

Dokumentasi Pengerjaan Tugas 1 VTOL Magang Bayucaraka 2024

Muhammad Zia Alhambra | 5024231059

Saya mengawali tugas VTOL dengan membuat direktori untuk menampung fork github penugasan serta node CPP yang saya buat.

```
mkdir -p px4_ws/src
```

```
cd px4_ws/src
```

Setelah membuat workspace penugasan, saya lanjut fork github penugasan VTOL. Perlu diketahui di dalam repo github penugasan sudah ada folder px4_msgs yang penting untuk menjalankan node pada saat simulasi. Setelah fork berhasil, saya masuk kedalam folder tersebut dan membuat package ros2.

Ros2 pkg create --build-type ament_cmake tugasLintasan

Pada file package.xml hanya diperlukan menambahkan beberapa dependencies agar node dapat jalan.

```
<buildtool_depend>ament_cmake</buildtool_depend>
<buildtool_depend>eigen3_cmake_module</buildtool_depend>

<build_depend>eigen</build_depend>
<build_depend>ros_environment</build_depend>

<depend>builtin_interfaces</depend>
<depend>rclcpp</depend>

<depend>px4_msgs</depend>
<depend>geometry_msgs</depend>
<depend>sensor_msgs</depend>

<depend>launch</depend>
<depend>launch_testing</depend>
<depend>launch_testing_ros</depend>

<exec_depend>roslaunch</exec_depend>
<exec_depend>roslaunch</exec_depend>

<buildtool_export_depend>eigen3_cmake_module</buildtool_export_depend>
<build_export_depend>eigen</build_export_depend>
<test_depend>ament_lint_auto</test_depend>
<test_depend>ament_lint_common</test_depend>
```

Pada file CMakeLists perlu saya tambahkan beberapa dependencies agar node dapat dijalankan.

```
find_package(ament_cmake REQUIRED)
find_package(builtin_interfaces REQUIRED)
find_package(eigen3_cmake_module REQUIRED)
find_package(Eigen3 REQUIRED)
find_package(geometry_msgs REQUIRED)
find_package(px4_msgs REQUIRED)
find_package(rclcpp REQUIRED)
find_package(sensor_msgs REQUIRED)
```

Selain dependencies, juga perlu menambahkan executable dan install ke node CPP tugas 1. Untuk node CPP tugas 1 saya namakan “tugas1.cpp” serta mengganti nama executable menjadi “bulat”

```
add_executable(bulat src/tugas1.cpp)
ament_target_dependencies(bulat rclcpp px4_msgs)
install(TARGETS bulat DESTINATION lib/${PROJECT_NAME})
```

Dalam node CPP, saya menambahkan float “droneRad” untuk yaw drone serta “pathRad” untuk pergerakan drone pada bidang x dan y. Dalam function public, diberi if statement bahwa saat offboardSetpointCounter lebih dari 40 maka droneRad akan berkurang sebesar 0.157 dan pathRad bertambah sebesar 0.157 setiap 50hz.

Lalu setelah droneRad kurang dari -6.28 maka code akan mengeksekusi VEHICLE_CMD_NAV_LAND setelah offboardSetpointCounter telah mencapai 380 dan drone akan disarm setelah mencapai 400.

```
if (offboardSetpointCounter_ % 5 == 0 && offboardSetpointCounter_ > 40){
    droneRad -= 0.157;
    pathRad += 0.157;
}
if (droneRad <= -6.28){
    if (offboardSetpointCounter_ < 380){
        this->publishVehicleCommand(VehicleCommand::VEHICLE_CMD_NAV_LAND, 1, 0);
    }
    if (offboardSetpointCounter_ == 400){
        this->disarm();
    }
}
if (offboardSetpointCounter_ <= 400){
    offboardSetpointCounter_++;
}
```

Pada function void publishTrajectorySetpoint, untuk msg.position saya masukkan “3*(float)cos(pathRad)” pada bidang x dan “3*(float)sin(pathRad)” pada bidang y sesuai rumus sin dan cos. Tetapi drone akan bergerak searah jarum jam jika rumus bidang y positif, maka saya jadikan rumus tersebut untuk dikali -3. Untuk msg.yaw, droneRad ditambah 1.57 karena heading drone akan diawali kearah bawah (-y).

```
void MissionControl::publishTrajectorySetpoint(){
    TrajectorySetpoint msg{};
    msg.position = {3*(float)cos(pathRad), -3*(float)sin(pathRad), -5.0};
    msg.yaw = droneRad + 1.57;
    msg.timestamp = this->get_clock()->now().nanoseconds() / 1000;
    trajectorySetpointPublisher_->publish(msg);
}
```

Setelah semua siap, saya build package tugasLintasan dan px4_msgs. Lalu saya jalankan simulasi dengan menjalankan agen, lalu buka gazebo pada terminal lain, dan akhirnya jalankan node dengan “ros2 run tugasLintasan bulat”

Video:

https://drive.google.com/file/d/1qPONjM1CHhkRziqOPULa8hVznS_3zawp/view?usp=sharing