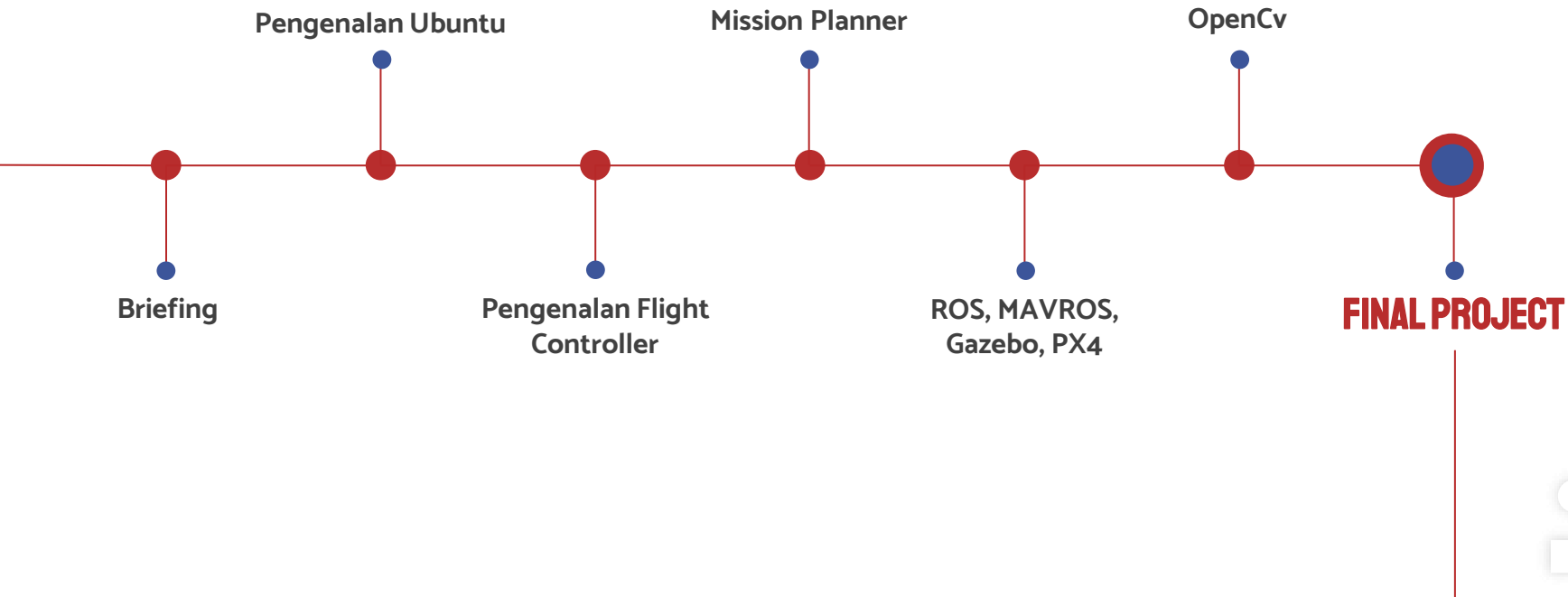
INTERNSHIP

BAYUCARAKA

ADNAN ABDULLAH JUAN | 5025221155

TIMELINE


Briefing
Pengenalan Ubuntu
Pengenalan Flight Controller
Mission Planner
ROS, MAVROS, Gazebo, PX4
OpenCv
FINAL PROJECT

FINAL PROJECT

MAIN CODE

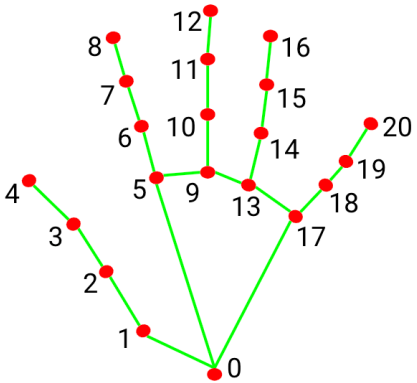
COMPUTER VISION

COMPUTER VISION

```python
import rospy
import cv2 as cv
import cv_bridge
import math
import mediapipe as mp
from std_msgs.msg import String
from bayucaraka.msg import poss

cap = cv.VideoCapture(0)
mpHands = mp.solutions.hands
hands = mpHands.Hands()
mpDraw = mp.solutions.drawing_utils
fingerCode = [4, 8, 12, 16, 20]
velDrone = 0
```

