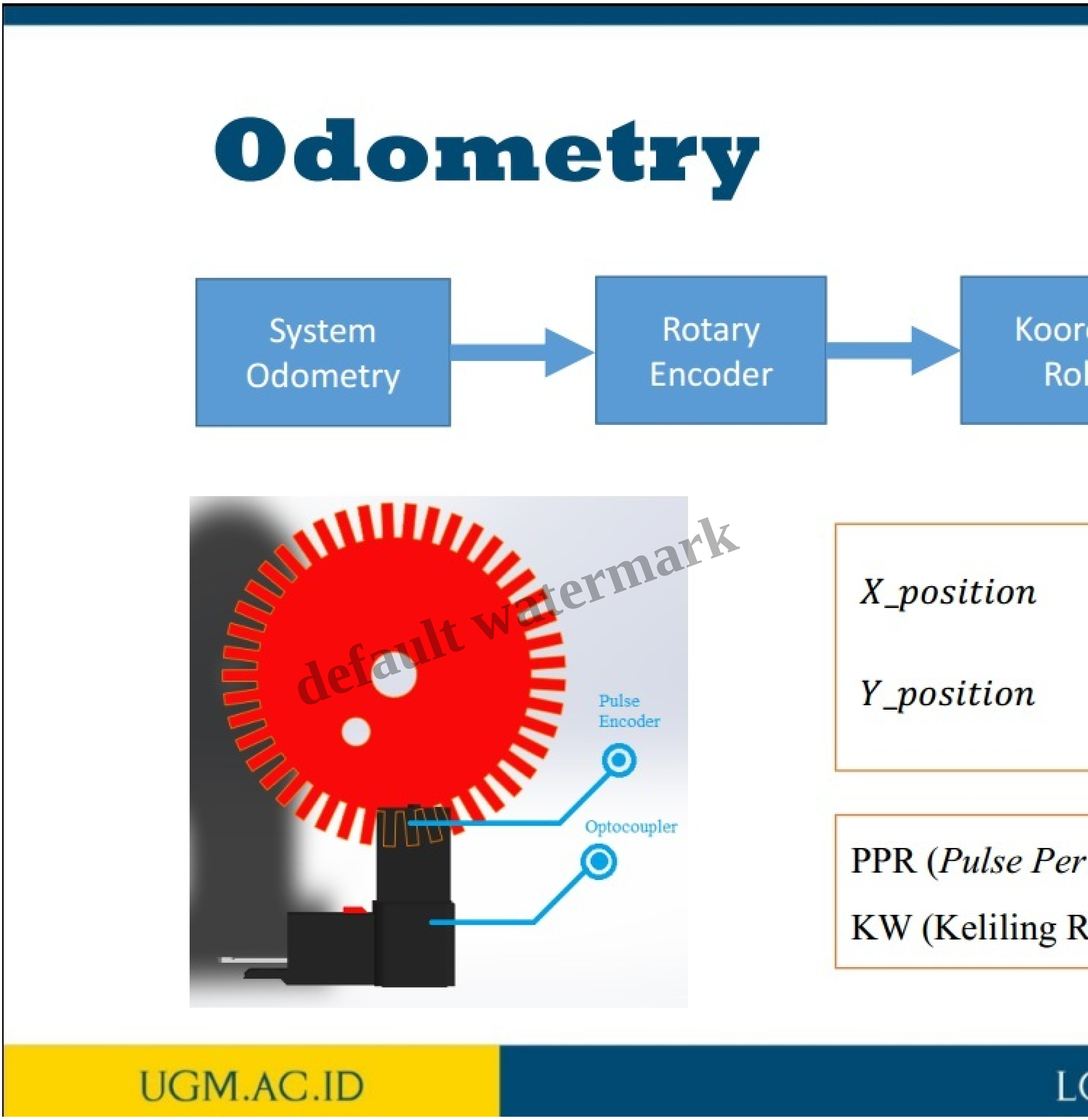

$$V_{w(n)} = \begin{bmatrix} -\sin(\theta \\ -\sin( \\ -\sin( \\ -\sin( \end{bmatrix}$$

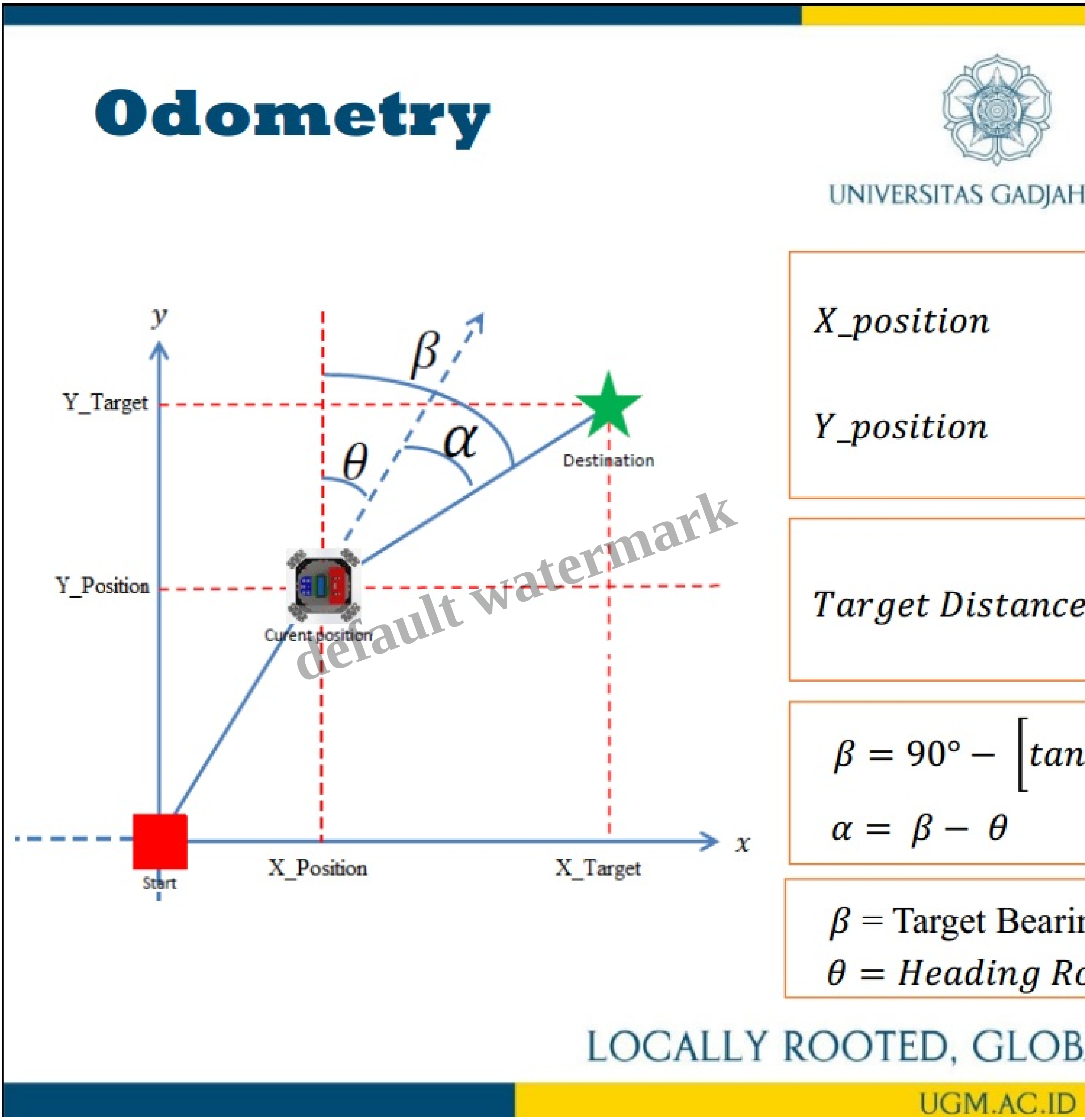
$$V_x = X_{target}$$
$$V_y = Y_{target}$$
$$\theta = heading$$

