Nama : Hamdan Syaifuddin Zuhri
NIM : 1103220220
Kelas : TK-45-G09

Tugas Week 4 – Robotika
Command Secara Terurut

1. source /opt/ros/iron/setup.bash
2. sudo apt install python3-rosdep
3. sudo rosdep init
   rosdep update
   sudo apt update
   sudo apt dist-upgrade
4. sudo apt install python3-colcon-common-extensions
   sudo apt install python3-colcon-mixin
   colcon mixin add default https://raw.githubusercontent.com/colcon/colcon-mixin-repository/master/index.yaml
   colcon mixin update default
5. sudo apt install python3-vcstool
6. mkdir -p ~/ws_moveit/src
7. cd ~/ws_moveit/src
   git clone -b rolling https://github.com/moveit/moveit2_tutorials
8. vcs import --recursive < moveit2_tutorials/moveit2_tutorials.repos
9. sudo apt remove ros-jazzy
10. sudo apt update && rosdep install -r --from-paths . --ignore-src --rosdistro jazzy -y
11. cd ~/ws_moveit
    colcon build --mixin release
12. source ~/ws_moveit/install/setup.bash
13. echo 'source ~/ws_moveit/install/setup.bash' >> ~/.bashrc
14. ros2 launch moveit2_tutorials demo.launch.py
15. ros2 launch moveit2_tutorials demo.launch.py rviz_config:=your_rviz_config.rviz
16. ros2 pkg create \
    --build-type ament_cmake \
     --dependencies moveit_ros_planning_interface rclcpp \
     --node-name hello_moveit hello_moveit

17. Program yang dimasukkan ke file hello_moveit.cpp sebagai berikut:

    ```
    #include <memory>

    #include <rclcpp/rclcpp.hpp>
    #include <moveit/move_group_interface/move_group_interface.h>

    int main(int argc, char * argv[])
    {
      // Initialize ROS and create the Node
      rclcpp::init(argc, argv);
      auto const node = std::make_shared<rclcpp::Node>(
        "hello_moveit",
        rclcpp::NodeOptions().automatically_declare_parameters_from_overrides(true)
      );

      // Create a ROS logger
      auto const logger = rclcpp::get_logger("hello_moveit");

      // Next step goes here

      // Shutdown ROS
      rclcpp::shutdown();
      return 0;
    }
    ```

18. cd ~/ws_moveit
19. colcon build --mixin debug
20. cd ~/ws_moveit

    ```
    source install/setup.bash
    ```

21. ros2 run hello_moveit hello_moveit

    ```
    auto const node = std::make_shared<rclcpp::Node>(
      "hello_moveit",
      rclcpp::NodeOptions().automatically_declare_parameters_from_overrides(true)
    );
    ```

22. // Create a ROS logger

    ```
    auto const logger = rclcpp::get_logger("hello_moveit");
    ```

23. // Shutdown ROS

    ```
    rclcpp::shutdown();
    return 0;
    ```

24.
    ```
    // Create the MoveIt MoveGroup Interface
    using moveit::planning_interface::MoveGroupInterface;
    auto move_group_interface = MoveGroupInterface(node, "manipulator");

    // Set a target Pose
    auto const target_pose = []{
      geometry_msgs::msg::Pose msg;
      msg.orientation.w = 1.0;
      msg.position.x = 0.28;
      msg.position.y = -0.2;
      msg.position.z = 0.5;
      return msg;
    }();
    move_group_interface.setPoseTarget(target_pose);

    // Create a plan to that target pose
    auto const [success, plan] = [&move_group_interface]{
      moveit::planning_interface::MoveGroupInterface::Plan msg;
      auto const ok = static_cast<bool>(move_group_interface.plan(msg));
      return std::make_pair(ok, msg);
    }();

