

PP/018/V/R4



LABORATORIUM
TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS AHMAD DAHLAN



PETUNJUK PRAKTIKUM

EDISI KURIKULUM OBE

ROBOTIKA INFORMATIKA

Penyusun:

Nuryono Satya Widodo, S.T., M. Eng.

Faisal Fajri Rahani S.Si., M.Cs.

2022

PETUNJUK PRAKTIKUM NAMA PRAKTIKUM

Copyright© 2022,

Nuryono Satya Widodo, S.T., M. Eng.

Faisal Fajri Rahani S.Si., M.Cs.

Hak Cipta dilindungi Undang-Undang

Dilarang mengutip, memperbanyak atau mengedarkan isi buku ini, baik sebagian maupun seluruhnya, dalam bentuk apapun, tanpa izin tertulis dari pemilik hak cipta dan penerbit.

Diterbitkan oleh:

Program Studi Teknik Informatika

Fakultas Teknologi Industri

Universitas Ahmad Dahlan

Jalan Ring Road Selatan, Tamanan, Banguntapan, Bantul Yogyakarta 55166

Penulis : Nuryono Satya Widodo, S.T., M. Eng.
Faisal Fajri Rahani S.Si., M.Cs.

Editor : Laboratorium Teknik Informatika, Universitas Ahmad Dahlan

Desain sampul : Laboratorium Teknik Informatika, Universitas Ahmad Dahlan

Tata letak : Laboratorium Teknik Informatika, Universitas Ahmad Dahlan

Ukuran/Halaman : 21 x 29,7 cm / 117 halaman

Didistribusikan oleh:



Laboratorium Teknik Informatika

Universitas Ahmad Dahlan

Jalan Ring Road Selatan, Tamanan, Banguntapan, Bantul Yogyakarta 55166

Indonesia



KATA PENGANTAR

Alhamdulillahirabbil'alamin, atas hidayah dari Allah SWT, Petunjuk Praktikum ini dapat diselesaikan sesuai rencana yang ditentukan. Petunjuk Praktikum ini ditulis dengan tujuan membantu mahasiswa dalam mempelajari materi yang berkaitan dengan mata kuliah Robotika Informatika.

Bahasan utama pada Petunjuk Praktikum ini mendasarkan pada telaah tentang robot AdroiT yang digunakan pada praktikum Robotika Informatika.

Melalui Petunjuk Praktikum ini penulis berharap dapat membantu mahasiswa dalam memahami dan menguasai mata kuliah Robotika Informatika.

Sekalipun telah diupayakan secara maksimal agar menjadi karya yang baik, namun disadari sepenuhnya bahwa Petunjuk Praktikum ini masih mengandung kekurangan, untuk itu saran dari semua pihak sangat dinanti kedatangannya. Akhirnya penulis berharap semoga Petunjuk Pratikum ini dapat digunakan sebagai bahan pertimbangan dalam usaha-usaha peningkatan kualitas pendidikan teknik di Indonesia.

Yogyakarta, 17 Agustus 2022

Penyusun

DAFTAR PENYUSUN

Nuryono Satya Widodo, S.T., M.Eng.

Foto Penyusun 1	Tuliskan biografi penyusun 1 disini
-----------------	-------------------------------------

Faisal Fajri Rahani, S.Si., M.Cs.

Foto Penyusun 2	Tuliskan biografi penyusun 2 disini
-----------------	-------------------------------------

HALAMAN REVISI

Yang bertanda tangan di bawah ini:

Nama : Faisal Fajri Rahani, S.Si., M.Cs.

NIY : 60191224

Jabatan : Dosen Pengampu Mata Kuliah Robotika Informatika

Dengan ini menyatakan pelaksanaan Revisi Petunjuk Praktikum Robotika Informatika untuk Program Studi Teknik Informatika telah dilaksanakan dengan penjelasan sebagai berikut:

No	Keterangan Revisi	Tanggal Revisi	Nomor Modul
1	a. Perubahan Ilustrasi pada Petunjuk Umum, Unit 1 dan Unit 2. b. Perubahan Tugas Akhir pada Unit 1, Unit 2, Unit 3, Unit 4.	4 Maret 2019	PP/019/I/1
2	Perubahan mengikuti format baru	16 Februari 2020	PP/018/V/R2
3	a. Perubahan mengikuti format baru b. Penambahan materi unit 6	17 Agustus 2021	PP/018/V/R3
4	Penambahan materi unit 7, 8, 9, dan 10	17 Agustus 2022	PP/018/V/R4

Yogyakarta, 17 Agustus 2022

Penyusun



Faisal Fajri Rahani, S.Si., M.Cs.