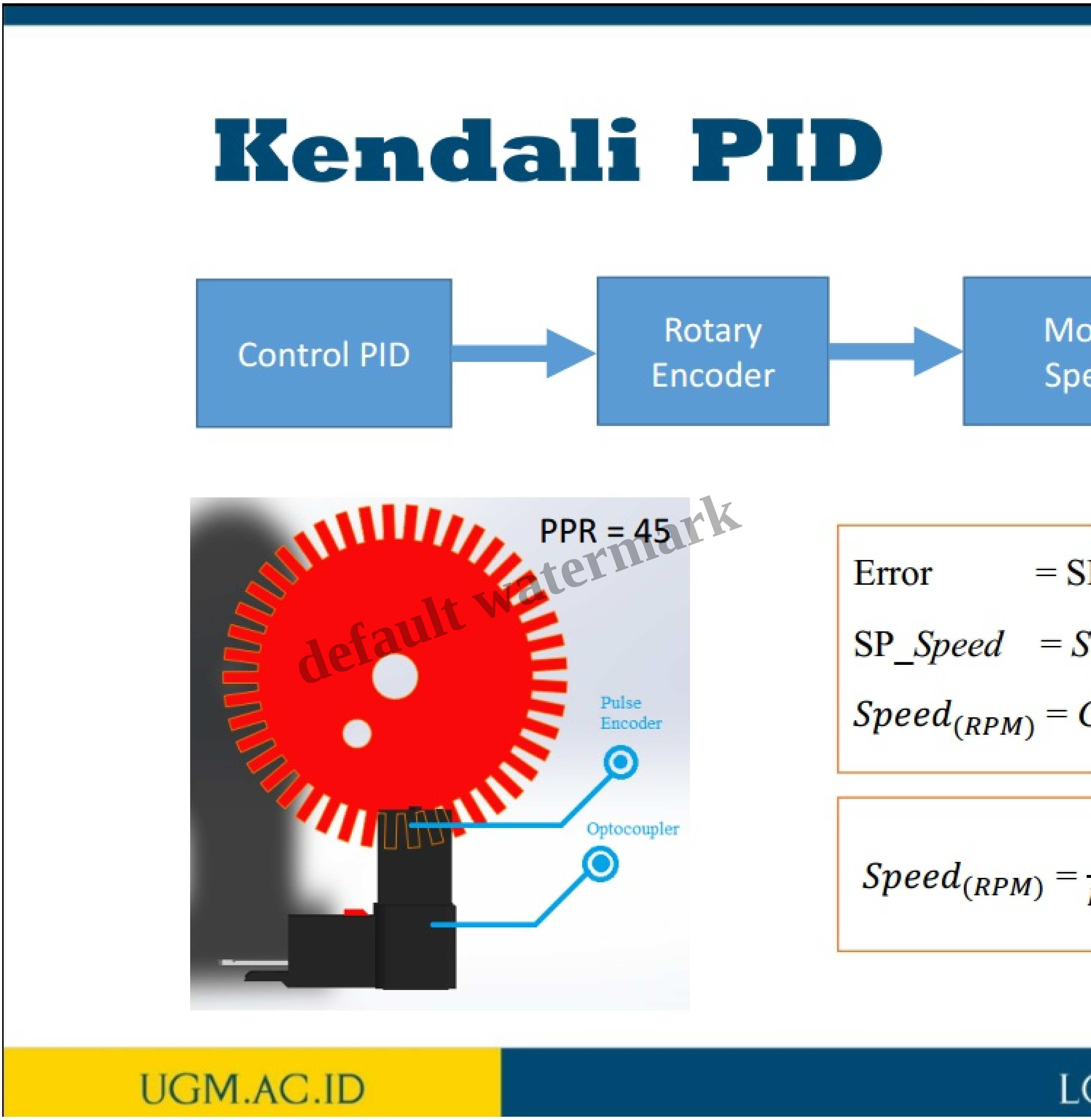


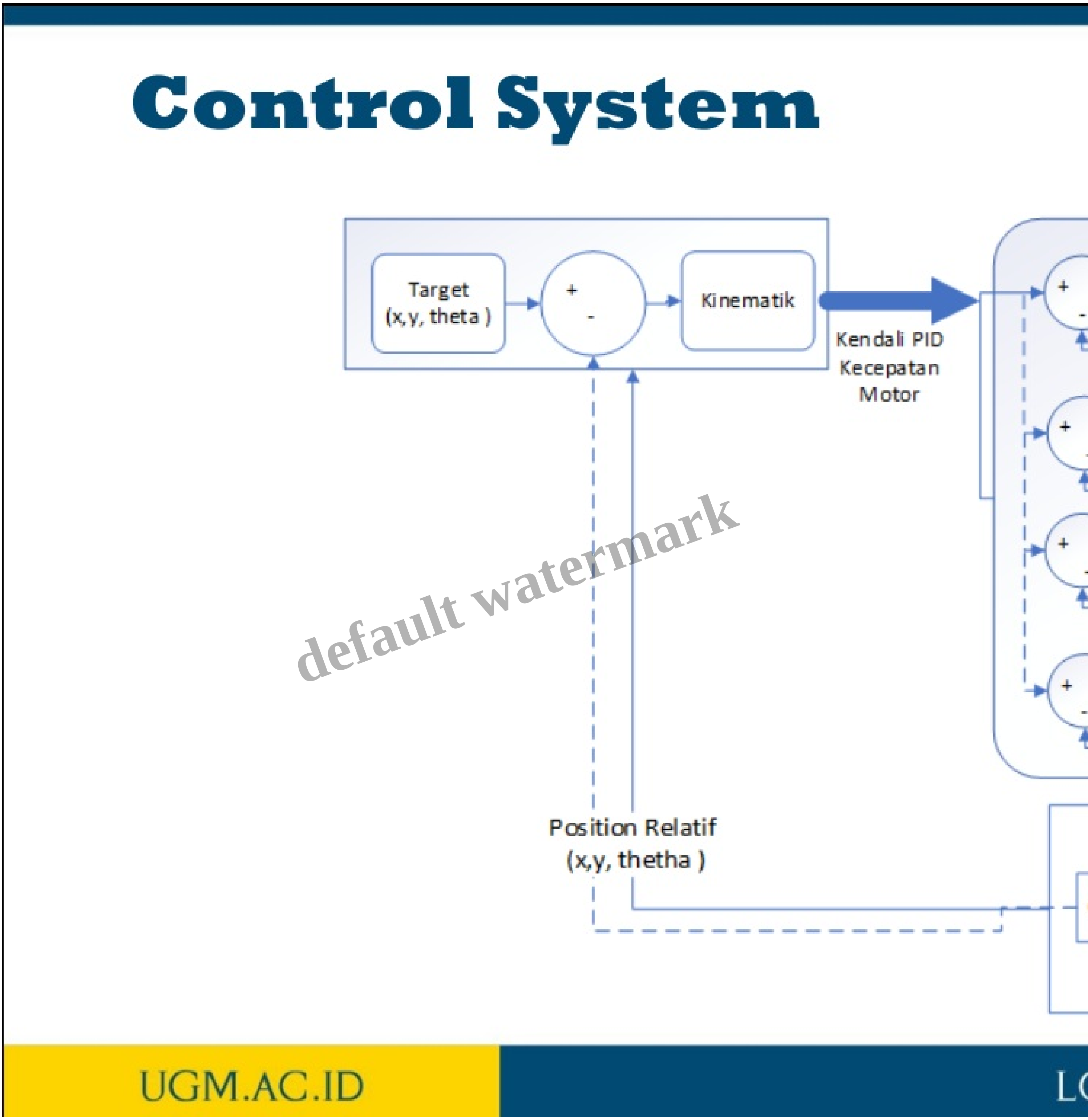
UGM.AC.ID

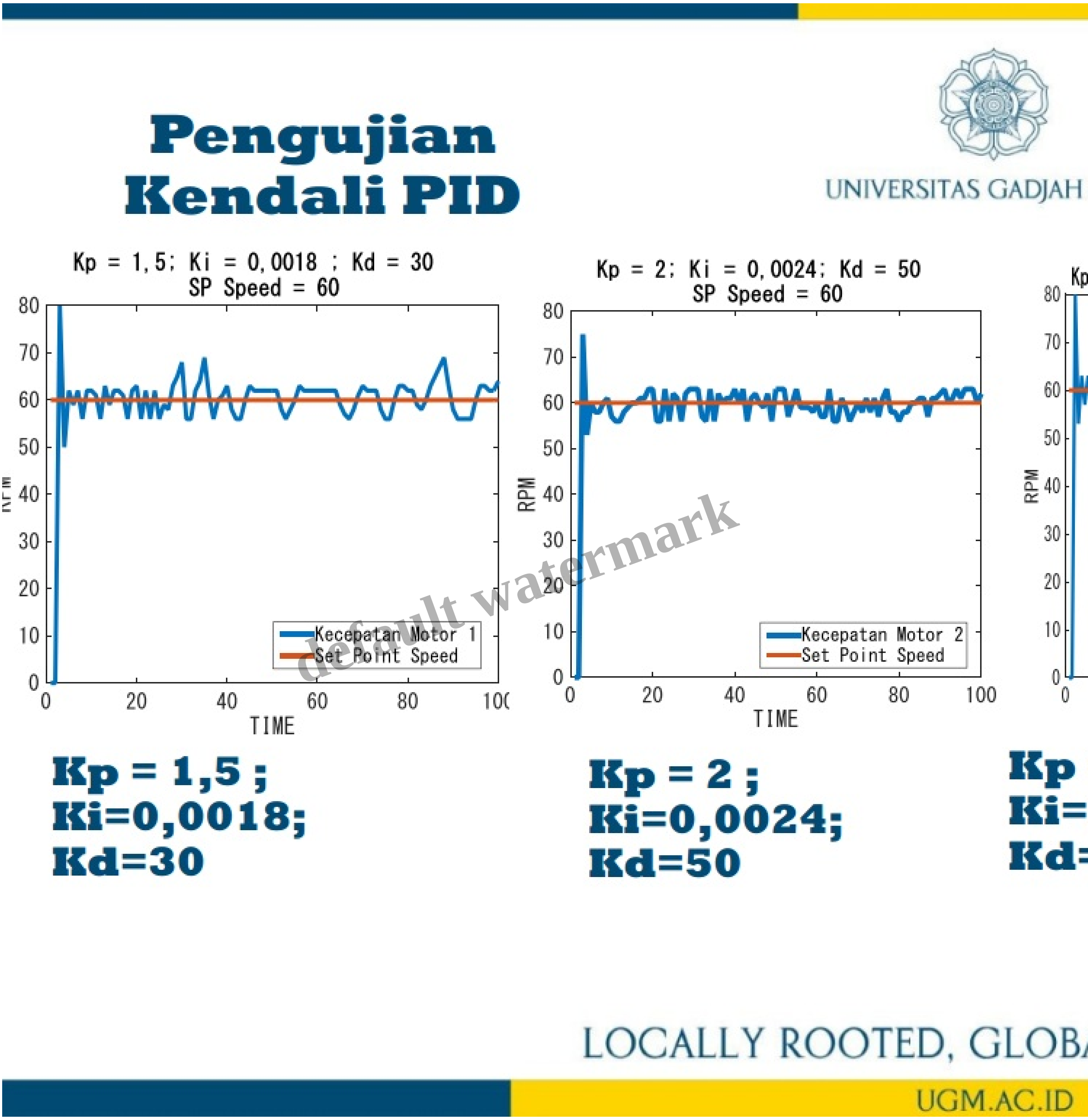
LC


Odometry
System Odometry
Rotary Encoder
X_position
Y_position
Pulse Encoder
Optocoupler
PPR (Pulse Per
KW (Keliling R
default watermark
UGM.AC.ID


Odometry
UNIVERSITAS GADJAH
y
Y_Target
Y_Position
β
θ
α
Destination
Curent position
X_Position
X_Target
x
Start
X_position
Y_position
Target Distance
β = 90° – [tan
α = β – θ
β = Target Beari
θ = Heading R
LOCALLY ROOTED, GLOB
UGM.AC.ID

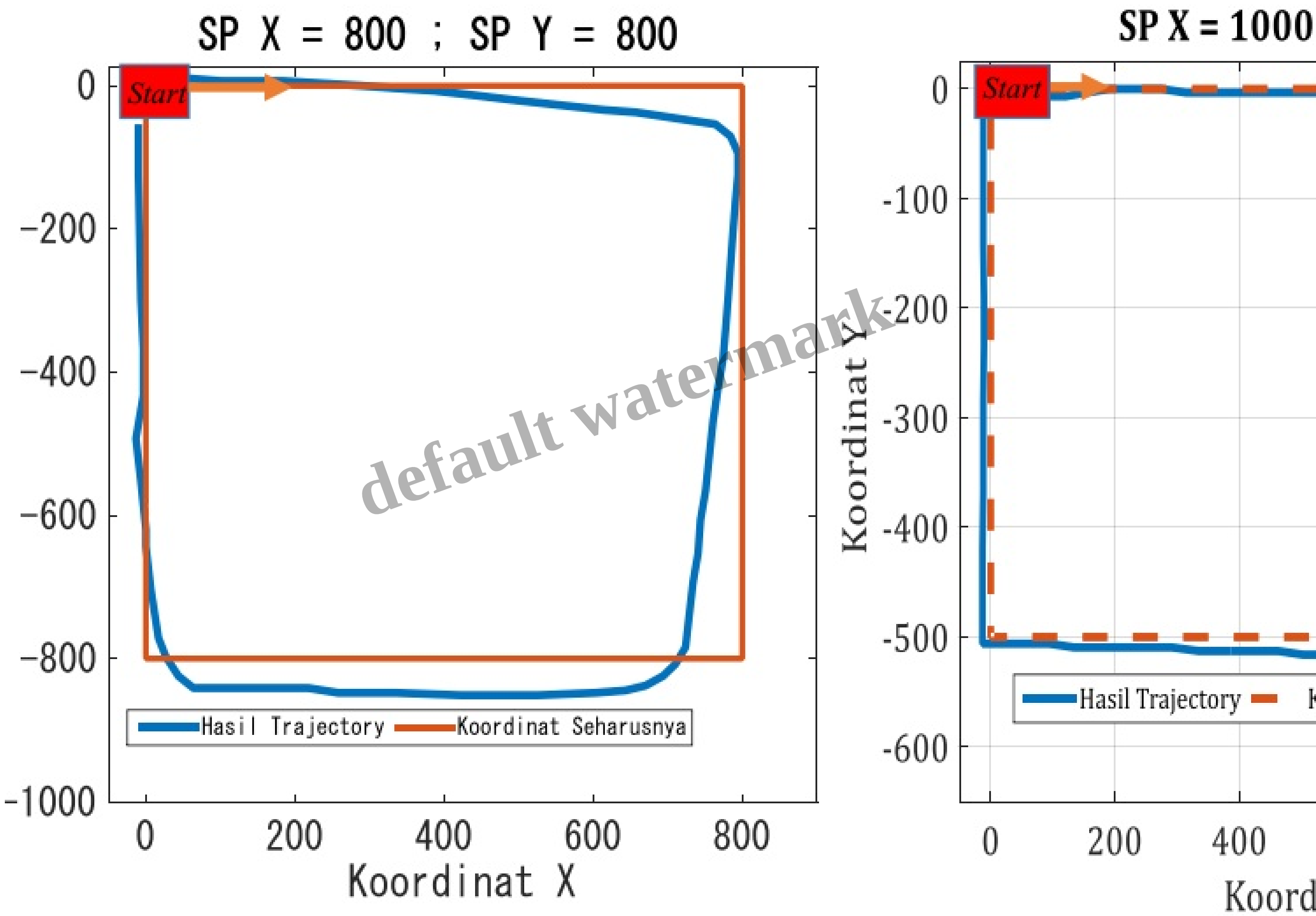

Kendali PID
Control PID
Rotary Encoder
PPR = 45
Pulse Encoder
Optocoupler
Error
SP_Speed
Speed(RPM)
Speed(RPM)
UGM.AC.ID


Control System
Target
(x,y, theta )
+
-
Kinematik
Kendali PID
Kecepatan
Motor
Position Relatif
(x,y, thetha )
UGM.AC.ID


Pengujian
Kendali PID
UNIVERSITAS GADJAH
Kp = 1,5; Ki = 0,0018 ; Kd = 30
SP Speed = 60
Kecepatan Motor 1
Set Point Speed
TIME
Kp = 2; Ki = 0,0024; Kd = 50
SP Speed = 60
RPM
Kecepatan Motor 2
Set Point Speed
TIME
Kp = 1,5 ;
Ki=0,0018;
Kd=30
Kp = 2 ;
Ki=0,0024;
Kd=50
LOCALLY ROOTED, GLOBA
UGM.AC.ID

