

NIM : 16521094
Nama : Hilmi Baskara Radanto
Fakultas : STEI
Jurusan : TPB

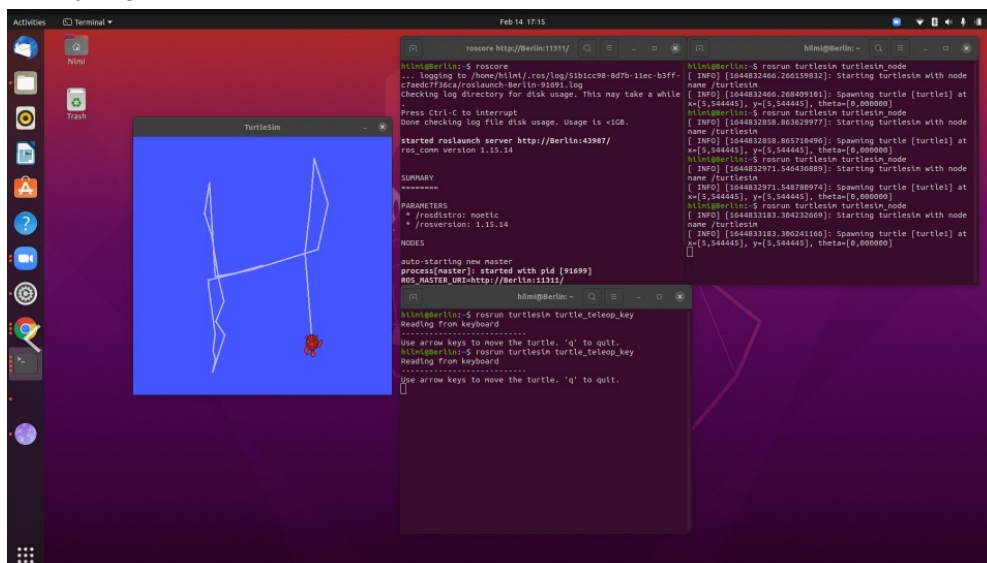
SOAL

Penjurusan ROS & OpenCV

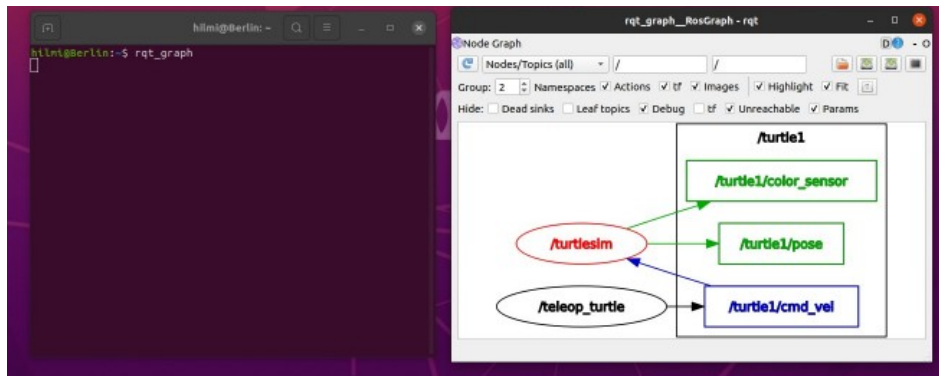
1. Buatlah Jalur yang dibuat oleh TurtleSim (dengan Keyboard) sesuai dengan huruf awalan pada nama. (5 poin)
2. Tampilkan rqt graph yang ada pada soal nomor 1. Kemudian, jelaskan Isi graph yang ada pada RQT Graph tersebut! (20 poin)
Hint: Masukkan Command rqt_graph pada terminal
3. Buatlah suatu program (source code) yang dapat mendeteksi dan men-decode QR Code yang terdapat pada KSM Masing-masing di SIX ITB dengan menggunakan OpenCV!
 - a. Jangan lupa sertakan kode programnya pada lembar jawaban dan upload kode program di repository github masing2!
 - b. Tampilkan Isi output dari Terminal dan Tampilan web QR Code tersebut
4. Jelaskan kode yang dibuat pada nomor 3 berdasarkan masing2 fungsinya! (10 poin)
5. Sebutkan dan jelaskan 4 konsep dasar OOP (Object Oriented Programming)! (10 poin)
6. Jelaskan kelebihan dan kekurangan OOP (Object Oriented Programming)! (10 poin)
7. Buatlah program untuk menghitung jarak antara 2 titik koordinat 3D! (15 poin)
Input : 2 titik pada peta, setiap titik berisi latitude, longitude, dan altitude (meter)
Output: Jarak (meter)
Test Case :
Input : 9, 3, 2
 3, 7, 4
Output: 7.48

