D

Berikut adalah langkah-langkah cara kontrol dan komunikasi servo tersebut menggunakan sistem publish subscribe pada Raspberry Pi :

1. Siapkan komponen yang diperlukan, yaitu
    - seluruh komponen yang dijelaskan pada 3.c,
    - Dynamixel AX-18A beserta kabel-kabel yang termasuk dengannya
    - Dynamixel U2D2 beserta kabel-kabel yang termasuk dengannya
2. Rakitkan komponen-komponen tersebut sehingga
    - Dynamixel AX-18A terpasang dengan Dynamixel U2D2
    - Dynamixel U2D2 terhubung dengan powersupply dan Raspberry Pi
    - Raspberry Pi terhubung seperti pada 3.c
3. Menyalakan perangkat anda dengan Raspberry Pi OS
4. Instalasi software Dynamixel SDK dengan ketik ini satu persatu line di command prompt

    ```
    sudo apt-get install -y cmake
    git clone https://github.com/ROBOTIS-GIT/DynamixelSDK.git
    cd DynamixelSDK/c++/build
    cmake ..
    make
    sudo make install
    ```

5. Install ROS pada Raspberry Pi OS
6. Konfigurasi komunikasi servo
    - Bikin node Publisher dengan ROS. Contoh :

    ```
    import rospy
    from std_msgs.msg import Float64

    def servo_control():
        pub = rospy.Publisher('servo_position', Float64, queue_size=10)
        rospy.init_node('servo_publisher', anonymous=True)
        rate = rospy.Rate(10)  # 10Hz
        while not rospy.is_shutdown():
            position = float(input("Enter desired servo position: "))
            pub.publish(position)
            rate.sleep()

    if __name__ == '__main__':
        try:
            servo_control()
        except rospy.ROSInterruptException:
            pass
    ```

- Bikin node Subscriber dengan ROS. Contoh :

```
import rospy
from std_msgs.msg import Float64
from dynamixel_sdk import *  # Library SDK Dynamixel

# Setup kontroler
port_handler = PortHandler('/dev/ttyUSB0')
packet_handler = PacketHandler(1.0)  # Versi protokol AX-18A

# Buka port dan set baudrate
port_handler.openPort()
port_handler.setBaudRate(1000000)  # AX-18A biasanya pada 1 Mbps

def move_servo(position):
    # Pastikan posisi dalam rentang 0-1023
    if 0 <= position <= 1023:
        packet_handler.write2ByteTxRx(port_handler, servo_id, 30,
int(position))

def callback(data):
    move_servo(data.data)

def servo_subscriber():
    rospy.init_node('servo_subscriber', anonymous=True)
    rospy.Subscriber('servo_position', Float64, callback)
    rospy.spin()

if __name__ == '__main__':
    try:
        servo_subscriber()
    except rospy.ROSInterruptException:
        port_handler.closePort()
```

7. Jalankan Program
   Pada terminal berbeda
   - Jalankan roscore untuk memulai master ROS
   - Jalankan publisher python3 publisher.py dan masukkan nilai posisi
   - Jalankan subscriber python3 subscriber.py yang akan menggerakkan servo ke posisi yang diterima dari publisher.