Finger Code yang dimaksud adalah
Landmark dari tangan

HAND LANDMARK MODEL



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

https://google.github.io/mediapipe/solutions/hands#python-solution-api

FUNCTION

```python
def findHands(img):
    imgRgb = cv.cvtColor(img, cv.COLOR_BGR2RGB)
    res = hands.process(imgRgb)
    # menggambar landmark tangan
    if res.multi_hand_landmarks:
        for lms in res.multi_hand_landmarks:
            mpDraw.draw_landmarks(img, lms, mpHands.HAND_CONNECTIONS)
    return img

def findPos(img, handNo=0):
    lmList = []
    imgRgb = cv.cvtColor(img, cv.COLOR_BGR2RGB)
    res = hands.process(imgRgb)
    if res.multi_hand_landmarks:
        # mencari koordinat dan id dari landmarks tangan
        myHand = res.multi_hand_landmarks[handNo]
        for id, lm in enumerate(myHand.landmark):
            h, w, c = img.shape
            cx, cy = int(lm.x * w), int(lm.y * h)
            lmList.append([id, cx, cy])
    return lmList
```

```python
def pitagoras(a, b):
    return math.sqrt(a*a + b*b)

def findDis (img, tip1, tip2, lmList):
    x1, y1 = lmList[tip1][1], lmList[tip1][2]
    x2, y2 = lmList[tip2][1], lmList[tip2][2]

    cv.circle(img,(x1,y1), 5, (0,0,0), cv.FILLED)
    cv.circle(img,(x2,y2), 5, (0,0,0), cv.FILLED)
    cv.line(img, (x1,y1), (x2,y2), (0,0,0), 3)

    # mencari jarak antara 2 buah jari
    length = pitagoras((x2-x1), (y2-y1))
    smooth = 10
    velocity = smooth * round(length/smooth)
    velocity = int(velocity/40)
    return velocity, img
```

```python
def talking():
    pub = rospy.Publisher("finalBayucaraka", poss, queue_size=10)
    rospy.init_node("publisher", anonymous=True)
    rate = rospy.Rate(10)
    while not rospy.is_shutdown():
        success, img = cap.read()
        # flip image
        img = cv.flip(img, 1)
        img = cv.resize(img,(800,600))
        img = findHands(img)
        lmList = findPos(img)
        msg = poss()
        fingers = []
        if len(lmList) != 0:
            # thumb
            if lmList[4][1] < lmList[3][1]:
                # opened
                fingers.append(1)
            else:
                # closed
                fingers.append(0)

            # fingers
            for id in range(1,5):
                if lmList[fingerCode[id]][2] < lmList[fingerCode[id]-2][2]:
                    fingers.append(1)
                else:
                    fingers.append(0)

            # publish x y middle finger
            msg.pos_x = (lmList[12][1] - 400)/20
            msg.pos_y = (lmList[12][2] - 300)/-20
```

```python
def findHands(img):
    imgRgb = cv.cvtColor(img, cv.COLOR_BGR2RGB)
    res = hands.process(imgRgb)
    # menggambar landmark tangan
    if res.multi_hand_landmarks:
        for lms in res.multi_hand_landmarks:
            mpDraw.draw_landmarks(img, lms,
mpHands.HAND_CONNECTIONS)
    return img
```

```python
def findPos(img, handNo=0):
    lmList = []
    imgRgb = cv.cvtColor(img, cv.COLOR_BGR2RGB)
    res = hands.process(imgRgb)
    if res.multi_hand_landmarks:
        # mencari koordinat dan id dari landmarks tangan
        myHand = res.multi_hand_landmarks[handNo]
        for id, lm in enumerate(myHand.landmark):
            h, w, c = img.shape
            cx, cy = int(lm.x * w), int(lm.y * h)
            lmList.append([id, cx, cy])
    return lmList
```

```python
        if(fingers == [1,1,0,0,0]):
            global velDrone
            # find distace between thump and index finger
            velDrone, img = findDis(img, 4, 8, lmList)
            # publish status drone
            msg.status = 6
        else:
            totalFinger = fingers.count(1)
            # publish status drone
            msg.status = totalFinger

        statusDrone = msg.status
        # publish velocity for drone
        msg.vel_res = velDrone

        # make a circle in center of image
        cv.circle(img,(400,300),5,[150,0,0],cv.FILLED)
        # show velocity and status
        if statusDrone == 6:
            cv.putText(img,f"status : change velocity",(20,30), cv.FONT_HERSHEY_DUPLEX,1,[0,0,0],2)
        else:
            cv.putText(img,f"status : {statusDrone}",(20,30), cv.FONT_HERSHEY_DUPLEX,1,[0,0,0],2)
        cv.putText(img,f"velocity : {velDrone}",(20,60), cv.FONT_HERSHEY_DUPLEX,1,[0,0,0],2)

        cv.imshow("image", img)
        pub.publish(msg)
        rate.sleep()
        if cv.waitKey(1) & 0xFF == ord('q'):
            break

if __name__ == "__main__":
    try:
        talking()
    except rospy.ROSInterruptException:
        pass
```

```python
def pitagoras(a, b):
    return math.sqrt(a*a + b*b)

def findDis (img, tip1, tip2, lmList):
    x1, y1 = lmList[tip1][1], lmList[tip1][2]
    x2, y2 = lmList[tip2][1], lmList[tip2][2]

    cv.circle(img,(x1,y1), 5, (0,0,0), cv.FILLED)
    cv.circle(img,(x2,y2), 5, (0,0,0), cv.FILLED)
    cv.line(img, (x1,y1), (x2,y2), (0,0,0), 3)

    # mencari jarak antara 2 buah jari
    length = pitagoras((x2-x1), (y2-y1))
    smooth = 10
    velocity = smooth * round(length/smooth)
    velocity = int(velocity/40)
    return velocity, img
```

