Laporan Pengerjaan Tugas 2 ROS2 Bayucaraka 2024

Muhammad Zia Alhambra | 5024231059

Tugas saya mulai dengan melakukan fork repo github dari Mas Juan. Lalu saya buat folder "tugas1", "tugas2", dan "tugas3" sesuai jumlah tugas yang diberikan. Lalu saya membuat folder baru "src" di dalam ketiga folder tersebut lalu membuat package dalam ketiga folder tersebut.

"*ros2 pkg create –build-type ament_cmake tugas1_cpp"*

"*ros2 pkg create –build-type ament_cmake tugas2_cpp"*

"*ros2 pkg create –build-type ament_cmake tugas3_cpp"*

Setelah membuat package, saya memulai coding dari CMakeLists.txt. code yang saya buat menggunakan rclcpp dan std_msgs maka saya menambahkannya pada CMakaLists.txt. Lalu saya menambahkan executable untuk publisher dan "subscriber". Saya menamakan publisher sebagai "talker" dan subscriber sebagai "listener" lalu saya meambahkan semuanya pada install.

```
find_package(ament_cmake REQUIRED)
find_package(rclcpp REQUIRED)
find_package(std_msgs REQUIRED)

add_executable(talker src/cpp_pub.cpp)
ament_target_dependencies(talker rclcpp std_msgs)

add_executable(listener src/cpp_sub.cpp)
ament_target_dependencies(listener rclcpp std_msgs)

install(TARGETS
  talker
  listener
  DESTINATION lib/${PROJECT_NAME}
)
```

Lalu saya lanjut ke package.xml. Disini saya hanya menambahkan rclcpp dan std_msgs agar pubsub bisa jalan.

```
<depend>rclcpp</depend>
<depend>std_msgs</depend>
```

Setalh CMakeLists dan package.xml telah diubah, saya lanjut coding publisher dan subscriber. Saya membuat 2 function yang yang melakukan generasi equation acak dan yang mempublish equation-nya ke subscriber.

Pada function generasi, saya declare angka acak sebagai "operand" dan operasi matematika sebagai "operation"

```
std::string generate_random_equation() {
  float operand1 = static_cast<float>(std::rand() % 10000 + 1);
  float operand2 = static_cast<float>(std::rand() % 10000 + 1);
  float operand3 = static_cast<float>(std::rand() % 10000 + 1);

  int operation1 = std::rand() % 5;
  int operation2 = std::rand() % 5;

  std::string equation;
  switch (operation1) {
    case 0:
      equation = std::to_string(operand1) + " + ";
      break;
    case 1:
      equation = std::to_string(operand1) + " - ";
      break;
    case 2:
      equation = std::to_string(operand1) + " * ";
      break;
    case 3:
      equation = std::to_string(operand1) + " / ";
      break;
    case 4:
      equation = std::to_string(static_cast<int>(operand1)) + " % ";
      break;
    default:
      break;
  }
```

(operation2 sama dengan operation1)

Setelah node publisher selesai, saya lanjut ke node subscriber. Node subscriber mempunyai tiga function yaitu function untuk melakukan operasi “perform_operation”, function untuk menghitung hasil “calculate_answer” dan function untuk menampilkan hasi di terminal “answer_callback”

Karena secara urutan operasi PEMDAS perkalian, pembagian, dan modulo lebih tinggi dari pertambahan dan perkurangan maka saya buat if statement supaya hasil lebih mematuhi urutan operasi PEMDAS.

```
float calculate_answer(const std::string& equation) {
  float operand1, operand2, operand3;
  char operation1, operation2;

  std::istringstream iss(equation);
  iss >> operand1 >> operation1 >> operand2 >> operation2 >> operand3;

  float result1, result2;

  if (operation1 == '*' || operation1 == '/' || operation1 == '%') {
      result1 = perform_operation(operand1, operand2, operation1);
      result2 = perform_operation(result1, operand3, operation2);
  } else if (operation1 == operation2) {
      result1 = perform_operation(operand1, operand2, operation1);
      result2 = perform_operation(result1, operand3, operation2);
  } else {
      result1 = perform_operation(operand2, operand3, operation2);
      result2 = perform_operation(operand1, result1, operation1);
  }
  return result2;
  }
```

Setelah selesai edit, saya build dengan "*colcon build"* di terminal dan menjalankan ros2 di dua terminal berbeda.