Jawaban

1. Berikut screenshot hasil jalur TurtleSim dengan keyboard berupa huruf H beserta code yang diinput di terminal



2. Berikut screenshot tampilan setelah memasukkan `rqt_graph` pada terminal



Rqt_graph menampilkan sebuah visualisasi mengenai bagaimana hubungan antar node di dalam ROS. Biasanya digunakan untuk melakukan debugging dan double-check terhadap ROS. Beberapa penjelasan diagramnya sebagai berikut:

- Penjelasan bentuk
 - Bentuk oval berisi node
 - Bentuk persegi panjang merupakan topic
 - Panah menunjukkan hubungan topic dengan node
 - Persegi panjang besar yang bertuliskan `/turtle1` dengan 3 persegi panjang kecil lainnya di dalam merupakan workspace yang berisi apa saja yang sedang disimulasikan
- Penjelasan diagram rqt_graph yang ditampilkan
`/teleop_turtle` merupakan node yang menerima input dari panah keyboard kemudian diteruskan ke topic `/turtle1/cmd_vel` yang menerima perintah kecepatan. `/turtle1/cmd_vel` termasuk di dalam space `.turtle1` Kemudian informasi dan data diteruskan ke `/turtlesim`, lalu dilanjutkan ke `/turtle1/pose` dan `/turtle1/color_sensor` yang merupakan bagian dari `/turtle1`
- Jika mouse diarahkan ke salah satu bentuk maka akan menampilkan warna dengan penjelasan:
 - Merah → current topic, topik yang sedang dilihat/diarahkan mouse
 - Biru → subscribed topic, yang mengirim data dan informasi ke current topic
 - Hijau → published topic, current topic mengirim data ke bentuk tersebut

Kemudian terdapat beberapa pilihan menu yang dapat menampilkan grafik sesuai seperti apa yang ingin kita lihat. Beberapa variasinya diantara lain:

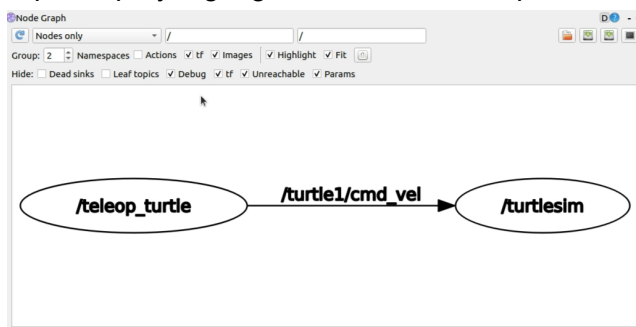


Diagram tersebut menunjukkan node /teleop_turtle yang menerima input dari panah keyboard meneruskan data dengan topic /turtle1/cmd_vel menuju node simulasi /turtlesim

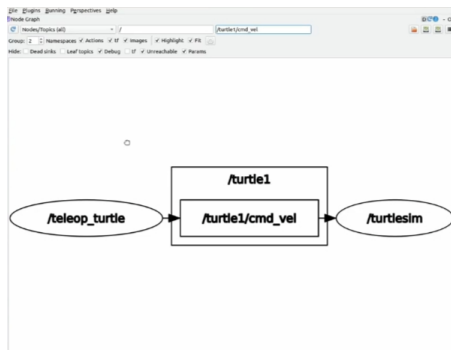


Diagram tersebut menunjukkan node /teleop_turtle yang menerima input dari panah keyboard meneruskan data dengan topic /turtle1/cmd_vel yang merupakan bagian dari space /turtle1 menuju node simulasi /turtlesim

3. Berikut merupakan code program untuk mendeteksi dan mendecode QR Code yang terdapat pada KSM:

```

import cv2
import numpy as np
from pyzbar.pyzbar import decode

img = cv2.imread('QRKSM.jpg')

for code in decode(img):
    print(code.type)
    print(code.data.decode('utf-8'))
  
```

- a. Berikut merupakan screenshot source code pada VS Code

```

barcodescan2.py > ...
1  import cv2
2  import numpy as np
3  from pyzbar.pyzbar import decode
4
5  img = cv2.imread('QRKSM.jpg')
6
7  for code in decode(img):
8      print(code.type)
9      print(code.data.decode('utf-8'))
  
```

Dan berikut adalah [link repository github](https://github.com/hilmiAksantara/barcodescan2) dari source code di atas