NIY. 60191224

HALAMAN PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama : Lisna Zahrotun, S.T., M.Cs.

NIK/NIY : 60150773

Jabatan : Kepala Laboratorium Teknik Informatika

Menerangkan dengan sesungguhnya bahwa Petunjuk Praktikum ini telah direview dan akan digunakan untuk pelaksanaan praktikum di Semester Gasal Tahun Akademik 2020/2021 di Laboratorium Praktikum Teknik Informatika, Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Ahmad Dahlan.

Yogyakarta, 1 Agustus 2021

Mengetahui,

Ketua Kelompok Keilmuan Sistem Cerdas

Kepala Laboratorium Teknik Informatika



Dewi Soyusiawaty, S.T., M.T.

NIY. 60040497



Lisna Zahrotun, S.T., M.Cs.

NIY. 60150773

VISI DAN MISI PRODI TEKNIK INFORMATIKA

VISI

Menjadi Program Studi Informatika yang diakui secara internasional dan unggul dalam bidang Informatika serta berbasis nilai-nilai Islam.

MISI

1. Menjalankan pendidikan sesuai dengan kompetensi bidang Informatika yang diakui nasional dan internasional
2. Meningkatkan penelitian dosen dan mahasiswa dalam bidang Informatika yang kreatif, inovatif dan tepat guna.
3. Meningkatkan kuantitas dan kualitas publikasi ilmiah tingkat nasional dan internasional
4. Melaksanakan dan meningkatkan kegiatan pengabdian masyarakat oleh dosen dan mahasiswa dalam bidang Informatika.
5. Menyelenggarakan aktivitas yang mendukung pengembangan program studi dengan melibatkan dosen dan mahasiswa.
6. Menyelenggarakan kerja sama dengan lembaga tingkat nasional dan internasional.
7. Menciptakan kehidupan Islami di lingkungan program studi.

TATA TERTIB LABORATORIUM TEKNIK INFORMATIKA

1.1. DOSEN/KOORDINATOR PRAKTIKUM

1. Dosen harus hadir saat praktikum minimal 15 menit di awal kegiatan praktikum untuk mengisi materi dan menandatangani presensi kehadiran praktikum.
2. Dosen membuat modul praktikum, soal seleksi asisten, pre-test, post-test, dan responsi dengan berkoordinasi dengan asisten dan pengampu mata praktikum.
3. Dosen berkoordinasi dengan koordinator asisten praktikum untuk evaluasi praktikum setiap minggu.
4. Dosen menandatangani surat kontrak asisten praktikum dan koordinator asisten praktikum.
5. Dosen yang tidak hadir pada slot praktikum tertentu tanpa pemberitahuan selama 2 minggu berturut-turut mendapat teguran dari Kepala Laboratorium, apabila masih berlanjut 2 minggu berikutnya maka Kepala Laboratorium berhak mengganti koordinator praktikum pada slot tersebut.