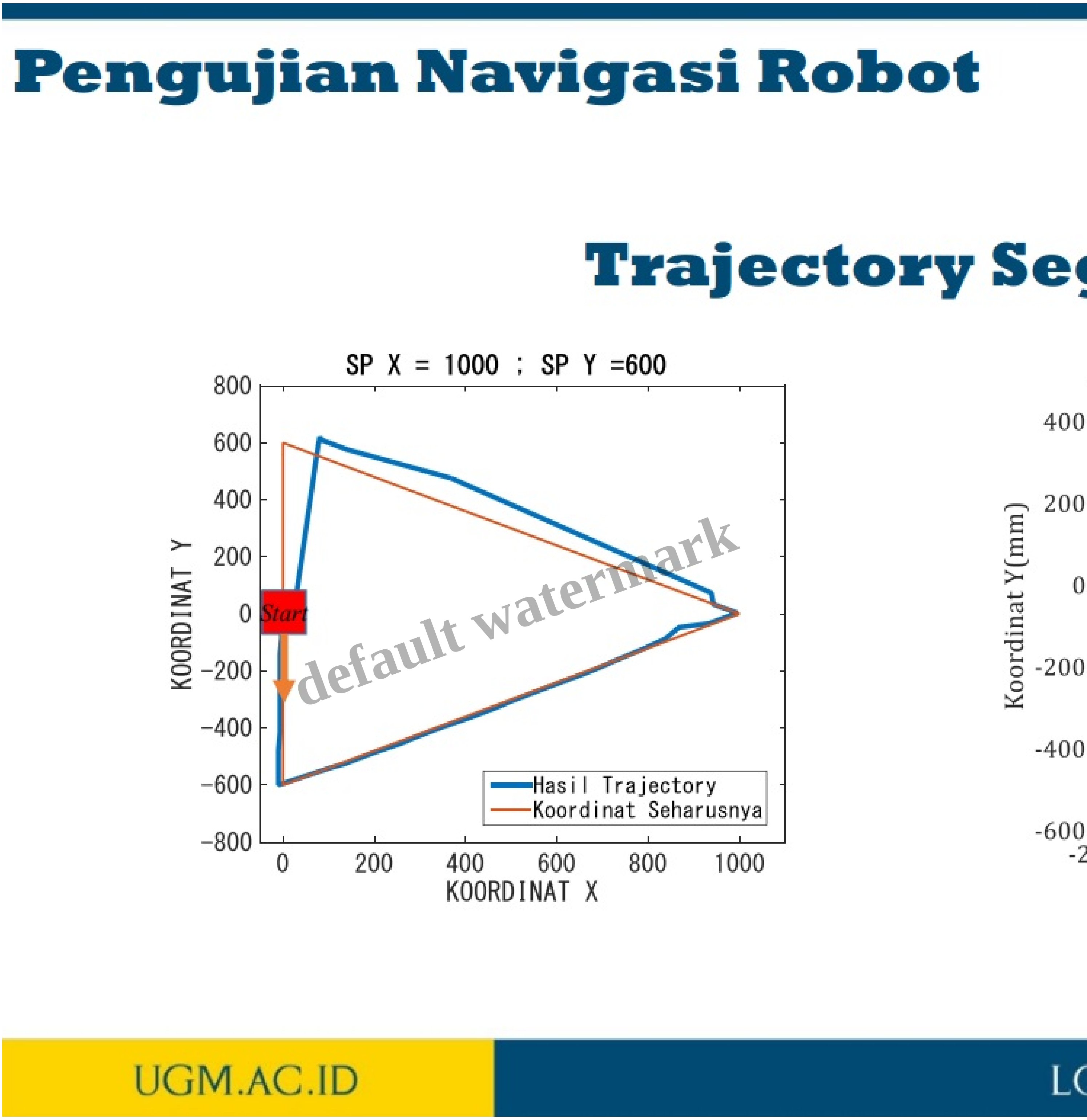
# Pengujian Navigasi Robot

## Trajectory F




UGM.AC.ID

LC

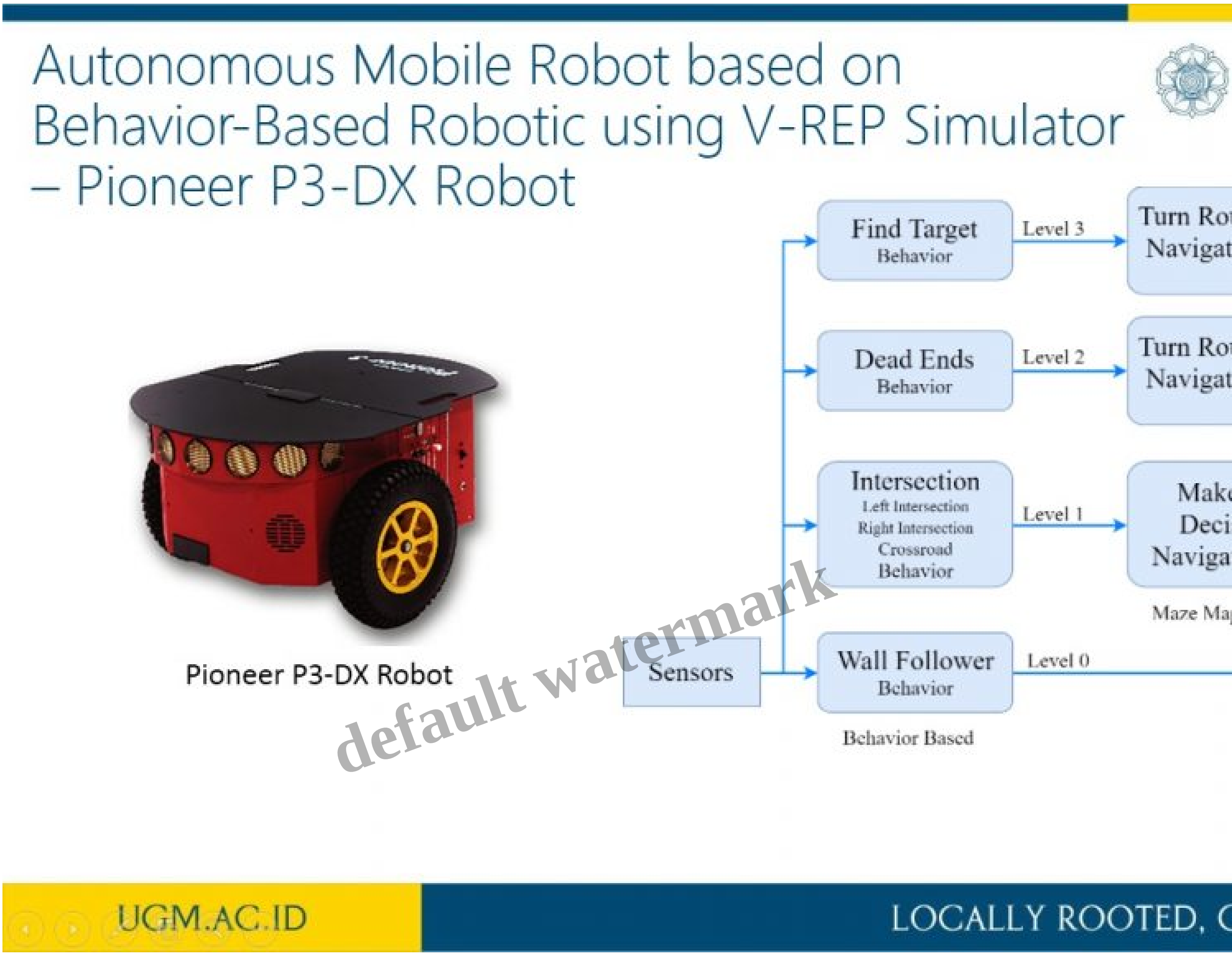


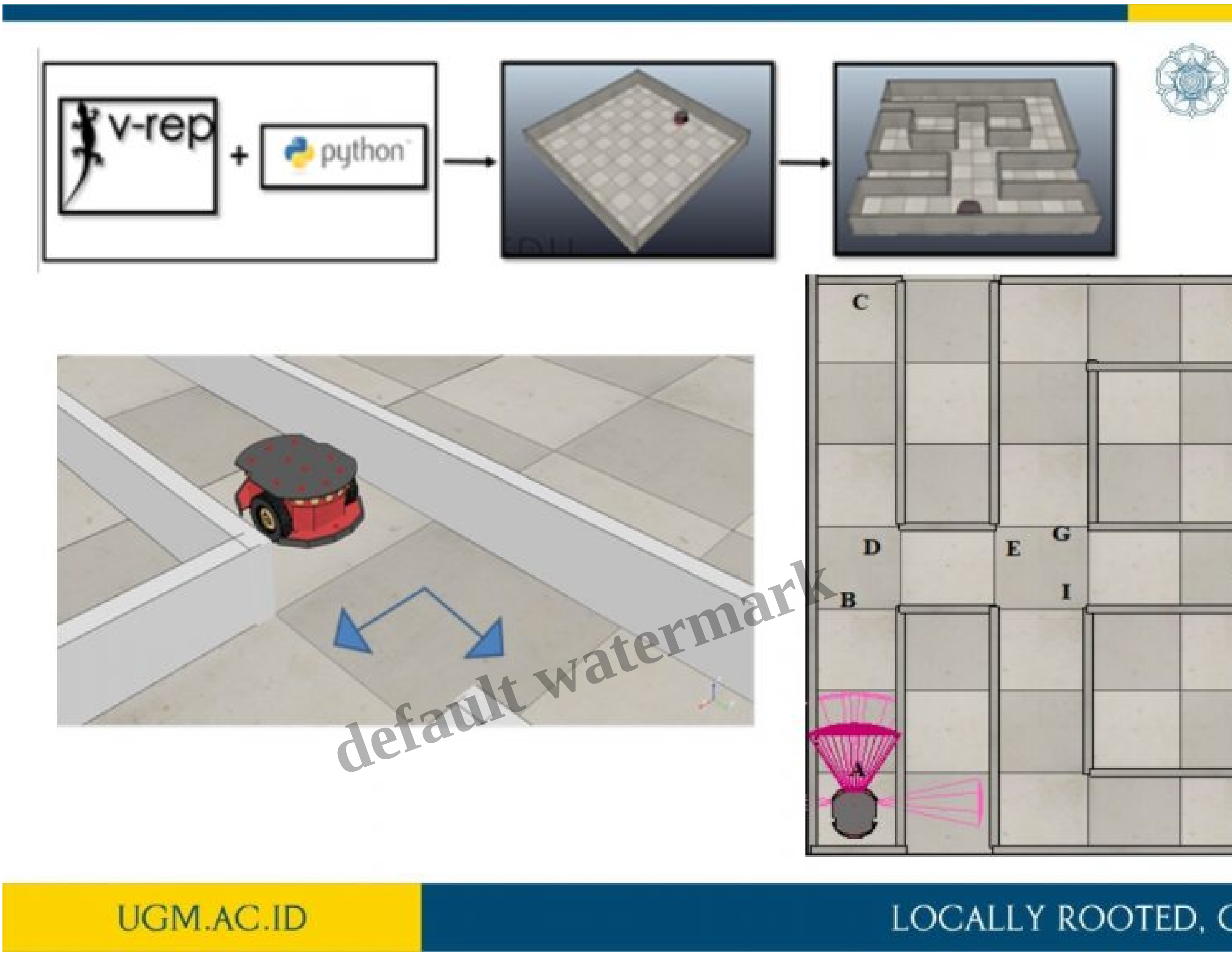

informasi lebih lanjut, silakan lihat publikasi kami di [sini](#).

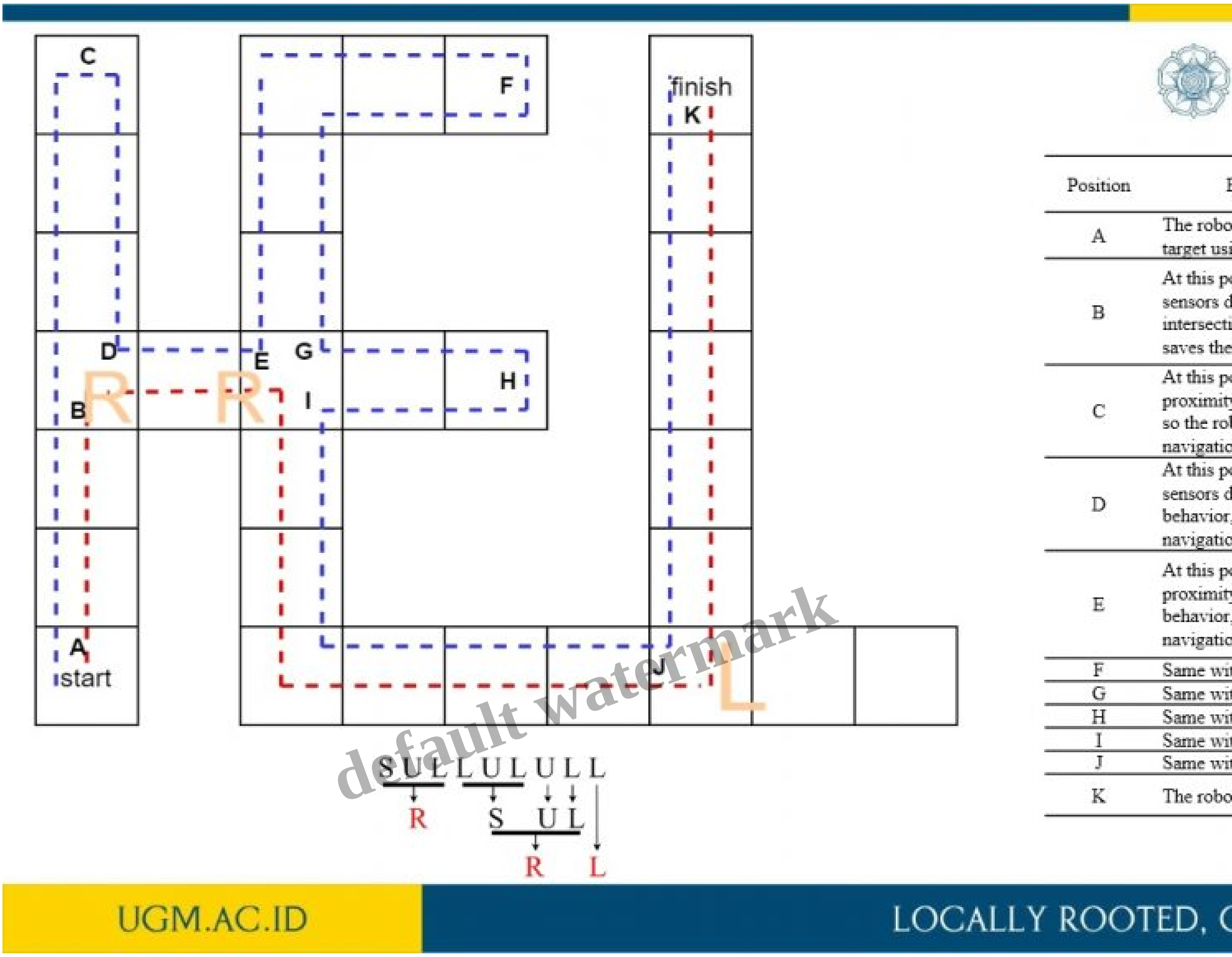
# Autonomous Mobile Robot based on Behavior-Based Robotic using V-REP Simulator – Pioneer P3-DX Robot

**Abstract**

This research describes the design and implementation of behavior-based robotic (BBR) algorithm on a wheeled mobile robot (WMR) Pioneer P3-DX in a maze exploration mission using V-REP simulator. This robot must trace and search for targets placed randomly on a labyrinth. After successfully meeting the objective, robot runs back to home position using the nearest path. Robot navigation system applies BBR algorithm to reach the target using behavior modules which work simultaneously to obtain the desired robot's trajectory. The most fundamental behavior which is highly affordable to build on the robot system is a wall-following behavior. To make the robot could follow the wall in a safe, smooth and responsive condition, proportional-integral-derivative (PID) controller is applied. PID controller runs by utilizing the reading of sixteen proximity sensors carried on Pioneer P3-DX robot toward the expected wall distance while the robot is exploring the labyrinth. To ensure the designed system works properly, several tests were conducted, including BBR test and PID controller test. BBR test shows that the system can choose the shortest track when returning to home position. The PID controller test produces robot movement with maximum deviation and settling time for about 0.013 m and 30 seconds, respectively.