MAIN CODE

INCLUDE

```
#include "ros/ros.h"
#include "bayucaraka/poss.h"
#include <geometry_msgs/PoseStamped.h>
#include <geometry_msgs/Twist.h>
#include <mavros_msgs/CommandBool.h>
#include <mavros_msgs/SetMode.h>
#include <mavros_msgs/State.h>
#include <mavros_msgs/PositionTarget.h>
#include <tf2_geometry_msgs/tf2_geometry_msgs.h>
#include <cmath>
#define PI 3.141592654

ros::Publisher local_pos_pub;
ros::Publisher local_vel_pub;
geometry_msgs::PoseStamped pose;
geometry_msgs::Twist msg;
mavros_msgs::State current_state;
tf2::Quaternion mengrotate;
double xnya, ynya, znya, inRad;
float x, y, vel_resultan = 0;
int stts;
```

FUNCTION

```
double pitagoras(double a, double b){
    return sqrt(a*a + b*b);
}

void gerak(double xTujuan, double yTujuan){
    double jarakNow = pitagoras(xTujuan-xnya, yTujuan-ynya);
    // sudutnya akan selalu berubah
    double sudut = (yTujuan-ynya) / (jarakNow);
    double inDegree = asin(sudut)*180/PI;

    // sudutnya itu 0 sampe 180 lalu -180 sampe 0
    if(yTujuan > ynya && xTujuan < xnya) inDegree = 180-inDegree;
    if(yTujuan < ynya && xTujuan < xnya) inDegree = -180-inDegree;
    double inRad = inDegree * PI / 180;

    // dengan adanya informasi sudut dari tujuan, bisa mencari v_x dan v_y
    msg.linear.x = vel_resultan*cos(inRad);
    msg.linear.y = vel_resultan*sin(inRad);

    local_vel_pub.publish(msg);
}
```

```
void state_cb(const mavros_msgs::State::ConstPtr& msg){
    current_state = *msg;
}

void chatterCallback(const bayucaraka::poss::ConstPtr& msg)
{
    stts = msg->status;
    vel_resultan = msg->vel_res;
    x = msg->pos_x;
    y = msg->pos_y;
}

void callBack(const geometry_msgs::PoseStamped::ConstPtr& oke){
    xnya = oke->pose.position.x;
    ynya = oke->pose.position.y;
    znya = oke->pose.position.z;
}
```

