Laporan dokumentasi pengerjaan tugas 3 px4 Bayucaraka 2024
Farrel Ganendra | 5024231036

Pada Package yang dibuat sebelumnya, buat file baru bernama “kotak.cpp”, lalu tambahkan perintah membuat executable, ament dependencies, dan install pada file CMakeLists.txt

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

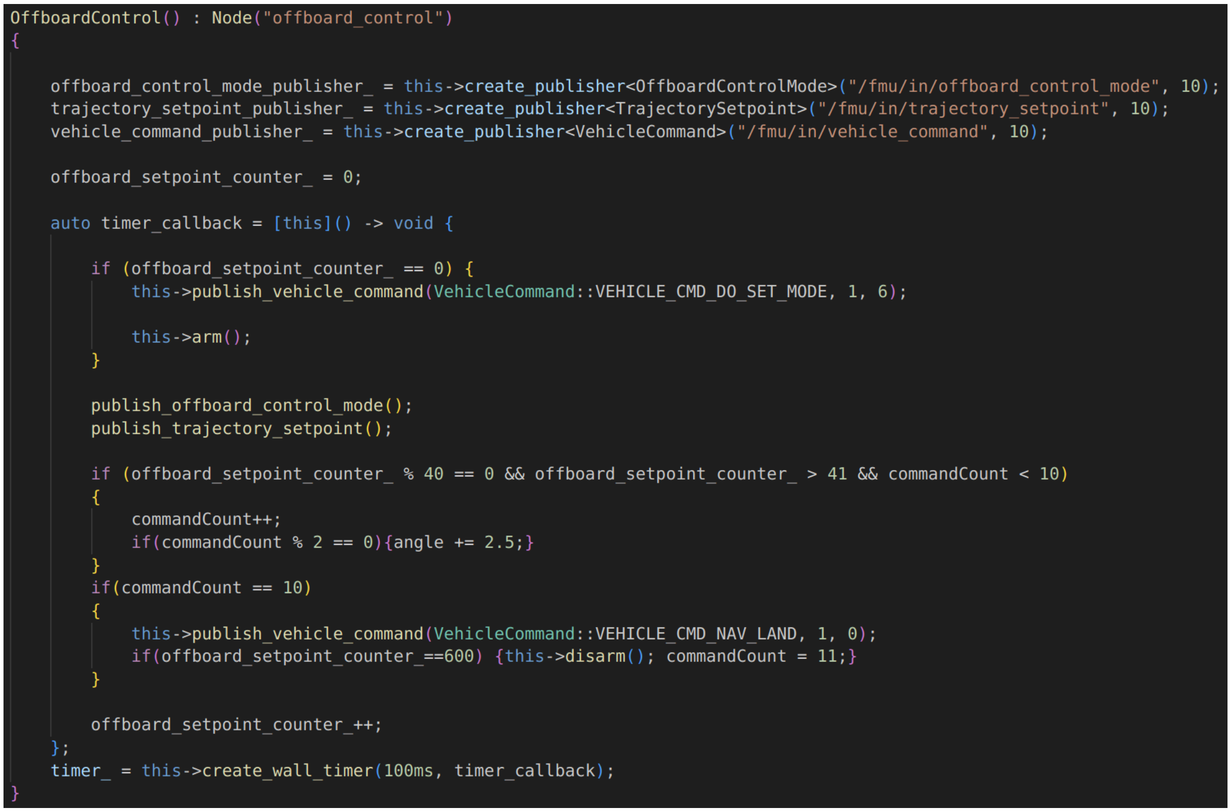
Lalu copy isi file kotak.cpp ke pentagram.cpp dan hapus array rotasi z, modifikasi dua array posisi x dan y ke bentuk pentagram. Lalu tambahkan variabel float global “angle” untuk mengatur angle dari drone. Berikut posisi posisi yang dibutuhkan.

```
int commandCount = 0;
float posX[10] = {0.0, 0.9, 0.9, -1.4, -1.4, 1.4, 1.4, -0.9, -0.9, 0.0};
float posY[10] = {0.0, 2.7, 2.7, 1.00, 1.00, 1.0, 1.0, 2.70, 2.70, 0.0};
float angle = 0.0;
```

Pada metode publish_trajectory_setpoint(), ubah msg.yaw menjadi 1.25 + angle. 1.25 adalah sudut awal yang kita inginkan.

```
void OffboardControl::publish_trajectory_setpoint()
{
   TrajectorySetpoint msg{};
   msg.position = {posX[commandCount], posY[commandCount], -5.0};
   msg.yaw = 1.25 + angle;
   msg.timestamp = this->get_clock()->now().nanoseconds() / 1000;
   trajectory_setpoint_publisher_->publish(msg);
}
```

Lalu pada constructor kelas OffboardControl, selain menambahkan commandCount setiap 4 detik, tambahkan juga angle sebesar 2.5 radian apabila commandCount genap. Kemudian landing ketika commandCount di 10 dan disarm setelah node berjalan selama 1 menit.



```
OffboardControl() : Node("offboard_control")
{

    offboard_control_mode_publisher_ = this->create_publisher<OffboardControlMode>("/fmu/in/offboard_control_mode", 10);
    trajectory_setpoint_publisher_ = this->create_publisher<TrajectorySetpoint>("/fmu/in/trajectory_setpoint", 10);
    vehicle_command_publisher_ = this->create_publisher<VehicleCommand>("/fmu/in/vehicle_command", 10);

    offboard_setpoint_counter_ = 0;

    auto timer_callback = [this]() -> void {

        if (offboard_setpoint_counter_ == 0) {
            this->publish_vehicle_command(VehicleCommand::VEHICLE_CMD_DO_SET_MODE, 1, 6);

            this->arm();
        }

        publish_offboard_control_mode();
        publish_trajectory_setpoint();

        if (offboard_setpoint_counter_ % 40 == 0 && offboard_setpoint_counter_ > 41 && commandCount < 10)
        {
            commandCount++;
            if(commandCount % 2 == 0){angle += 2.5;}
        }
        if(commandCount == 10)
        {
            this->publish_vehicle_command(VehicleCommand::VEHICLE_CMD_NAV_LAND, 1, 0);
            if(offboard_setpoint_counter_==600) {this->disarm(); commandCount = 11;}
        }

        offboard_setpoint_counter_++;
    };
    timer_ = this->create_wall_timer(100ms, timer_callback);
}
```

Build package lintasan, jalankan agen dan juga simulasi, lalu node jalan_pentagram. Berikut video hasilnya:

px4-pentagram.mkv

https://drive.google.com/file/d/1gUNQ0XNy40735C5vBROYpv5HI131mNPw/view?usp=sharing