Autonomous Mobile Robot based on Behavior-Based Robotic using V-REP Simulator – Pioneer P3-DX Robot
Pioneer P3-DX Robot
Sensors
Find Target
Behavior
Level 3
Dead Ends
Behavior
Level 2
Intersection
Left Intersection
Right Intersection
Crossroad
Behavior
Level 1
Wall Follower
Behavior
Level 0
Behavior Based
UGM.AC.ID
LOCALLY ROOTED,


v-rep
+
python
C
D
E
G
B
I
A
UGM.AC.ID
LOCALLY ROOTED,





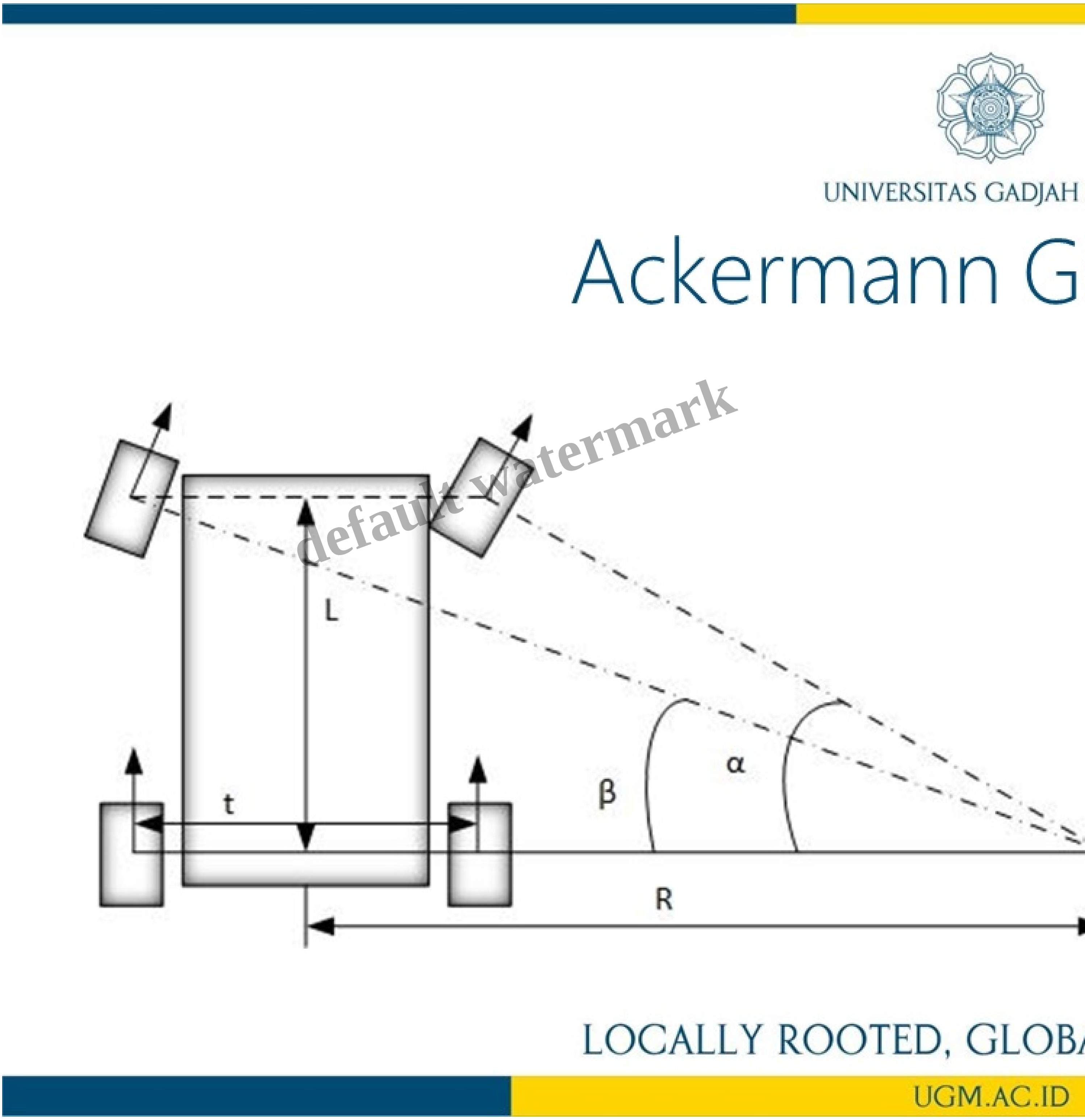

more detail visit our publication here.

# Kendali Logika Fuzzy pada Car Like Mobile Robot (CLMR) Penjejak Garis

**Abstrak**

Penelitian ini memaparkan perancangan sistem kendali logika fuzzy untuk mengatur kecepatan dan arah sudut steering pada car like mobile robot (CLMR) dengan menggunakan metode Ackermann steering. CLMR penjejak garis dirancang menggunakan 16 buah photodiode, dan terdapat 7 buah membership fuzzfikasi dari pembacaan error dan last error sehingga terbentuk 49 aturan. Untuk menguji perfoma kendali fuzzy pada sistem CLMR dalam mengikuti lintasan garis maka dilakukan pengujian dengan bentuk lintasan berupa garis lurus dan berbelok serta zig-zag dalam satu lintasan putar. Proses variasi nilai keanggotaan fuzzifikasi masukan dan defuzzifikasi keluaran dilakukan

sebanyak lima kali. Dari hasil pengujian diperoleh bahwa kendali logika fuzzy yang diaplikasikan pada sistem mampu membuat pergerakan CLMR sukses mengikuti lintasan uji selama 9,38 detik lebih baik 0,53 detik dari kendali PID. Selanjutnya, hasil rancangan sistem CLMR ini merupakan sebuah prototipe self-driving car.