MAIN

```cpp
int main(int argc, char **argv)
{
    ros::init(argc, argv, "listener");
    ros::NodeHandle nh;

    ros::Subscriber sub = nh.subscribe("finalBayucaraka", 1000, chatterCallback);
    ros::Subscriber state_sub = nh.subscribe<mavros_msgs::State>("mavros/state", 15, state_cb);
    ros::Subscriber inpo = nh.subscribe<geometry_msgs::PoseStamped>("mavros/local_position/pose", 15, callBack);
    local_pos_pub = nh.advertise<geometry_msgs::PoseStamped>("mavros/setpoint_position/local", 15);
    local_vel_pub = nh.advertise<geometry_msgs::Twist>("mavros/setpoint_velocity/cmd_vel_unstamped", 15);
    ros::ServiceClient arming_client = nh.serviceClient<mavros_msgs::CommandBool>("mavros/cmd/arming");
    ros::ServiceClient set_mode_client = nh.serviceClient<mavros_msgs::SetMode>("mavros/set_mode");

    //the setpoint publishing rate MUST be faster than 2Hz
    ros::Rate rate(20);

    // wait for FCU connection
    while(ros::ok() && !current_state.connected){
        ros::spinOnce();
        rate.sleep();
    }
    pose.pose.position.x = 0;
    pose.pose.position.y = 0;
    pose.pose.position.z = 2;
    for(int i = 100; ros::ok() && i > 0; --i){
        local_pos_pub.publish(pose);
        ros::spinOnce();
        rate.sleep();
    }
    mavros_msgs::SetMode offb_set_mode;
    offb_set_mode.request.custom_mode = "OFFBOARD";

    mavros_msgs::CommandBool arm_cmd;
    arm_cmd.request.value = true;

    ros::Time last_request = ros::Time::now();
```