- b. Berikut merupakan isi output dari terminal

```

hilmi@Berlin:~/Code/Aksantara$ /bin/python3.9 /home/hilmi/Code/Aksantara/barcodescan2.py
QRCODE
https://akademik.itb.ac.id/ver/ksm/2021/2/2021004736/8a36385a47ecde8445121683ba5b5d96c6a00a14
  
```

Berikut merupakan tampilan web QR Codenya

Activities

Google Chrome

Kartu Studi Mahasiswa

Feb 19 11:28

ad

4. Penjelasan kode pada nomor 3

```
import cv2
```

Import library cv2 untuk melakukan pembacaan gambar

```
from pyzbar.pyzbar import decode
```

Import decode dari library pyzbar untuk melakukan scan pada qr code

```
img = cv2.imread('QRKSM.jpg')
```

Melakukan pembacaan terhadap file gambar QRKSM.jpg di satu folder yang sama dengan library cv2

```
for code in decode(img):
```

Menggunakan fungsi decode pada konten yang ada di gambar, seperti type code yang discan, bentuk code yang discan, dan posisi code pada gambar.

```
print(code.type)
```

Mengeluarkan type code apa yang berhasil discan, seperti QRCODE

```
print(code.data.decode('utf-8'))
```

Mengeluarkan output dari hasil code yang telah discan

5. Object Oriented Programming adalah metode pemrograman yang berorientasi pada objek yang bertujuan untuk mempermudah pengembangan program dengan mengikuti model yang telah ada di kehidupan sehari-hari. Terdapat 4 konsep dasar pada OOP (Object Oriented Programming), yaitu:

a. Inheritance

Secara harfiah berarti pewarisan. Inheritance adalah proses dimana suatu objek memperoleh sifat serta perilaku dari objek lainnya. Hal ini disebabkan karena konsep inheritance memungkinkan ketika mendefinisikan sebuah class (induk) ke class lain (anak) dan memberi kesempatan menggunakan member (function, property, variabel) dari class yang diwariskan tersebut. Keuntungannya membuat developer menjadi lebih cepat dan efisien dalam menulis kode membuat program.

Ada beberapa istilah seperti:

- i. Superclass → mewariskan member di dalamnya. Biasanya disebut induk, base atau parent class.
- ii. SubClass → mewarisi member milik SuperClass. Class ini memiliki membernya sendiri selain yang diwariskan dari SuperClassnya.

b. Encapsulation

Encapsulation adalah membungkus class dan menjaga apa yang ada di dalam class tersebut, baik function ataupun property agar tidak dapat diakses oleh class lainnya. Dengan begini, data tersebut dapat digunakan tanpa harus tahu bagaimana proses yang terjadi sampai data tersebut bisa digunakan.

Untuk menjaga hal tersebut dalam Encapsulation dikenal sebuah istilah Hak Akses Modifier yang terdiri dari :

- Private
Memberikan hak akses hanya pada class itu sendiri, artinya member yang ada di dalam class A baik itu function ataupun property hanya bisa diakses oleh class A saja, class lain tidak bisa mengaksesnya.
- Public memberikan hak akses kepada function atau atribut agar bisa diakses oleh siapapun (property atau class lain diluar class yang bersangkutan), artinya method atau atribut yang ada di class A dapat diakses oleh siapa pun baik itu class A, class B dan class lainnya.
- Protected = memberikan hak akses kepada class itu sendiri dan class hasil turunannya (inheritance), artinya member yang ada di class A hanya bisa diakses oleh class A sendiri dan class yang menjadi SubClass dari class A. Namun harus dipahami class lain yang berada dalam satu package dengan class A mampu mengakses tipe data protected, Sedangkan yang tidak mampu mengakses adalah class-class yang berada diluar package class A. Untuk dapat mengaksesnya, class yang berada diluar package class A harus menjadi SubClass dari class A.

Selain itu, kode yang berada di dalam aplikasi diringkas berdasarkan bagusnya sehingga dapat dengan mudah diubah tanpa mempengaruhi

bagian lainnya. Kode yang sudah dienkapsulasi tersebut dapat digunakan kembali di seluruh bagian aplikasi.

Perlu ditekankan bahwa encapsulation menyebabkan data tersembunyi, bukan disembunyikan.