# Contoh Penghitungan Acker

- L=3 meter
- t=1,2 meter
- R=2,5 meter

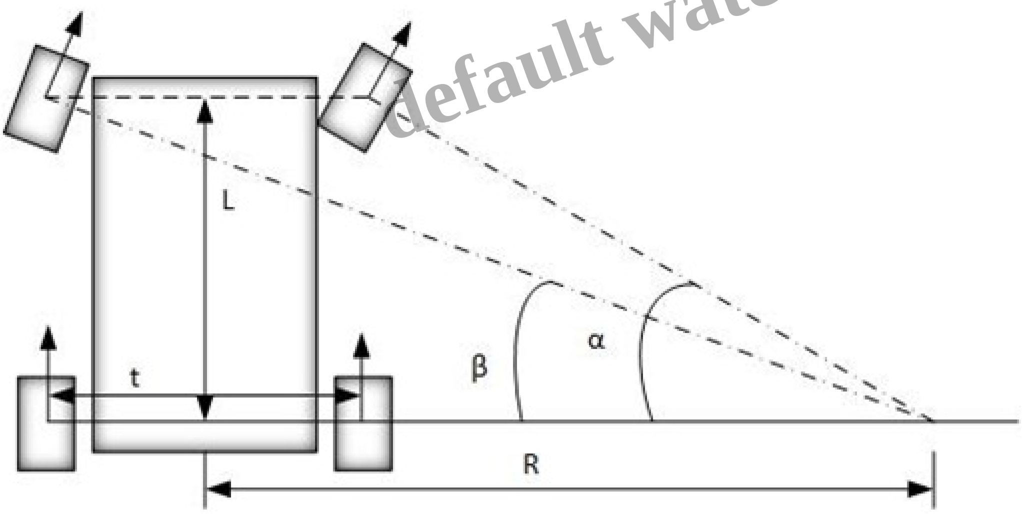
$$\alpha_{Right} = \tan^{-1}$$

$$\alpha_{Right} = \tan^{-1}$$

$$\alpha_{Right} =$$

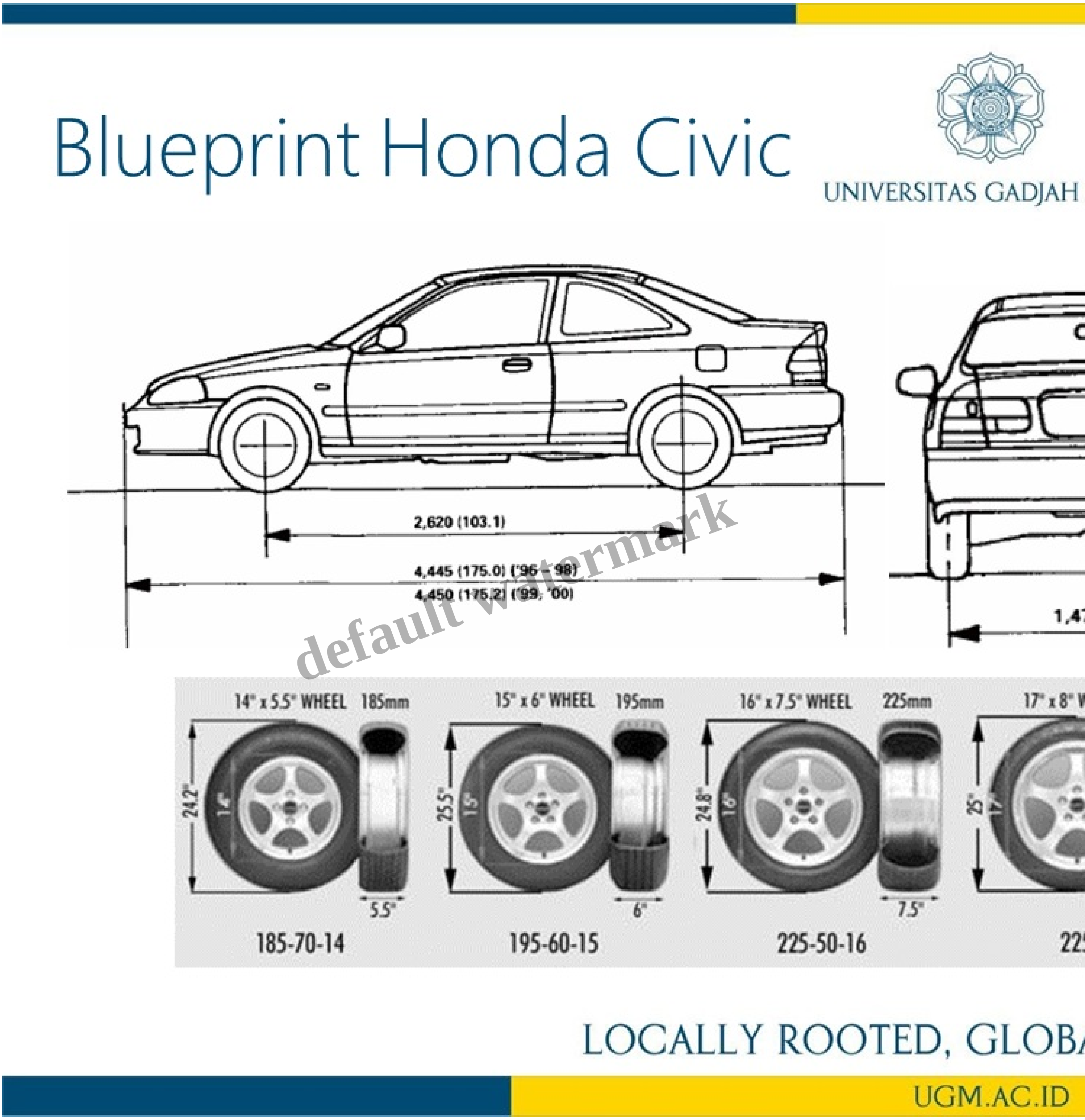
$$\alpha_{Right} =$$

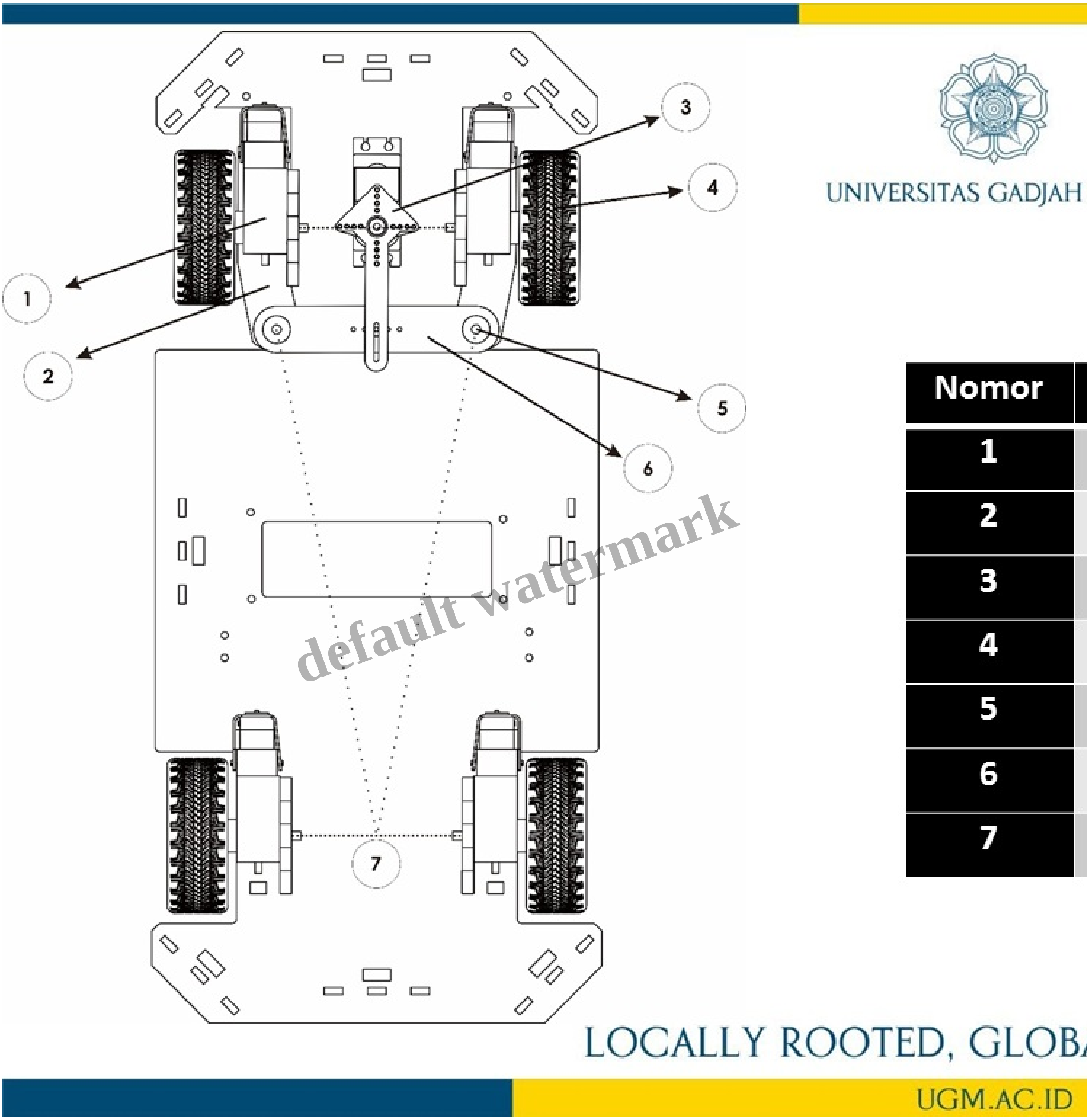




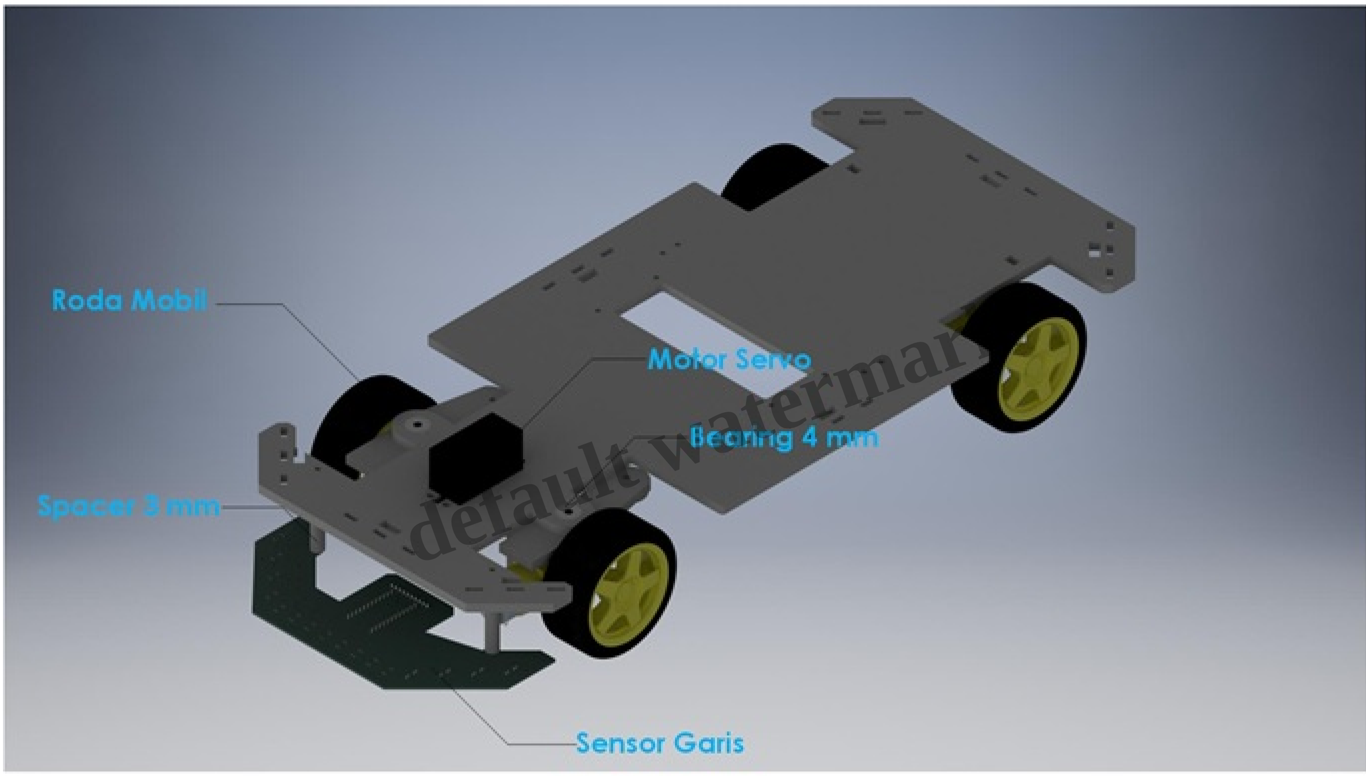

UGM.AC.ID

LOCA


Blueprint Honda Civic
UNIVERSITAS GADJAH
2,620 (103.1)
4,445 (175.0) ('96 – '98)
4,450 (175.2) ('99, '00)
14" x 5.5" WHEEL
185mm
24.2"
14"
5.5"
185-70-14
15" x 6" WHEEL
195mm
25.5"
15"
6"
195-60-15
16" x 7.5" WHEEL
225mm
24.8"
16"
7.5"
225-50-16
LOCALLY ROOTED,
UGM.AC.ID


1
2
3
4
5
6
7
UNIVERSITAS GADJAH
Nomor
1
2
3
4
5
6
7
LOCALLY ROOTED, GLOB
UGM.AC.ID

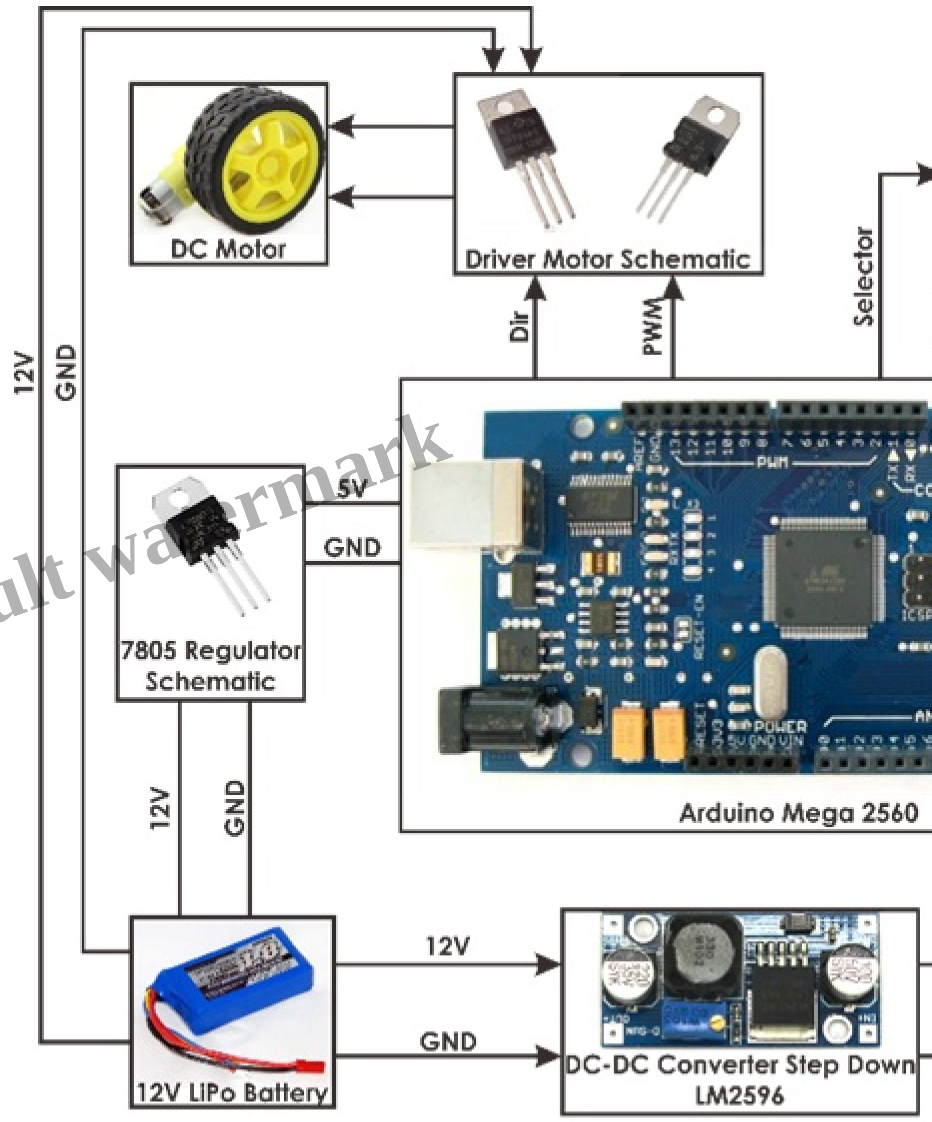
# Hasil Rancangan





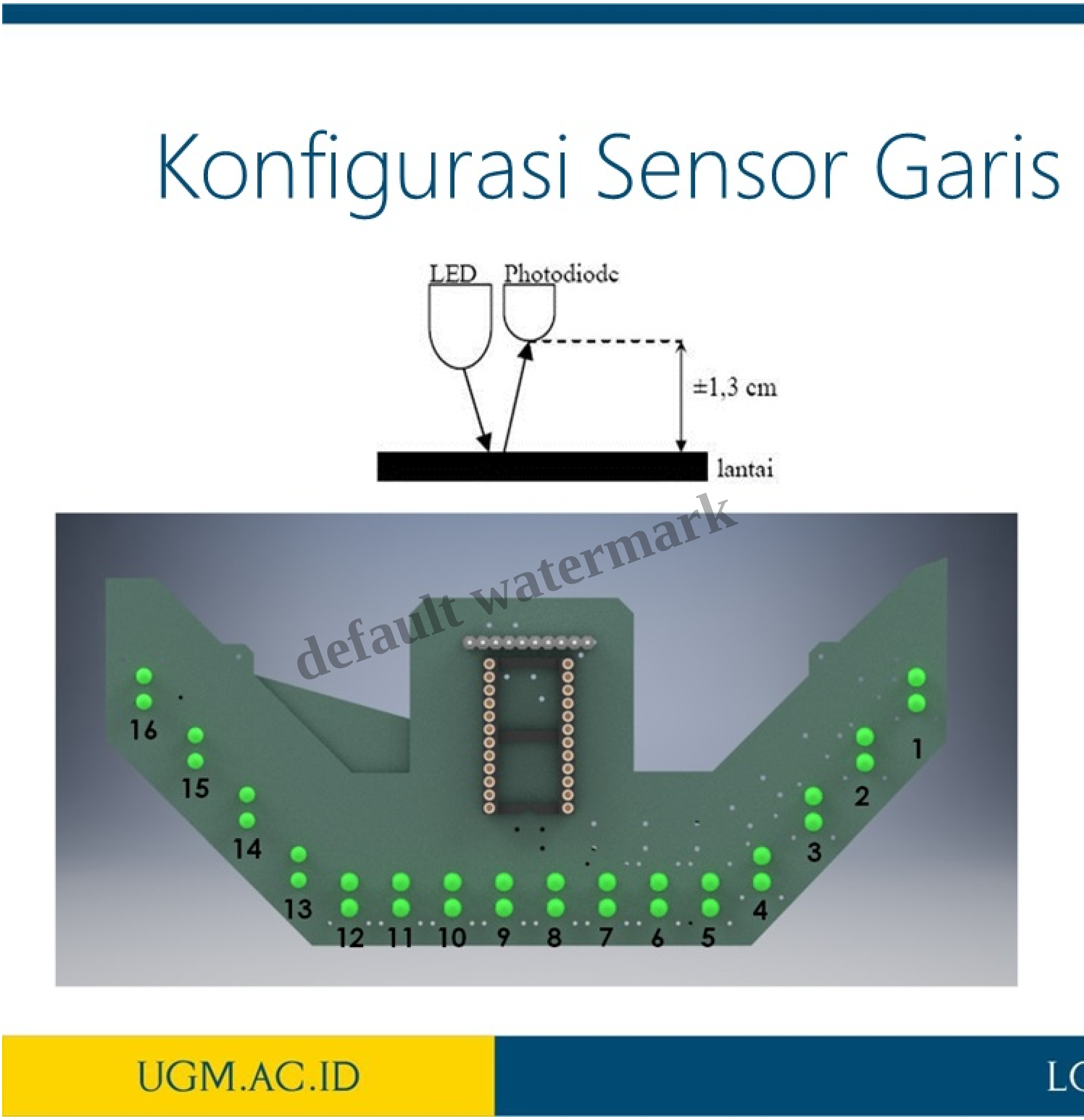

UGM.AC.ID

# Sistem Elektronika






UGM.AC.ID



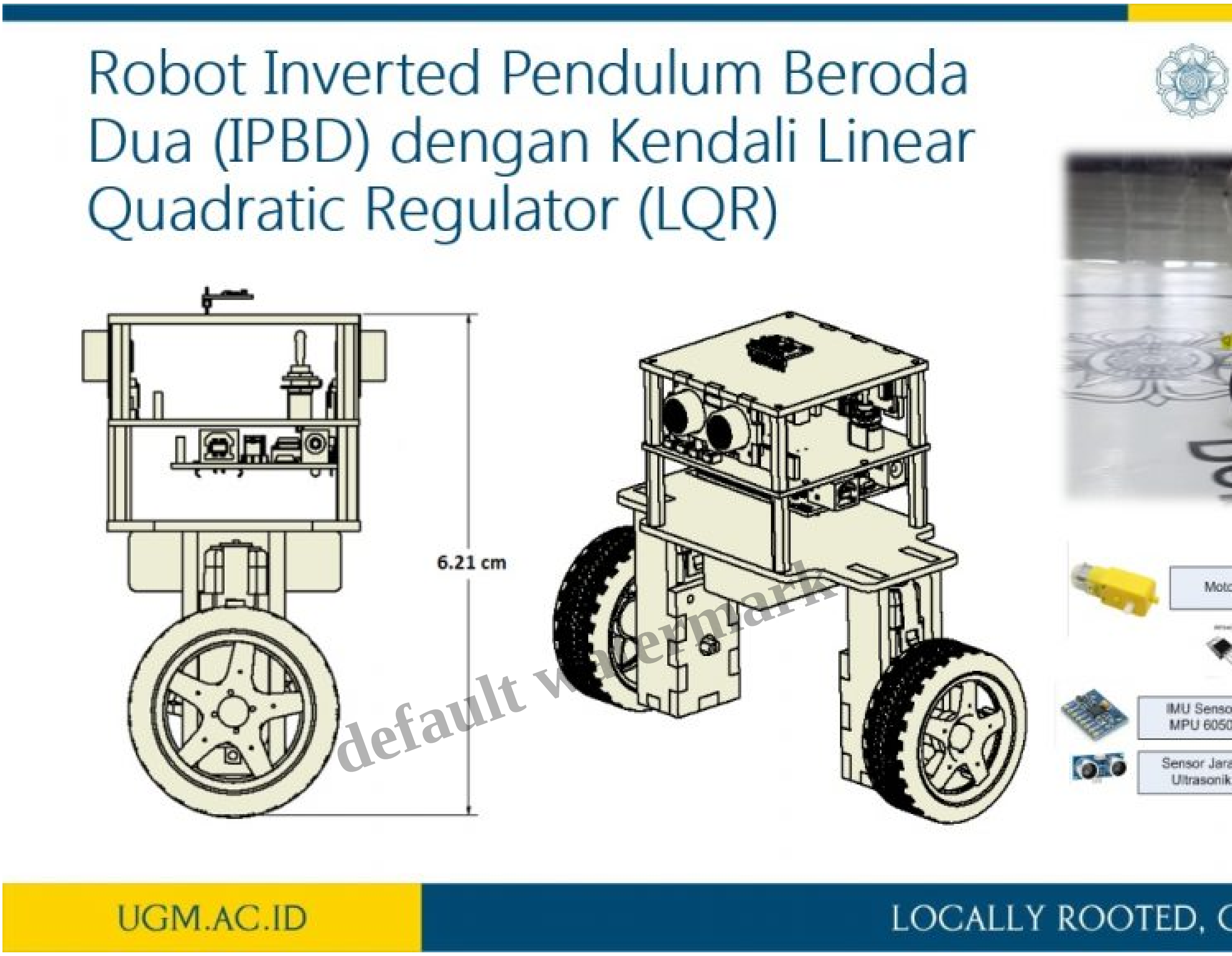



informasi lebih lanjut, silakan lihat publikasi kami di sini.

