

Laporan chapter 2

Nama : Mochammad Qussay Alhindi Achmadi

NIM : 1103213087

1. Pendahuluan

Di bab kedua ini, kita akan membahas langkah-langkah untuk memulai pemrograman dengan Robot Operating System (ROS). Setelah memahami konsep dasar ROS di bab sebelumnya, kita akan fokus pada pengaturan lingkungan pengembangan, pembuatan paket ROS pertama, serta penulisan dan pelaksanaan node sederhana. Kita juga akan melihat bagaimana cara menjalankan publisher dan subscriber menggunakan perintah **roslaunch**.

2. Mengatur Lingkungan Pengembangan

2.1. Instalasi ROS

Sebelum memulai pemrograman, pastikan Anda telah menginstal ROS di komputer Anda. ROS umumnya diinstal pada sistem operasi Ubuntu. Berikut adalah langkah-langkah umum untuk instalasi:

1. **Perbarui Sistem:** Jalankan perintah untuk memperbarui sistem Anda.

```
bash Verify Open In Editor ▶ 📄  
1 sudo apt update  
2 sudo apt upgrade
```

2. **Instal ROS:** Pilih versi ROS yang sesuai dan ikuti petunjuk instalasi di Wiki ROS.
3. **Inisialisasi ROS:** Setelah instalasi, inisialisasi ROS dengan menjalankan perintah berikut:

```
bash Verify Open In Editor ▶ 📄  
1 source /opt/ros/<distro>/setup.bash
```

2.2. Mengatur Workspace

Workspace adalah direktori tempat Anda akan mengembangkan paket ROS. Untuk membuat workspace, ikuti langkah-langkah berikut:

1. **Buat Direktori Workspace:**

```
bash Verify Open In Editor ▶ 📄  
  
1 mkdir -p ~/catkin_ws/src  
2 cd ~/catkin_ws/
```

2. **Inisialisasi Workspace:**

```
bash Verify Open In Editor ▶ 📄  
  
1 catkin_make
```

3. **Sumber Workspace:**

```
bash Verify Open In Editor ▶ 📄  
  
1 source devel/setup.bash
```

3. Membuat Paket ROS Pertama

Setelah lingkungan pengembangan siap, langkah selanjutnya adalah membuat paket ROS pertama Anda.

3.1. Membuat Paket

1. **Navigasi ke Direktori src:**

```
bash Verify Open In Editor ▶ 📄  
  
1 cd ~/catkin_ws/src
```

2. Buat Paket Baru:

```
bash Verify Open In Editor ▶ 📄  
  
1 catkin_create_pkg matering_ros_demo_pkg std_msgs rospy roscpp
```

3.2. Struktur Paket

Setelah paket dibuat, Anda akan melihat struktur direktori berikut:

```
Verify Open In Editor ▶ 📄  
  
1 matering_ros_demo_pkg/  
2 |— CMakeLists.txt  
3 |— package.xml  
4 |— src/
```

- **CMakeLists.txt:** File ini digunakan untuk mengonfigurasi proses build.
- **package.xml:** File ini berisi informasi tentang paket, termasuk dependensi.
- **src/:** Direktori ini adalah tempat Anda akan menulis kode sumber.

4. Menulis dan Menjalankan Node Publisher dan Subscriber

4.1. Menulis Node Publisher

1. **Buat File Node Publisher:** Navigasi ke direktori **src** dalam paket Anda dan buat file Python baru:

```
bash Verify Open In Editor ▶ 📄  
  
1 cd ~/catkin_ws/src/matering_ros_demo_pkg/src  
2 touch demo_topic_publisher.py  
3 chmod +x demo_topic_publisher.py
```

2. **Tulis Kode Node Publisher:** Buka `demo_topic_publisher.py` dan tambahkan kode berikut:

```
python Verify Open In Editor ▶ 📄

1  #!/usr/bin/env python
2  import rospy
3  from std_msgs.msg import String
4
5  def publisher():
6      rospy.init_node('demo_topic_publisher', anonymous=True)
7      pub = rospy.Publisher('demo_topic', String, queue_size=10)
8      rate = rospy.Rate(1)  # 1 Hz
9      while not rospy.is_shutdown():
10         message = "Halo dari Publisher! Waktu: %s" % rospy.get_time()
11         rospy.loginfo(message)
12         pub.publish(message)
13         rate.sleep()
14
15  if __name__ == '__main__':
16      try:
17         publisher()
18     except rospy.ROSInterruptException:
19         pass
```

4.2. Menulis Node Subscriber

1. **Buat File Node Subscriber:** Buat file Python baru untuk subscriber:

```
bash Verify Open In Editor ▶ 📄

1  touch demo_topic_subscriber.py
2  chmod +x demo_topic_subscriber.py
```

2. **Tulis Kode Node Subscriber:** Buka `demo_topic_subscriber.py` dan tambahkan kode berikut:

```
python Verify Open In Editor ▶ 📄  
  
1 #!/usr/bin/env python  
2 import rospy  
3 from std_msgs.msg import String  
4  
5 def callback(data):  
6     rospy.loginfo("Pesan diterima: %s", data.data)  
7  
8 def subscriber():  
9     rospy.init_node('demo_topic_subscriber', anonymous=True)  
10    rospy.Subscriber('demo_topic', String, callback)
```

`rospy.spin()`

`if name == 'main': try: subscriber() except rospy.ROSInterruptException: pass`

```
Verify Open In Editor ▶ 📄  
  
1  
2 ### 4.3. Menjalankan Node Publisher dan Subscriber  
3  
4 1. **Kompilasi Paket**:  
5 Kembali ke direktori root workspace dan kompilasi paket:  
6 ```bash  
7 cd ~/catkin_ws  
8 catkin_make
```

2. **Jalankan ROS Master:** Buka terminal baru dan jalankan:

```
bash Verify Open In Editor ▶ 📄  
  
1 roscore
```

3. **Jalankan Node Publisher:** Buka terminal baru dan jalankan publisher Anda:

```
bash Verify Open In Editor ▶ 📄  
  
1 rosrun matering_ros_demo_pkg demo_topic_publisher
```

4. **Jalankan Node Subscriber:** Buka terminal baru dan jalankan subscriber Anda:

```
bash Verify Open In Editor ▶ 📄  
  
1 rosrun matering_ros_demo_pkg demo_topic_subscriber
```

5. Kesimpulan

Di bab ini, kita telah membahas langkah-langkah awal untuk memulai pemrograman dengan ROS. Kita telah mengatur lingkungan pengembangan, membuat paket ROS pertama, serta menulis dan menjalankan node publisher dan subscriber. Dengan pemahaman ini, Anda dapat mulai mengeksplorasi lebih dalam tentang pemrograman dengan ROS dan mengembangkan aplikasi robotika yang lebih kompleks.