OFFBOARD MODE

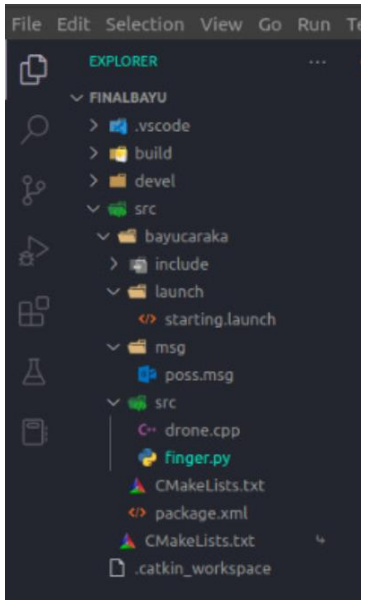
```
    while(ros::ok()){
        if( current_state.mode != "OFFBOARD" &&
            (ros::Time::now() - last_request > ros::Duration(5.0))){
            if( set_mode_client.call(offb_set_mode) &&
                offb_set_mode.response.mode_sent){
                ROS_INFO("Offboard enabled");
            }
            last_request = ros::Time::now();
        } else {
            if( !current_state.armed &&
                (ros::Time::now() - last_request > ros::Duration(5.0))){
                if( arming_client.call(arm_cmd) &&
                    arm_cmd.response.success){
                    ROS_INFO("Vehicle armed");
                }
                last_request = ros::Time::now();
            }
        }
        // jika udah diposisi nya break dari loopingan
        if(current_state.mode == "OFFBOARD" && current_state.armed && znya >= 0.5) break;
        local_pos_pub.publish(pose);

        ros::spinOnce();
        rate.sleep();
    }
```

```cpp
int rot = 0;
while(ros::ok()){
    ros::Rate loop_rate(15);

    if(stts == 1){
        // drone naik
        pose.pose.position.z += 0.05;
    } else if(stts == 2){
        // drone turun
        pose.pose.position.z -= 0.05;
    } else if(stts == 4){
    //      // gerak melingkar
        rot+=1;
        if(rot > 180) rot -=360;
        gerak((10*cos(rot*PI/180)),(10*sin(rot*PI/180)));
    }else if(stts == 5){
        // gerak mengikuti tangan (khususnya jari tengah)
        gerak(x,y);
    }

    if(stts != 1 && stts != 2){
        pose.pose.position.z = znya;
    }

    // eular to quartenion
    // https://eater.net/quaternions
    // http://wiki.ros.org/tf2/Tutorials/Quaternions
    mengrotate.setRPY(0, 0 , inRad);
    mengrotate = mengrotate.normalize();
    pose.pose.position.x = xnya;
    pose.pose.position.y = ynya;
    pose.pose.orientation.x = mengrotate.getX();
    pose.pose.orientation.y = mengrotate.getY();
    pose.pose.orientation.z = mengrotate.getZ();
    pose.pose.orientation.w = mengrotate.getW();

    local_pos_pub.publish(pose);

    ros::spinOnce();
    loop_rate.sleep();
  }
 return 0;
}
```

```cpp
double pitagoras(double a, double b){
    return sqrt(a*a + b*b);
}

void gerak(double xTujuan, double yTujuan){
    double jarakNow = pitagoras(xTujuan-xnya, yTujuan-ynya);
    // sudutnya akan selalu berubah
    double sudut = (yTujuan-ynya) / (jarakNow);
    double inDegree = asin(sudut)*180/PI;

    // sudutnya itu 0 sampe 180 lalu -180 sampe 0
    if(yTujuan > ynya && xTujuan < xnya) inDegree = 180-inDegree;
    if(yTujuan < ynya && xTujuan < xnya) inDegree = -180-inDegree;
    double inRad = inDegree * PI / 180;

    // dengan adanya informasi sudut dari tujuan, bisa mencari v_x dan v_y
    msg.linear.x = vel_resultan*cos(inRad);
    msg.linear.y = vel_resultan*sin(inRad);

    local_vel_pub.publish(msg);
}
```

https://eater.net/quaternions

http://wiki.ros.org/tf2/Tutorials/Quaternions



Custom message

```
uint32 status
float32 vel_res
float32 pos_x
float32 pos_y
```

Launch file

```
<launch>
    <node name="mengFinger" pkg="bayucaraka" type="finger.py" output="screen"></node>
    <node name="mengDrone" pkg="bayucaraka" type="drone" output="screen"></node>
</launch>
```

Add executable file

```
include_directories(
include
  ${catkin_INCLUDE_DIRS}
)

catkin_install_python(PROGRAMS src/finger.py
  DESTINATION ${CATKIN_PACKAGE_BIN_DESTINATION}
)

add_executable(drone src/drone.cpp)
target_link_libraries(drone ${catkin_LIBRARIES})
```