Perlu digaris bawahi bahwa encapsulation bukan menyembunyikan sebuah data. Tetapi, encapsulation yang menyebabkan data tersebut tersembunyi.

c. Abstraction

Abstraction adalah mekanisme saat proses dalam sebuah objek disembunyikan. Abstraction memungkinkan developer memerintahkan fungsi tanpa harus mengetahui bagaimana fungsi tersebut bekerja sehingga dapat menghindarkan developer dari kompleksitas sistem. Sekilas abstraction mirip dengan encapsulation. Namun, terdapat beberapa perbedaan di antaranya:

- Pada abstraction, fokusnya terhadap suatu objek tanpa perlu tahu bagaimana itu dilakukan. Sedangkan encapsulation menyembunyikan mekanismenya
- Encapsulation menyembunyikan data dengan menyediakan getter setter untuk mengaksesnya, sedangkan abstraction memanfaatkan abstract class, interface, dan lain sebagainya

Terdapat istilah yang bernama abstraction level yang merupakan mekanisme memisahkan dua kompleksitas sebuah sistem. Dalam proses komputasi, abstraction layer atau level merupakan cara menyembunyikan detail implementasi yang kompleks dari serangkaian fungsionalitas tertentu dengan tujuan agar dapat memisahkan masalah seperti interoperabilitas.

Contoh abstraction di dunia nyata adalah ketika ingin mengendarai motor kita hanya perlu menarik gas tanpa perlu tahu apa yang terjadi ketika gas tersebut ditarik.

d. Polymorphism

Polymorphism merupakan kemampuan objek, variabel, atau fungsi yang dapat memiliki berbagai bentuk. Memiliki berbagai bentuk bermaksud kita dapat menimpa (override), suatu method, yang berasal dari parent class (super class) dimana object tersebut diturunkan, sehingga memiliki kelakuan yang berbeda.

Polymorphism dapat dianalogikan sebuah laptop yang memiliki kemampuan sebagai alat bantu dalam bekerja, sebagai alat bantu untuk belajar dan juga sebagai alat bantu untuk sarana hiburan seperti bermain game. Laptop tersebut adalah sebuah class yang akan dapat di salin menjadi sebuah object sedangkan kemampuan sebagai alat bantu adalah polymorphism itu sendiri.

Pada dasarnya polymorphism terbagi menjadi dua jenis, yaitu :

- Compile time polymorphism adalah sebuah proses di mana sebuah method atau fungsi dipanggil saat kompilasi. Ini diakibatkan oleh sebuah konsep bernama method overloading. Method overloading merupakan kondisi di mana kita bisa membuat dua atau lebih fungsi

yang memiliki jumlah, tipe, dan urutan parameter yang berbeda di dalam sebuah class.

- Runtime polymorphism adalah proses di mana sebuah fungsi dipanggil pada saat runtime. Contoh dari runtime polymorphism adalah method overriding, yaitu sebuah kelas yang memiliki fungsi dengan nama sama dengan fungsi yang di dalam kelas induknya.

6. Kelebihan OOP :

- a. Dengan memanfaatkan inheritance, kode akan lebih ringkas, dan kelas yang ada dapat diperpanjang
- b. Mudah mempertahankan dan memodifikasi kode karena objek baru dapat dibuat dengan sedikit perbedaan
- c. Kerangka kerja yang baik untuk pustaka kode sehingga komponen software yang disediakan mudah diadaptasi dan dimodifikasi programmer
- d. Programmer dapat membangun program yang aman karena bisa melakukan penyembunyian data
- e. Sistem mudah ditingkatkan dari kecil ke besar
- f. Penggunaan kembali kode lebih mudah dibanding bahasa pemrograman konvensional

Kekurangan OOP :

- a. Eksekusi lambat karena ukuran program besar dibanding pemrograman prosedural konvensional
- b. Membuat dan mengelola kode akan lebih sulit bagi programmer yang kurang handal
- c. Merancang dan mengimplementasikan konsep object-oriented programming dengan benar cukup rumit dan memberatkan programmer.

7. Berikut merupakan source code untuk menghitung jarak antara 2 titik koordinat 3D

```
# Program
# Menghitung jarak antara 2 titik koordinat

# Kamus
# a, b, c, d, e, f : int
# koordinatAkhir : int

# Algoritma
print('Petunjuk: Masukkan angka koordinat 3D dengan separasi koma
dan spasi \n format: <int>, <int>, <int> \n contoh: 9, 3, 2')
x1, y1, z1 = [int(x) for x in input("Masukkan koordinat awal:
").split(", ")] # input koordinat awal
x2, y2, z2 = [int(x) for x in input("Masukkan koordinat akhir:
").split(", ")] # input koordinat akhir

jarak = ((x2-x1)**2 + (y2-y1)**2 + (z2-z1)**2)**0.5 #
menghitung jarak dengan rumus matematika #

((x2-x1)**2 + (y2-y1)**2 + (z2-z1)**2)**0.5

print("%.2f" % jarak) # mengeluarkan output jarak dengan maksimal 2
```