    // Execute the plan
    if(success) {
      move_group_interface.execute(plan);
    } else {
      RCLCPP_ERROR(logger, "Planning failed!");
    }
    ```
25.
    ```
    colcon build --mixin debug
    ```
26.
    ```
    ros2 launch moveit2_tutorials demo.launch.py
    ```
27.
    ```
    ros2 run hello_moveit hello_moveit
    ```
28.
    ```
    [ERROR] [1644181704.350825487] [hello_moveit]: Could not find parameter robot_description and did not receive robot_description via std_msgs::msg::String subscription within 10.000000 seconds.
    ```
29.
    ```
    using moveit::planning_interface::MoveGroupInterface;
    auto move_group_interface = MoveGroupInterface(node, "manipulator");
    ```
30.
    ```
    // Set a target Pose
    auto const target_pose = []{
      geometry_msgs::msg::Pose msg;
    ```

```
  msg.orientation.w = 1.0;
  msg.position.x = 0.28;
  msg.position.y = -0.2;
  msg.position.z = 0.5;
  return msg;
}();
move_group_interface.setPoseTarget(target_pose);

// Create a plan to that target pose
auto const [success, plan] = [&move_group_interface]{
  moveit::planning_interface::MoveGroupInterface::Plan msg;
  auto const ok = static_cast<bool>(move_group_interface.plan(msg));
  return std::make_pair(ok, msg);
}();
```

31.
    ```
    // Execute the plan
    if(success) {
      move_group_interface.execute(plan);
    } else {
      RCLCPP_ERROR(logger, "Planning failed!");
    }
    ```
32.
    ```
    <depend>moveit_visual_tools</depend>
    ```
33.
    ```
    find_package(moveit_visual_tools REQUIRED)
    ```
34.
    ```
    ament_target_dependencies(
      hello_moveit
      "moveit_ros_planning_interface"
      "moveit_visual_tools"
      "rclcpp"
    )
    ```
35.
    ```
    #include <moveit_visual_tools/moveit_visual_tools.h>
    ```
36.
    ```
    cd ~/ws_moveit
    colcon build --mixin debug
    ```
37.
    ```
    #include <thread>  // <---- add this to the set of includes at the top
    ```
38.
    ```
    // Create a ROS logger
    auto const logger = rclcpp::get_logger("hello_moveit");
      // Spin up a SingleThreadedExecutor for MoveItVisualTools to interact with ROS
      rclcpp::executors::SingleThreadedExecutor executor;
      executor.add_node(node);
      auto spinner = std::thread([&executor]() { executor.spin(); });
    ```

```
     // Create the MoveIt MoveGroup Interface

    ...
39. // Shutdown ROS
    rclcpp::shutdown();  // <--- This will cause the spin function in the thread to return
    spinner.join();  // <--- Join the thread before exiting
    return 0;
40. // Create the MoveIt MoveGroup Interface
    using moveit::planning_interface::MoveGroupInterface;
    auto move_group_interface = MoveGroupInterface(node, "manipulator");

    // Construct and initialize MoveItVisualTools
    auto moveit_visual_tools = moveit_visual_tools::MoveItVisualTools{
        node, "base_link", rviz_visual_tools::RVIZ_MARKER_TOPIC,
        move_group_interface.getRobotModel()};
    moveit_visual_tools.deleteAllMarkers();
    moveit_visual_tools.loadRemoteControl();
41. // Create closures for visualization
    auto const draw_title = [&moveit_visual_tools](auto text) {
      auto const text_pose = [] {
        auto msg = Eigen::Isometry3d::Identity();
        msg.translation().z() = 1.0;  // Place text 1m above the base link
        return msg;
      }();
      moveit_visual_tools.publishText(text_pose, text, rviz_visual_tools::WHITE,
                                      rviz_visual_tools::XLARGE);
    };
    auto const prompt = [&moveit_visual_tools](auto text) {
      moveit_visual_tools.prompt(text);
    };
    auto const draw_trajectory_tool_path =
        [&moveit_visual_tools,
         jmg = move_group_interface.getRobotModel()->getJointModelGroup(
             "manipulator")](auto const trajectory) {
          moveit_visual_tools.publishTrajectoryLine(trajectory, jmg);
        };
42. // Set a target Pose
    auto const target_pose = [] {
      geometry_msgs::msg::Pose msg;
```

```
  msg.orientation.w = 1.0;
  msg.position.x = 0.28;
  msg.position.y = -0.2;
  msg.position.z = 0.5;
  return msg;
}();
move_group_interface.setPoseTarget(target_pose);

// Create a plan to that target pose
prompt("Press 'Next' in the RvizVisualToolsGui window to plan");
draw_title("Planning");
moveit_visual_tools.trigger();
auto const [success, plan] = [&move_group_interface] {
  moveit::planning_interface::MoveGroupInterface::Plan msg;
  auto const ok = static_cast<bool>(move_group_interface.plan(msg));
  return std::make_pair(ok, msg);
}();

// Execute the plan
if (success) {
  draw_trajectory_tool_path(plan.trajectory);
  moveit_visual_tools.trigger();
  prompt("Press 'Next' in the RvizVisualToolsGui window to execute");
  draw_title("Executing");
  moveit_visual_tools.trigger();
  move_group_interface.execute(plan);
} else {
  draw_title("Planning Failed!");
  moveit_visual_tools.trigger();
  RCLCPP_ERROR(logger, "Planning failed!");
}
```