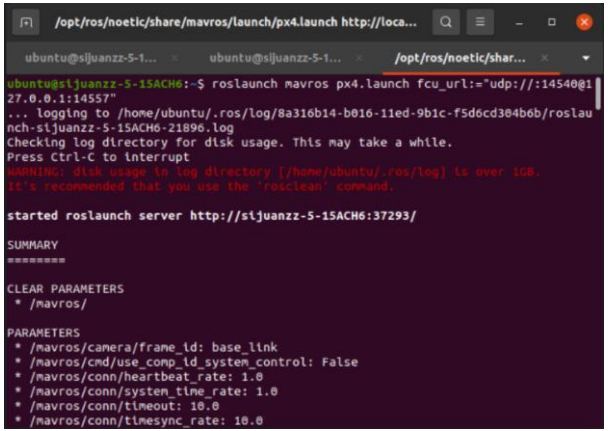
CMakeLists.txt

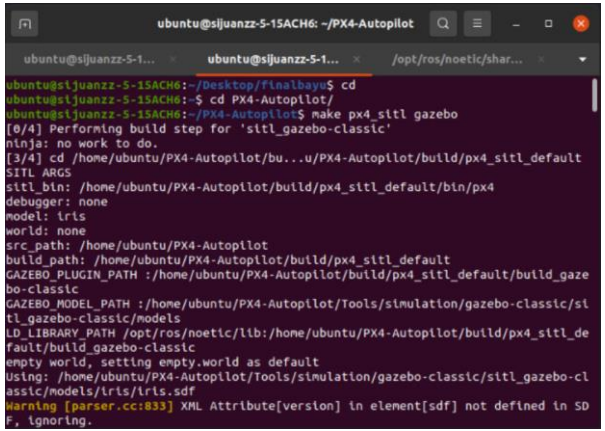
```
find_package(catkin REQUIRED COMPONENTS
  cv_bridge
  geometry_msgs
  mavros_msgs
  message_generation
  roscpp
  rospy
  std_msgs
  tf2_geometry_msgs
)

## Generate messages in the 'msg' folder
add_message_files(
  FILES
  poss.msg
)

## Generate added messages and services with any dependencies listed here
generate_messages(
  DEPENDENCIES
  #   geometry_msgs#   mavros_msgs
  std_msgs
)
```

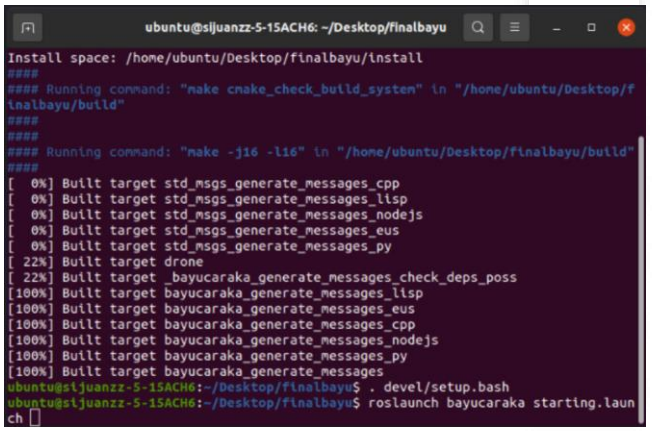
Package.xml

```
<build_depend>message_generation</build_depend>
<build_export_depend>message_generation</build_export_depend>
<exec_depend>message_runtime</exec_depend>
```

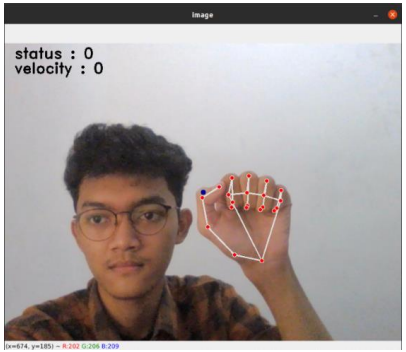
HOW TO RUN

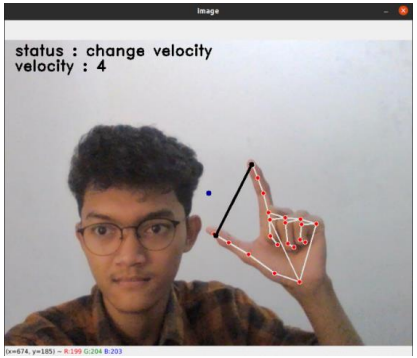

/opt/ros/noetic/share/mavros/launch/px4.launch http://loca...
ubuntu@sijuanzz-5-1...
ubuntu@sijuanzz-5-1...
/opt/ros/noetic/shar...
ubuntu@sijuanzz-5-15ACH6:~$ roslaunch mavros px4.launch fcu_url:="udp://:14540@127.0.0.1:14557"
... logging to /home/ubuntu/.ros/log/8a316b14-b016-11ed-9b1c-f5d6cd304b6b/roslaunch-sijuanzz-5-15ACH6-21896.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
WARNING: disk usage in log directory [/home/ubuntu/.ros/log] is over 1GB.
It's recommended that you use the 'rosclean' command.
started roslaunch server http://sijuanzz-5-15ACH6:37293/
SUMMARY
CLEAR PARAMETERS
* /mavros/
PARAMETERS
* /mavros/camera/frame_id: base_link
* /mavros/cmd/use_comp_id_system_control: False
* /mavros/conn/heartbeat_rate: 1.0
* /mavros/conn/system_time_rate: 1.0
* /mavros/conn/timeout: 10.0
* /mavros/conn/timesync_rate: 10.0