digit desimal

Screenshot pada VS Code :

```
JarakKoordinat.py X
THT Final > JarakKoordinat.py > ...
1 # Program
2 # Menghitung jarak antara 2 titik koordinat
3
4 # Kamus
5 # a, b, c, d, e, f : int
6 # koordinatAkhir : int
7
8 # Algoritma
9 print('Petunjuk: Masukkan angka koordinat 3D dengan separasi koma dan spasi \n format: <int>, <int>, <int> \n contoh: 9, 3, 2')
10 x1, y1, z1 = [int(x) for x in input("Masukkan koordinat awal: ").split(", ")] # input koordinat awal
11 x2, y2, z2 = [int(x) for x in input("Masukkan koordinat akhir: ").split(", ")] # input koordinat akhir
12
13 jarak = ((x2-x1)**2 + (y2-y1)**2 + (z2-z1)**2)**0.5 # menghitung jarak dengan rumus matematika
14 # ((x2-x1)**2 + (y2-y1)**2 + (z2-z1)**2)**0.5
15
16 print("%.2f" % jarak) # mengeluarkan output jarak dengan maksimal 2 digit desimal
```

Output yang dihasilkan :

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Python + - [ ] [X]
ologi Bandung/Kegiatan/Aksantara/Oprec/THT/Coding/Jarak_Koordinat.py"
Petunjuk: Masukkan angka koordinat 3D dengan separasi koma dan spasi
format: <int>, <int>, <int>
contoh: 9, 3, 2
Masukkan koordinat awal: 9, 3, 2
Masukkan koordinat akhir: 3, 7, 4
7.48
```

SOAL WAJIB TERKAIT KEHADIRAN DI BANDUNG

Di Aksantara, Anda akan bekerja sama dengan anggota lainnya pada satu tim lomba. Ketika sudah masuk ke dalam tim lomba masing-masing, kehadiran Anda di Bandung dibutuhkan untuk melakukan kegiatan bersama tim secara langsung pada bulan Juni s.d. Oktober. Kehadiran di Bandung akan berpengaruh pada performa tim sampai pembagian beban kerja setiap anggota tim, sehingga kehadiran Anda di Bandung akan sangat diharapkan.

1. Apakah Anda dapat memenuhi kehadiran di Bandung tersebut?
2. Untuk mempermudah kehadiran Anda di Bandung, hal-hal apa saja yang Anda harap Aksantara dapat berikan/fasilitasi? (seperti fasilitas, surat keterangan lomba, rekomendasi tempat tinggal, obat-obatan, SOP kesehatan, dll).
 - a. Jika ya, jelaskan secara lengkap rencana yang akan Anda lakukan untuk dapat hadir di Bandung, seperti izin orang tua, akomodasi tempat tinggal, transportasi, dll.
 - b. Jika tidak, jelaskan secara lengkap kondisi, kekhawatiran, keraguan, hingga alasan yang membuat Anda sulit untuk memenuhi kehadiran di Bandung.

Jawaban Soal Wajib Terkait Kehadiran di Bandung

1. Ya, dapat.

2. Rencana saya akan selalu hadir sesuai dengan kebutuhan tim. Mengenai izin orang tua dan akomodasi akan mudah karena memiliki saudara dekat di Bandung. Transportasi juga aman karena ada motor.

Lampiran

[Link GitHub](#)

Referensi

- Falahqie, F., 2020. *Konsep Dasar Object Oriented Programming*. [online] Blog Sekolah Mobile. Available at: <https://blog.sekolahmobile.com/konsep-dasar-object-oriented-programming/#Konsep_pada_OOP> [Accessed 17 February 2022].
- Monitor Teknologi. 2020. Apa Itu OOP (Object-Oriented Programming)? - Monitor Teknologi. [online] Available at: <<https://www.monitorteknologi.com/apa-itu-oop-object-oriented-programming/#kelebihan-dan-kekurangan-oop>> [Accessed 17 February 2022].
- Python Enthusiast. (2020). Retrieved 19 February 2022, from <https://www.youtube.com/watch?v=IOhZqmSrjIE&t=228s>
- RoboJackets Training. (2020). Retrieved 19 February 2022, from <https://www.youtube.com/watch?v=o90laCRje2I&t=296s>
- Behl, M., 2022. *F/10 Autonomous Racing*. [online] Linklab-uva.github.io. Available at: <<https://linklab-uva.github.io/autonomusracing/assets/files/ros-turtlesim.pdf>> [Accessed 19 February 2022].