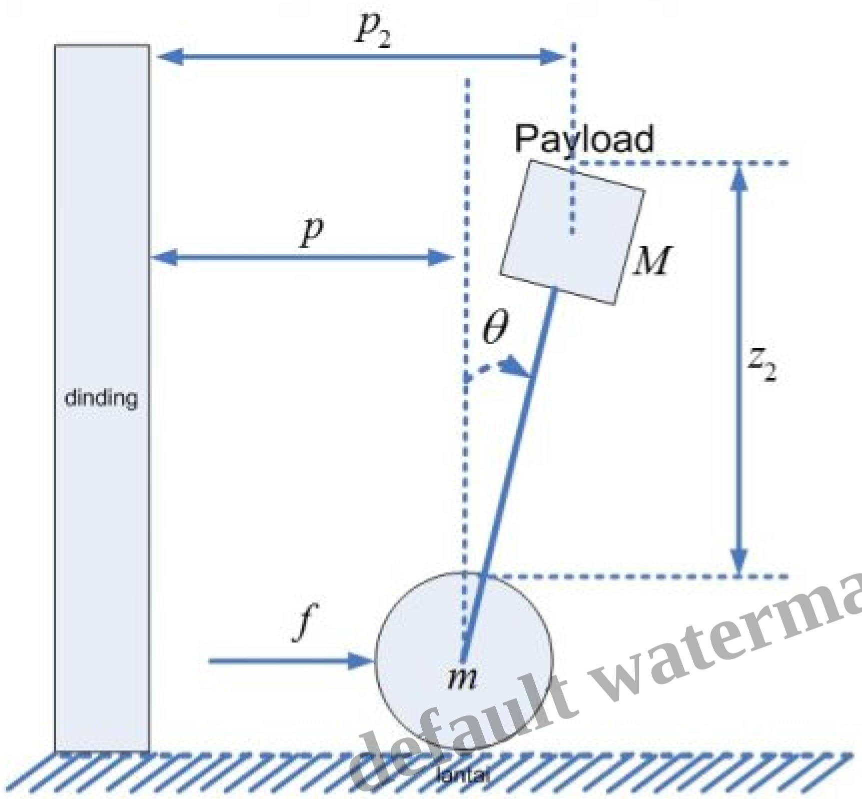
# Robot Inverted Pendulum Beroda Dua (IPBD) dengan Kendali *Linear Quadratic Regulator* (LQR)

**Abstrak**

Penelitian ini memaparkan proses pemodelan robot inverted pendulum beroda dua (IPBD) menggunakan dinamika Lagrange. Setelah sistem model robot IPBD diperoleh, teknik kendali optimal dalam hal ini menggunakan linear quadratic regulator (LQR) digunakan untuk melihat step respon sistem dan tanggapan respon sistem terhadap gangguan. Sebelum kendali LQR diimplementasikan, simulasi menggunakan Simulink Matlab dilakukan untuk mendapat parameter gain K pada kendali LQR. Selanjutnya, dengan mengubah-ubah matriks pembobot Q akan diperoleh variasi gain K. Pada penelitian ini dilakukan variasi matriks pembobotan Q sebanyak lima jenis. Sedangkan matriks elemen R di-tuning dengan nilai satu. Dari  hasil pengujian diperoleh bahwa dengan membesarkan pembobotan matriks Q, dihasilkan respon menuju keadaan steady lebih cepat dan overshoot berkurang. Parameter gain K dari hasil simulasi selanjutnya akan diimplementasikan secara embedded programming ke dalam Arduino Uno pada sistem robot IPBD.


Robot Inverted Pendulum Beroda Dua (IPBD) dengan Kendali Linear Quadratic Regulator (LQR)
6.21 cm
IMU Senso
MPU 6050
Sensor Jara
Ultrasonik
UGM.AC.ID
LOCALLY ROOTED,

# Model Matematis IPBD





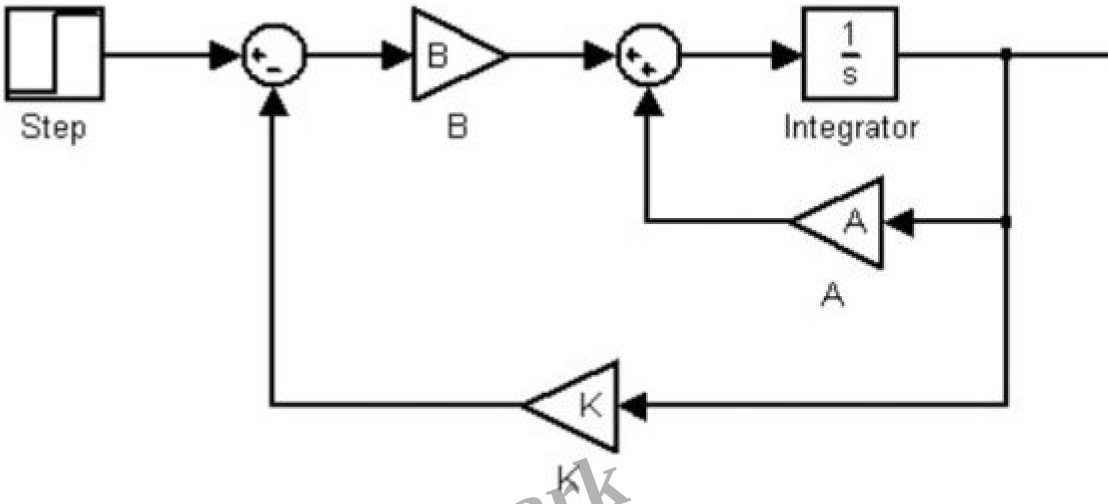

$$\dot{x} = Ax + Bu$$

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ \left(\frac{3m+2M}{3ml}\right)g & 0 \\ 0 & 0 \\ -\frac{2M}{3m}g & 0 \end{bmatrix}$$

UGM.AC.ID

LOCALLY ROOTED, G

# Linear Quadratic Regulator (LQR)

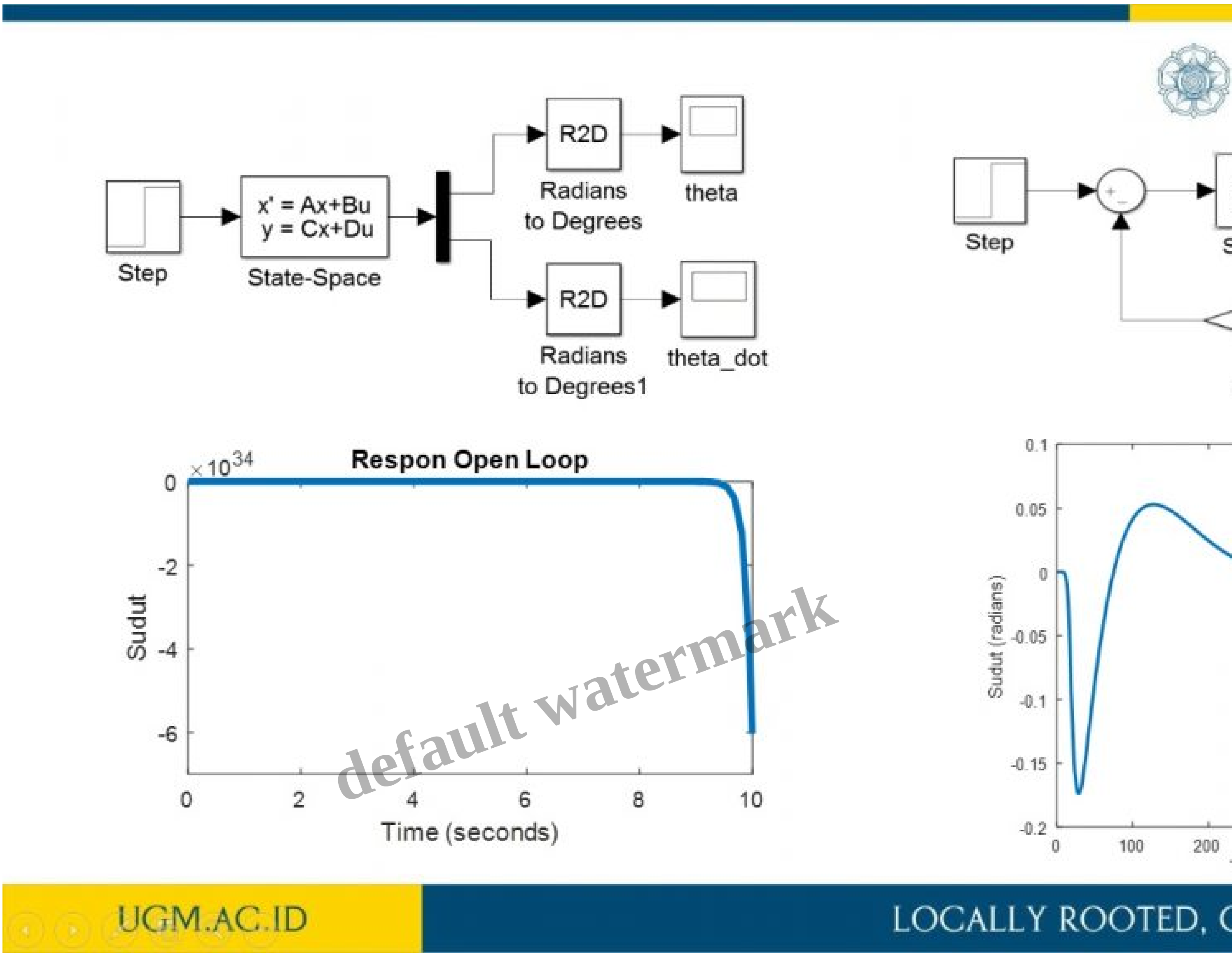





$$J = \frac{1}{2}\int_{t_0}^{T} \{x^T(t)Qx(t) + u^T(t)Ru(t)\}dt$$

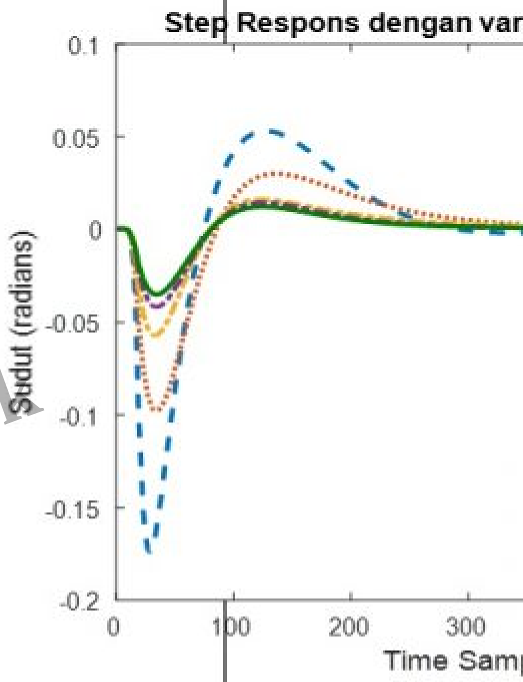
UGM.AC.ID

LOCALLY ROOTED,


Step
x' = Ax+Bu
y = Cx+Du
State-Space
R2D
Radians
to Degrees
theta
R2D
Radians
to Degrees1
theta_dot
Step
Respon Open Loop
Sudut
Time (seconds)
Sudut (radians)
UGM.AC.ID
LOCALLY ROOTED,

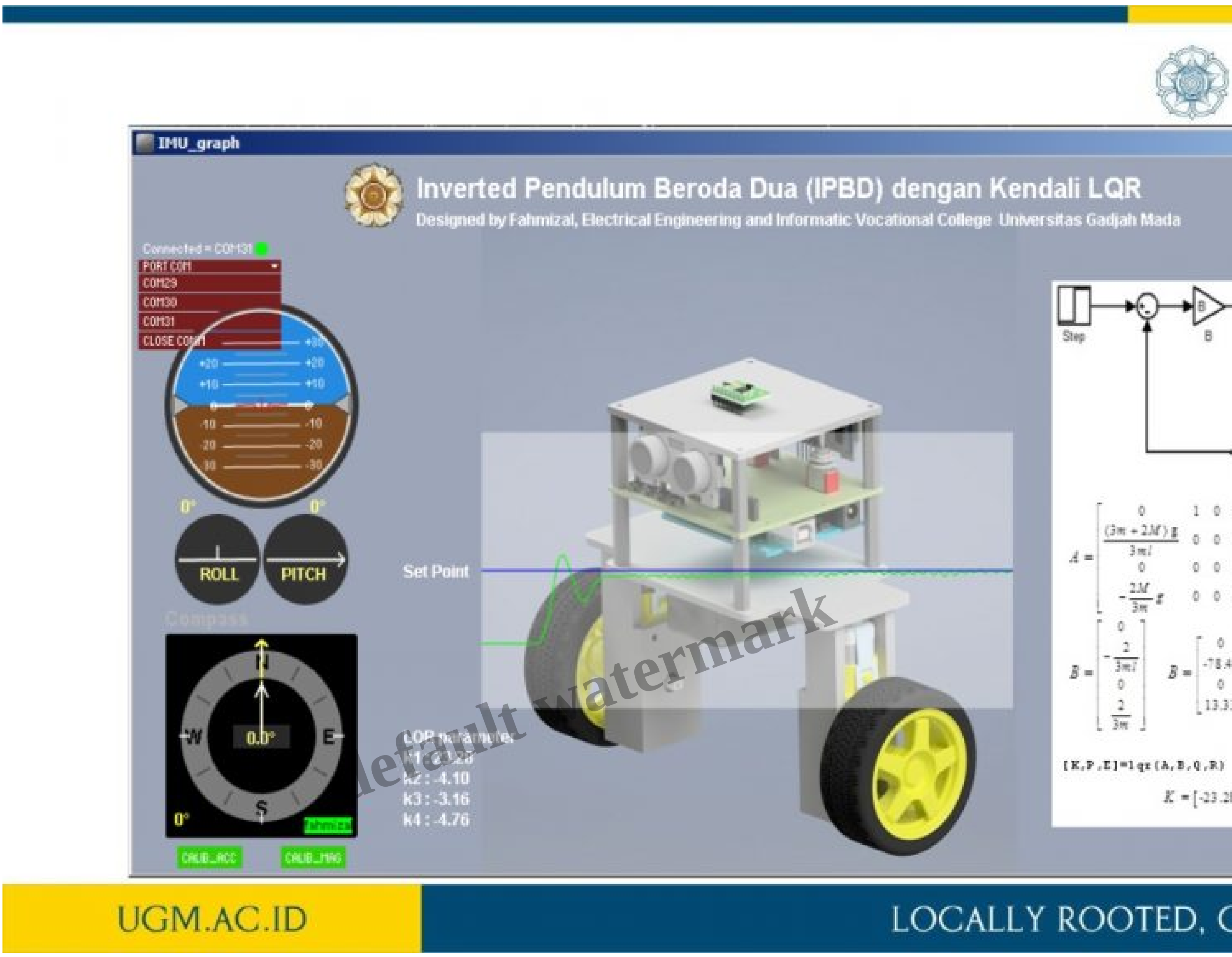
```
%parameter robot IPBD
m=0.05;
M=0.3;
l=0.17;
g=9.8;

%state space system robot IPBD
A =[0 1 0 0;((3*m)+(2*M))*g/(3*m*l) 0 0 0;0 0 0 1;-(2*M*g)/(3*m) 0 0
0;];
B =[0; -2/(3*m*l);0;2/(3*m)];
C= [1 0 0 0; 0 0 1 0];
D= [0;0];

%parameter LQR
Q=transpose(C)*C;
R=1;
[K,P,E]=lqr(A,B,Q,R)

K =
-8.6964   -0.6384   -1.0000   -1.0123
P =
2.3235    0.2038    0.6384    0.5464
0.2038    0.0234    0.0999    0.0900
0.6384    0.0999    1.0123    0.5124
0.5464    0.0900    0.5124    0.4536

E =
-17.1313 + 2.3109i
-17.1313 - 2.3109i
-1.1545 + 1.1133i
-1.1545 - 1.1133i
```