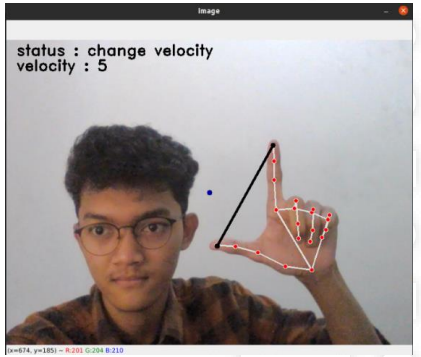

ubuntu@sijuanzz-5-15ACH6: ~/PX4-Autopilot
ubuntu@sijuanzz-5-1...
ubuntu@sijuanzz-5-1...
/opt/ros/noetic/shar...
ubuntu@sijuanzz-5-15ACH6:~/Desktop/finalbayu$ cd
ubuntu@sijuanzz-5-15ACH6:~$ cd PX4-Autopilot/
ubuntu@sijuanzz-5-15ACH6:~/PX4-Autopilot$ make px4_sitl gazebo
[0/4] Performing build step for 'sitl_gazebo-classic'
ninja: no work to do.
[3/4] cd /home/ubuntu/PX4-Autopilot/bu...u/PX4-Autopilot/build/px4_sitl_default
SITL ARGS
sitl_bin: /home/ubuntu/PX4-Autopilot/build/px4_sitl_default/bin/px4
debugger: none
model: iris
world: none
src_path: /home/ubuntu/PX4-Autopilot
build_path: /home/ubuntu/PX4-Autopilot/build/px4_sitl_default
GAZEBO_PLUGIN_PATH :/home/ubuntu/PX4-Autopilot/build/px4_sitl_default/build_gazebo-classic
GAZEBO_MODEL_PATH :/home/ubuntu/PX4-Autopilot/Tools/simulation/gazebo-classic/sitl_gazebo-classic/models
LD_LIBRARY_PATH /opt/ros/noetic/lib:/home/ubuntu/PX4-Autopilot/build/px4_sitl_default/build_gazebo-classic
empty world, setting empty.world as default
Using: /home/ubuntu/PX4-Autopilot/Tools/simulation/gazebo-classic/sitl_gazebo-classic/models/iris/iris.sdf
Warning [parser.cc:833] XML Attribute[version] in element[sdf] not defined in SDF, ignoring.