43.
    ```
    cd ~/ws_moveit
    source /opt/ros/rolling/setup.bash
    colcon build --mixin debug
    ```
44.
    ```
    cd ~/ws_moveit
    source install/setup.bash
    ros2 launch moveit2_tutorials demo.launch.py
    ```
45.
    ```
    cd ~/ws_moveit
    ```

source install/setup.bash
ros2 run hello_moveit hello_moveit

46. ros2 launch moveit2_tutorials demo.launch.py rviz_config:=kinova_hello_moveit.rviz
47. cd ~/ws_moveit/src
git clone -b <branch> https://github.com/moveit/moveit_task_constructor.git
48. rosdep install --from-paths . --ignore-src --rosdistro $ROS_DISTRO
49. cd ~/ws_moveit
colcon build --mixin release
50. ros2 launch moveit_task_constructor_demo demo.launch.py
ros2 launch moveit_task_constructor_demo run.launch.py exe:=cartesian
ros2 launch moveit_task_constructor_demo run.launch.py exe:=modular
ros2 launch moveit_task_constructor_demo run.launch.py exe:=pick_place_demo
51. ros2 pkg create \
--build-type ament_cmake \
--dependencies moveit_task_constructor_core rclcpp \
--node-name mtc_node mtc_tutorial
52. cd ~/ws_moveit
colcon build --mixin debug
53. In the second terminal ros2 launch moveit2_tutorials demo.launch.py
54. In the third terminal ros2 run hello_moveit hello_moveit
55. cd ~/ws_moveit
colcon build --mixin debug
56. cd ~/ws_moveit
source /opt/ros/rolling/setup.bash
colcon build --mixin debug
57. Open new terminal
cd ~/ws_moveit
source install/setup.bash
ros2 launch moveit2_tutorials demo.launch.py
58. cd ~/ws_moveit
source install/setup.bash
ros2 run hello_moveit hello_moveit
59. cd ~/ws_moveit/src
git clone -b rolling https://github.com/moveit/moveit_task_constructor.git
60. rosdep install --from-paths . --ignore-src --rosdistro jazzy
61. cd ~/ws_moveit

```
    colcon build --mixin release
62. ros2 launch moveit_task_constructor_demo demo.launch.py
63. ros2 launch moveit_task_constructor_demo run.launch.py exe:=cartesian
64. ros2 launch moveit_task_constructor_demo run.launch.py exe:=modular
65. ros2 launch moveit_task_constructor_demo run.launch.py exe:=pick_place_demo
66. ros2 pkg create \
    --build-type ament_cmake \
    --dependencies moveit_task_constructor_core rclcpp \
    --node-name mtc_node mtc_tutorial

67. cd ~/ws_moveit
68. colcon build --mixin release
69. source ~/ws_moveit/install/setup.bash
70. ros2 launch moveit2_tutorials mtc_demo.launch.py
71. ros2 launch mtc_tutorial pick_place_demo.launch.py
72. ros2 launch moveit2_tutorials mtc_demo_minimal.launch.py
73. ros2 launch mtc_tutorial pick_place_demo.launch.py
74. ros2 launch mtc_tutorial pick_place_demo.launch.py
75. ros2 launch moveit2_tutorials mtc_demo.launch.py
76. ros2 launch moveit2_tutorials pick_place_demo.launch.py
77. uint32_t const unique_stage_id = task_.stages()->findChild(stage_name)-
    >introspectionId();
```