UGM.AC.ID

LOCALLY ROOTED,





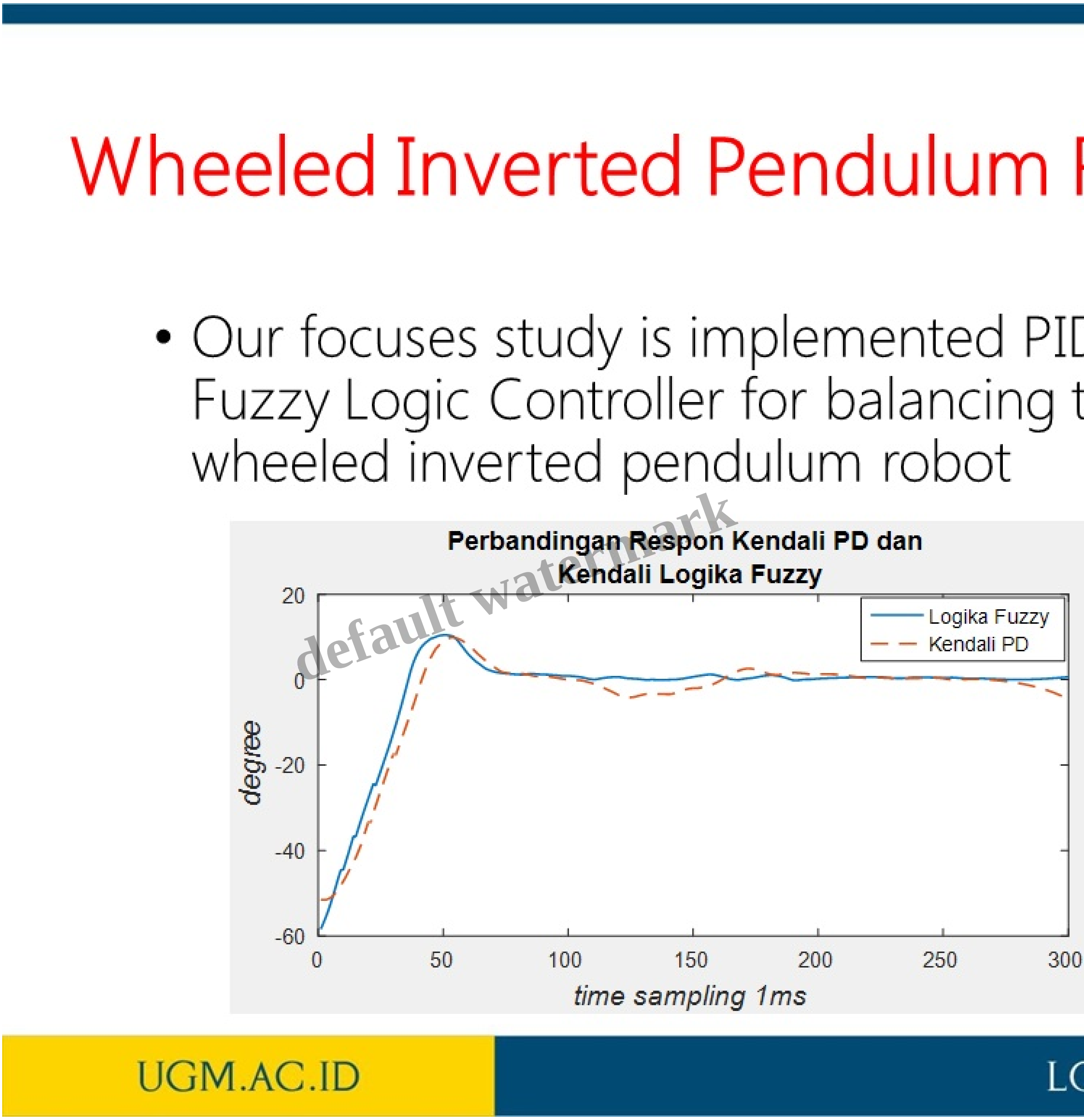
informasi lebih lanjut, silakan lihat publikasi kami di [sini](#).

# [Logika Fuzzy pada Robot Inverted Pendulum Beroda Dua (IPBD)](#)

**Abstrak**

Robot *inverted pendulum* beroda dua (IPBD) merupakan sistem yang tidak stabil dan bersifat *non-linear*. Motor DC sebagai penggerak robot yang terletak pada masing-masing roda kiri dan kanan memberikan variabel gaya untuk mempertahankan kestabilan robot. Oleh karena itu diperlukan suatu kendali yang dapat menjaga keseimbangan dari robot. Makalah ini memaparkan kendali logika *fuzzy* dalam hal pengendali keseimbangan robot. Pada perancangan robot ini, penulis menggunakan senor *inertia measurement unit* (IMU) versi MPU 6050 sebagai sensor pendeteksi keseimbangan robot. Nilai *setpoint* sudut robot yang diberikan adalah sudut elevasi robot terhadap sumbu horizontal atau pada

sumbu *pitch*. Selanjutnya, nilai keluaran sensor IMU dibandingkan dengan *setpoint*. Lebih lanjut, nilai kesalahan (*error*) dan nilai perubahan kesalahan (*delta errror*) yang dihasilkan akan digunakan sebagai masukan logika *fuzzy*. Hubungan relasi masukan *fuzzy* diselesaikan dengan aturan Mamdani. Keluaran dari logika *fuzzy* diselesaikan dengan perhitungan *weight average* (WA). Hasil keluaran logika *fuzzy* berupa nilai putaran motor kiri dan kanan yang dikendalikan dengan cara mengatur lebar pulsa sinyal *pulse with modulation* (PWM). Dari hasil pengujian diperoleh bahwa kendali logika *fuzzy* yang diaplikasikan pada robot IPBD dapat menjaga keseimbangan robot dengan osilasi pada sudut -2 hingga 2 derajat.






informasi lebih lanjut, silakan lihat publikasi kami di sini.

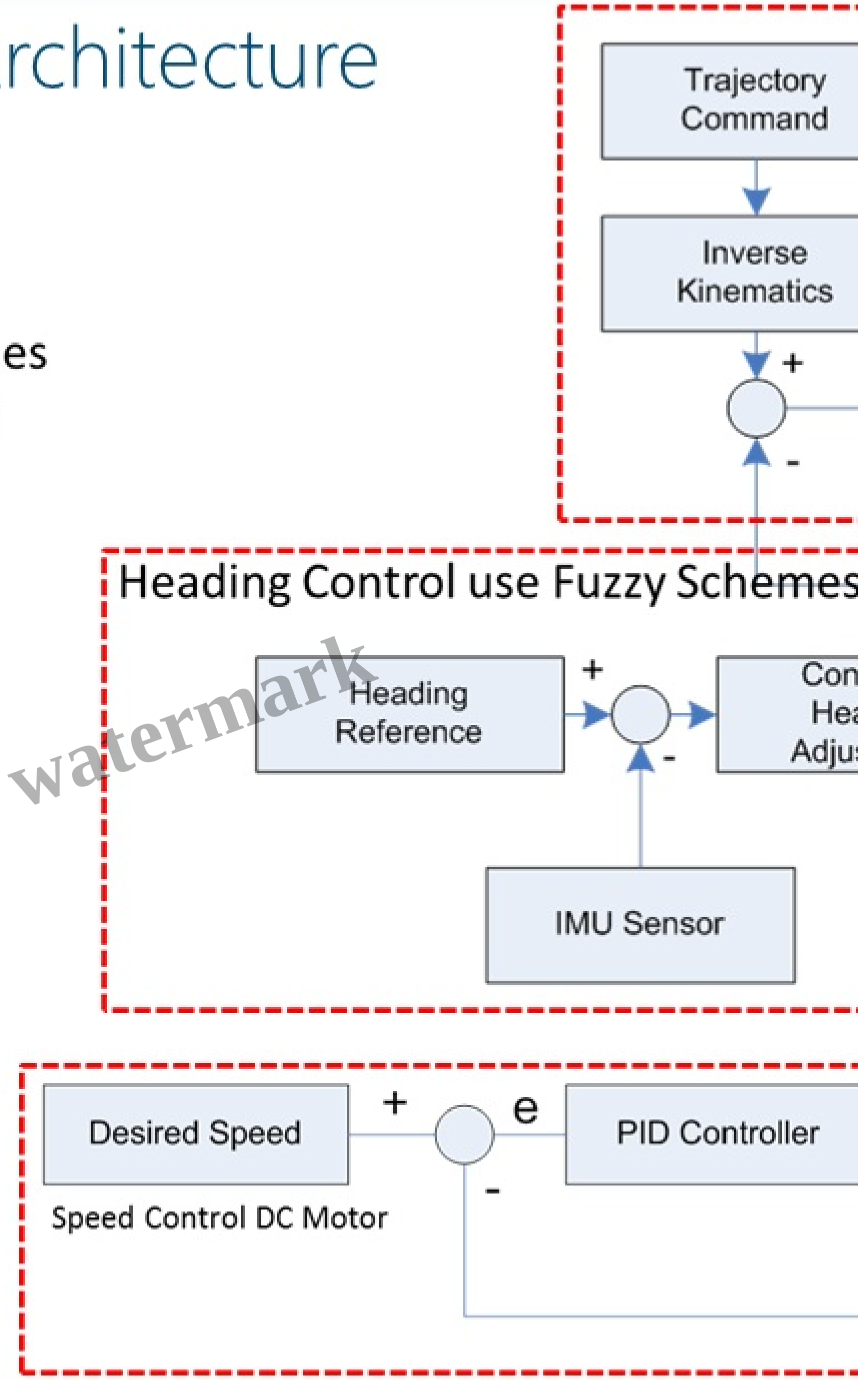
# Trajectory and Heading Tracking of a Mecanum Wheeled Robot using Fuzzy Logic Control

**Abstract**

This research presents trajectory tracking and heading adjustment systems of a Mecanum wheel robot using fuzzy logic controller (FLC) and inertia measurement unit (IMU) sensor. It is known that when a Mecanum wheel robot moves, its wheels may slip due to various floor conditions. During such a slip condition, the exact position and orientation of the robot often deviate from that which was intended. In order to reduce the size of this deviation, the Mecanum wheel was equipped with an IMU sensor which measures the orientation (heading) of the Mecanum wheel robot. The implementation of the IMU sensor as a heading sensor usually achieved using a complementary filterThe FLC, on the other hand, is used to force the Mecanum wheel robot to follow a given trajectory with minimum heading deviation. As a result, the FLC system and the IMU sensor provide a robust navigation scheme for Mecanum wheel robot without the need for external references such as beacons or visual markers. The implementation of our approach on the Mecanum wheel robot was achieved using embedded microcontroller for motion control and information acquisition. The developed FLC system was also deployed with a friendly graphical user interface which allows for the adjustment of fuzzification methods.