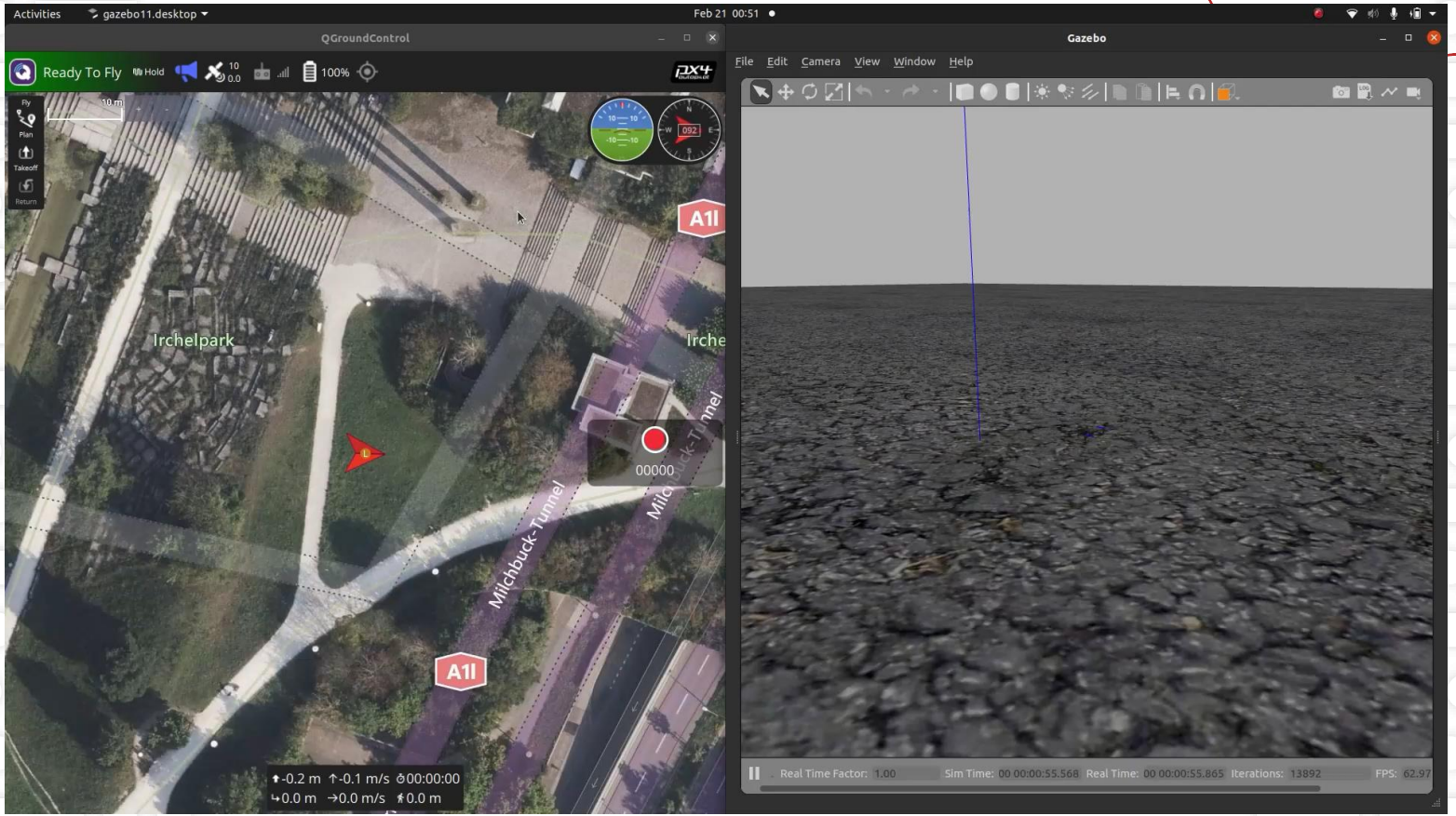

ubuntu@sijuanzz-5-15ACH6: ~/Desktop/finalbayu
Install space: /home/ubuntu/Desktop/finalbayu/install
Running command: "make cmake_check_build_system" in "/home/ubuntu/Desktop/finalbayu/build"
Running command: "make -j16 -l16" in "/home/ubuntu/Desktop/finalbayu/build"
[0%] Built target std_msgs_generate_messages_cpp
[0%] Built target std_msgs_generate_messages_lisp
[0%] Built target std_msgs_generate_messages_nodejs
[0%] Built target std_msgs_generate_messages_eus
[0%] Built target std_msgs_generate_messages_py
[22%] Built target drone
[22%] Built target _bayucaraka_generate_messages_check_deps_poss
[100%] Built target bayucaraka_generate_messages_lisp
[100%] Built target bayucaraka_generate_messages_eus
[100%] Built target bayucaraka_generate_messages_cpp
[100%] Built target bayucaraka_generate_messages_nodejs
[100%] Built target bayucaraka_generate_messages_py
[100%] Built target bayucaraka_generate_messages
ubuntu@sijuanzz-5-15ACH6:~/Desktop/finalbayu$. devel/setup.bash
ubuntu@sijuanzz-5-15ACH6:~/Desktop/finalbayu$ roslaunch bayucaraka starting.launch

DOCUMENTATION









Link Video : https://youtu.be/gOb8xgLG5hw

BAYUCARAKA ITS

GARUDAKU JAYA SELALU

#flytoinfinite