# Overall System Architecture

1. IMU Fusion for Heading
2. Kinematics Control
3. Trajectory and Heading Control use Fuzzy Schemes
4. Speed Control DC Motor






UGM.AC.ID

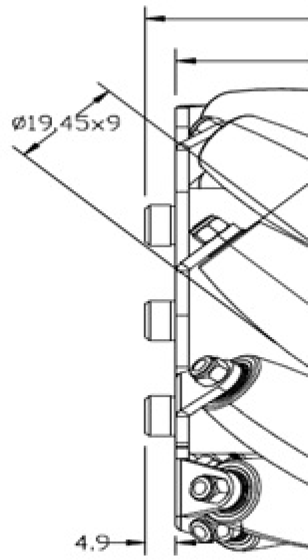
# Kinematic Control Mecanum Whee

**Mecanum wheel :**

- compact design
- high load capacity
- discontinuous wheel contact
- high sensitivity to floor irregularities
- complex wheel design

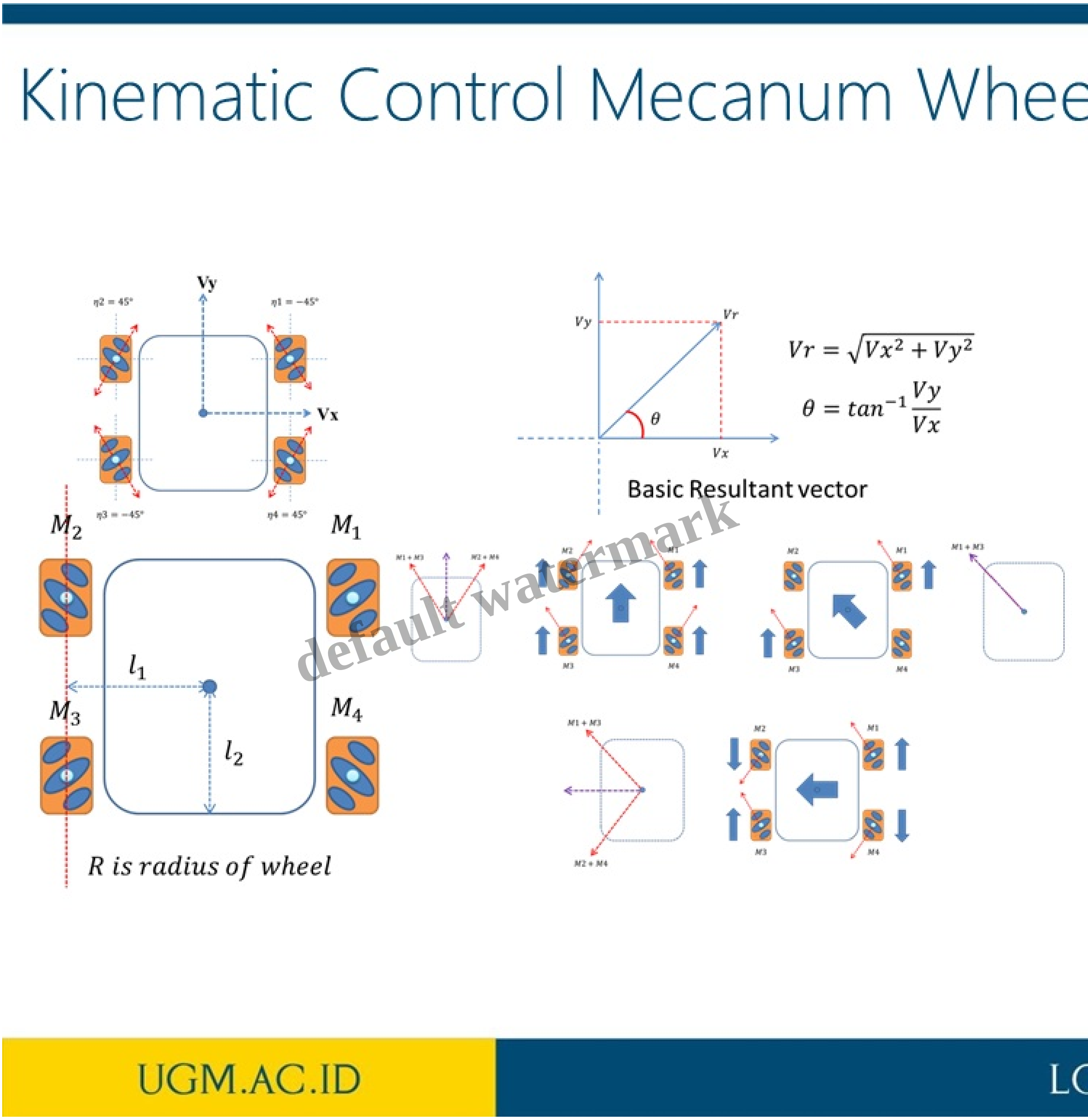


**One** of tl
designs i
in 1973 b
**Swedish**




UGM.AC.ID

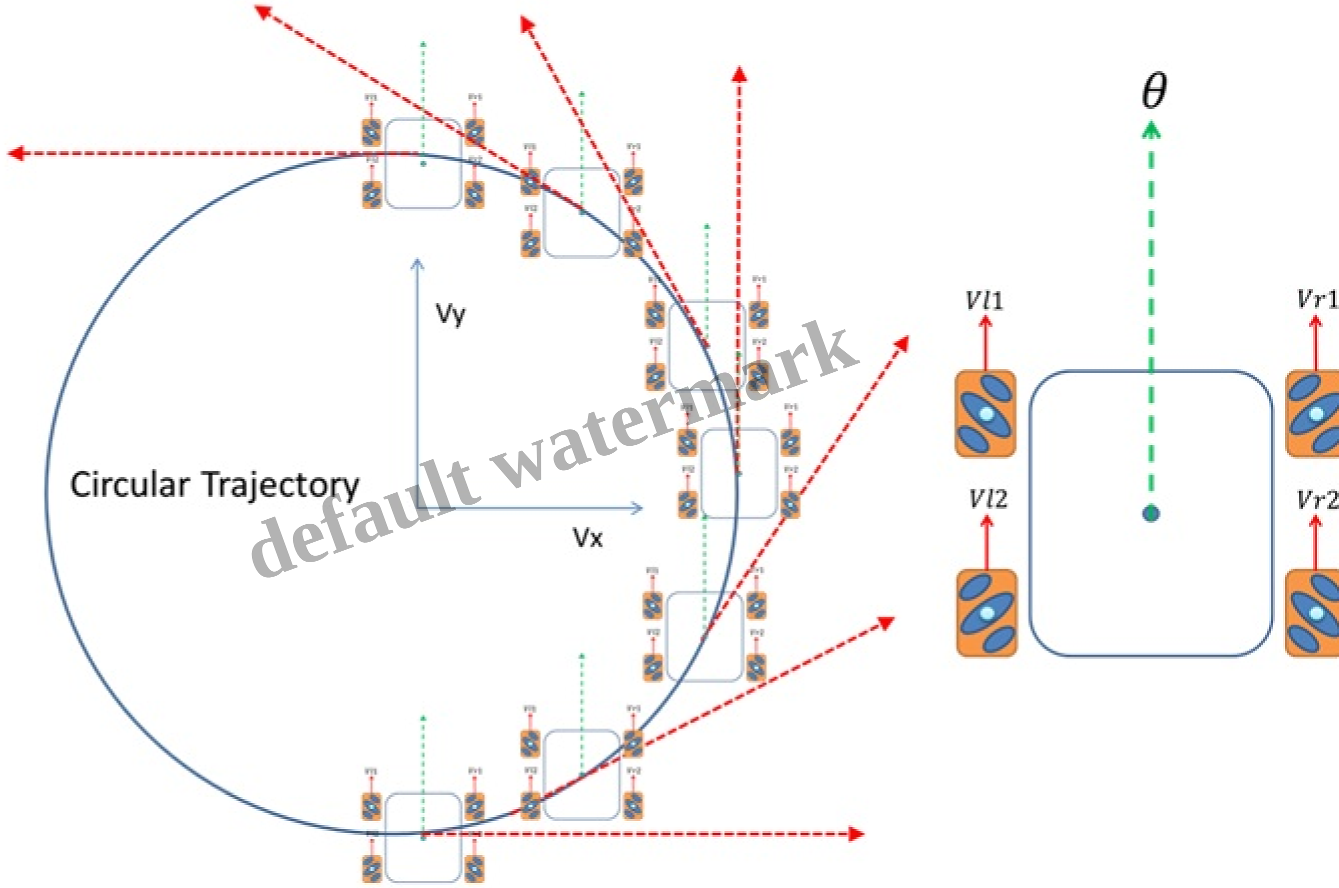
LC


Kinematic Control Mecanum Whee
Vy
Vx
η2 = 45°
η1 = −45°
η3 = −45°
η4 = 45°
M2
M1
M3
M4
l1
l2
R is radius of wheel
Vy
Vr
θ
Vx
Vr = √(Vx² + Vy²)
θ = tan⁻¹ Vy/Vx
Basic Resultant vector
M1 + M3
M2 + M4
UGM.AC.ID
LC

# Trajectory and Heading Control use Fuzzy Sch

(Determine heading error)

$error = \theta \ - ($




Ideal Trajectory and Heading without Slip

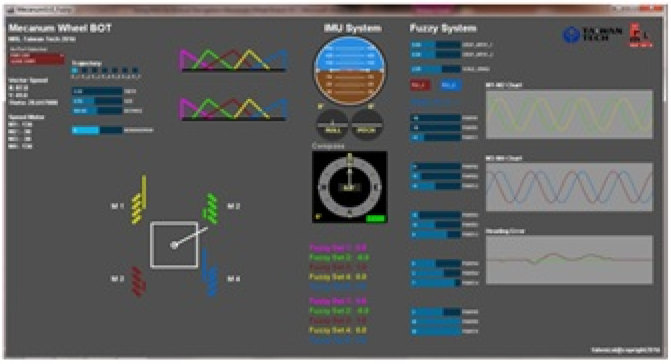


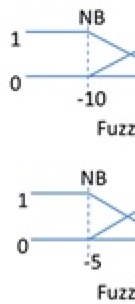
UGM.AC.ID LC

# Trajectory and Heading Control use Fuz

## (Embedded Fuzzy Control System)

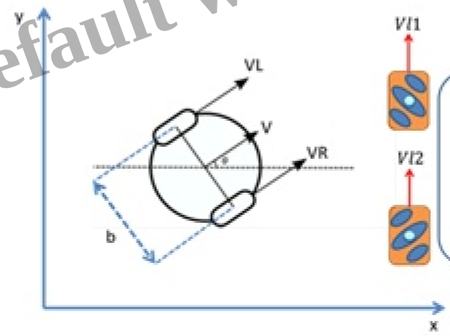
Fuzzy

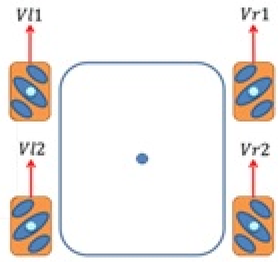


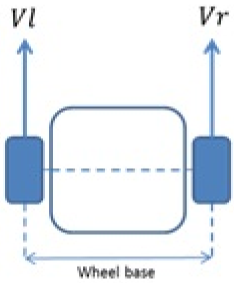



Differential Steering Mobile Robot
for adjusting Heading of mobile robot





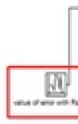



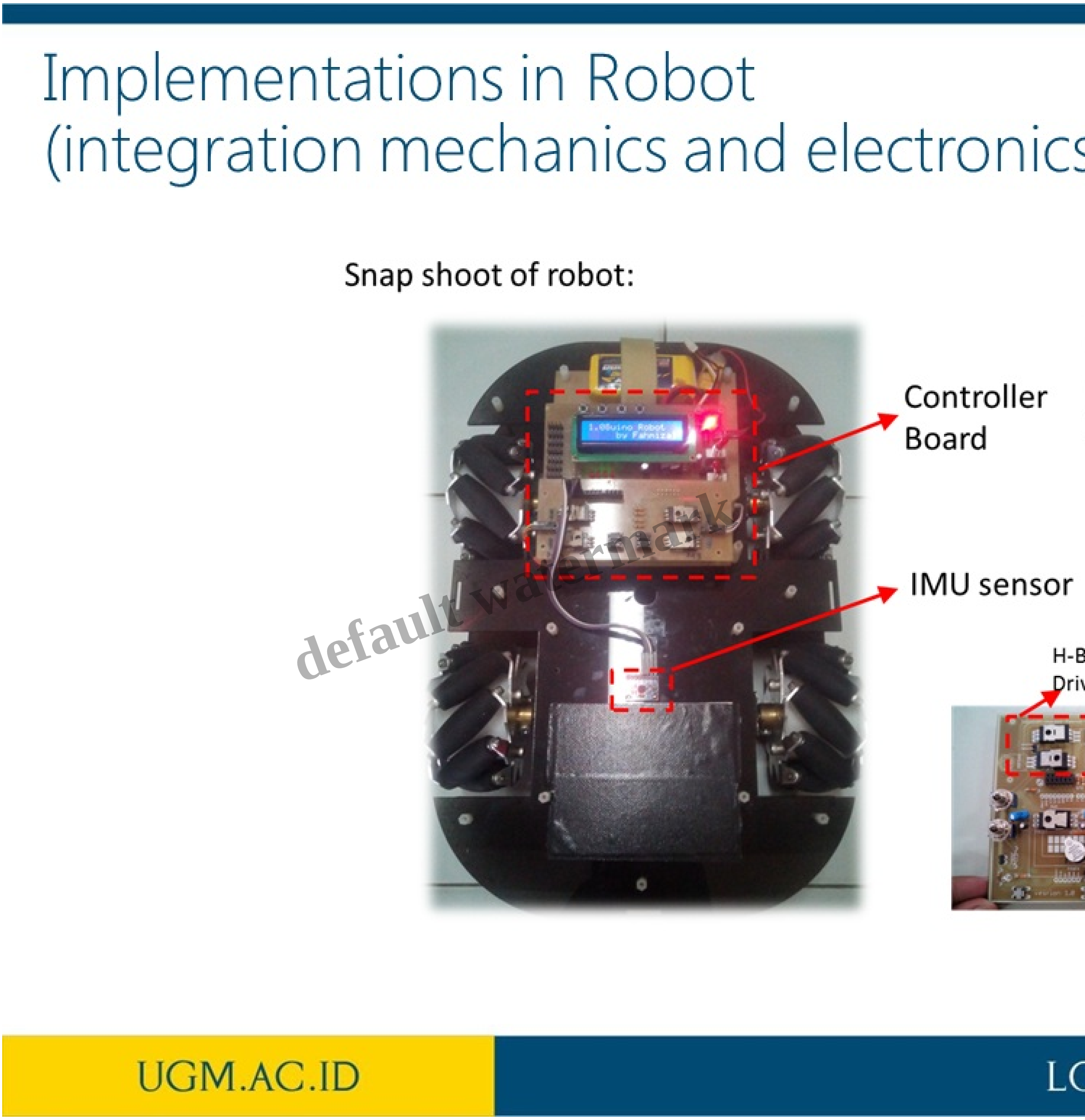

$$V = \frac{v_R + v_L}{2} = \frac{\omega_R + \omega_L}{2} r$$

$$\omega = \frac{v_R - v_L}{b} = \frac{\omega_R - \omega_L}{b} r$$

$$\int \omega = \theta$$



UGM.AC.ID LC






more detail visit our publication [here]

**Date Created**
October 9, 2018
**Author**
fahmizal