1.2. PRAKTIKAN

1. Praktikan harus hadir 15 menit sebelum kegiatan praktikum dimulai, dan dispensasi terlambat 15 menit dengan alasan yang jelas (kecuali asisten menentukan lain dan patokan jam adalah jam yang ada di Laboratorium, terlambat lebih dari 15 menit tidak boleh masuk praktikum & dianggap Inhal).
2. Praktikan yang tidak mengikuti praktikum dengan alasan apapun, wajib mengikuti INHAL, maksimal 4 kali praktikum dan jika lebih dari 4 kali maka praktikum dianggap GAGAL.
3. Praktikan harus berpakaian rapi sesuai dengan ketentuan Universitas, sebagai berikut:
 - a. Tidak boleh memakai Kaos Oblong, termasuk bila ditutupi Jaket/Jas Almamater (Laki-laki / Perempuan) dan Topi harus Dilepas.
 - b. Tidak Boleh memakai Baju ketat, Jilbab Minim dan rambut harus tertutup jilbab secara sempurna, tidak boleh kelihatan di jidat maupun di punggung (khusus Perempuan).
 - c. Tidak boleh memakai baju minim, saat duduk pun pinggang harus tertutup rapat (Laki-laki / Perempuan).
 - d. Laki-laki tidak boleh memakai gelang, anting-anting ataupun aksesoris Perempuan.
4. Praktikan tidak boleh makan dan minum selama kegiatan praktikum berlangsung, harus menjaga kebersihan, keamanan dan ketertiban selama mengikuti kegiatan praktikum atau selama berada di dalam laboratorium (tidak boleh membuang sampah sembarangan baik kertas, potongan kertas, bungkus permen baik di lantai karpet maupun di dalam ruang CPU).
5. Praktikan dilarang meninggalkan kegiatan praktikum tanpa seizin Asisten atau Laboran.
6. Praktikan harus meletakkan sepatu dan tas pada rak/loker yang telah disediakan.
7. Selama praktikum dilarang NGENET/NGE-GAME, kecuali mata praktikum yang membutuhkan atau menggunakan fasilitas Internet.
8. Praktikan dilarang melepas kabel jaringan atau kabel power praktikum tanpa sepengetahuan laboran
9. Praktikan harus memiliki FILE Petunjuk praktikum dan digunakan pada saat praktikum dan harus siap sebelum praktikum berlangsung.
10. Praktikan dilarang melakukan kecurangan seperti mencontek atau menyalin pekerjaan praktikan yang lain saat praktikum berlangsung atau post-test yang menjadi tugas praktikum.


11. Praktikan dilarang mengubah setting software/hardware komputer baik menambah atau mengurangi tanpa permintaan asisten atau laboran dan melakukan sesuatu yang dapat merugikan laboratorium atau praktikum lain.
12. Asisten, Koordinator Praktikum, Kepala laboratorium dan Laboran mempunyai hak untuk menegur, memperingatkan bahkan meminta praktikan keluar ruang praktikum apabila dirasa anda mengganggu praktikan lain atau tidak melaksanakan kegiatan praktikum sebagaimana mestinya dan atau tidak mematuhi aturan lab yang berlaku.
13. Pelanggaran terhadap salah satu atau lebih dari aturan diatas maka Nilai praktikum pada pertemuan tersebut dianggap 0 (NOL) dengan status INHAL.

1.3. ASISTEN PRAKTIKUM

1. Asisten harus hadir 15 Menit sebelum praktikum dimulai (konfirmasi ke koordinator bila mengalami keterlambatan atau berhalangan hadir).
2. Asisten yang tidak bisa hadir WAJIB mencari pengganti, dan melaporkan kepada Koordinator Asisten.
3. Asisten harus berpakaian rapi sesuai dengan ketentuan Universitas, sebagai berikut:
 - a. Tidak boleh memakai Kaos Oblong, termasuk bila ditutupi Jaket/Jas Almamater (Laki-laki / Perempuan) dan Topi harus Dilepas.
 - b. Tidak Boleh memakai Baju ketat, Jilbab Minim dan rambut harus tertutup jilbab secara sempurna, tidak boleh kelihatan di jidat maupun di punggung (khusus Perempuan).
 - c. Tidak boleh memakai baju minim, saat duduk pun pinggang harus tertutup rapat (Laki-laki / Perempuan).
 - d. Laki-laki tidak boleh memakai gelang, anting-anting ataupun aksesoris Perempuan.
4. Asisten harus menjaga kebersihan, keamanan dan ketertiban selama mengikuti kegiatan praktikum atau selama berada di laboratorium, menegur atau mengingatkan jika ada praktikan yang tidak dapat menjaga kebersihan, ketertiban atau kesopanan.
5. Asisten harus dapat merapikan dan mengamankan presensi praktikum, Kartu Nilai serta tertib dalam memasukan/Input nilai secara Online/Offline.
6. Asisten harus dapat bertindak secara profesional sebagai seorang asisten praktikum dan dapat menjadi teladan bagi praktikan.
7. Asisten harus dapat memberikan penjelasan/pemahaman yang dibutuhkan oleh praktikan berkenaan dengan materi praktikum yang diasistensi sehingga praktikan dapat melaksanakan dan mengerjakan tugas praktikum dengan baik dan jelas.
8. Asisten tidak diperkenankan mengobrol sendiri apalagi sampai membuat gaduh.
9. Asisten dimohon mengkoordinasikan untuk meminta praktikan agar mematikan komputer untuk jadwal terakhir dan sudah dilakukan penilaian terhadap hasil kerja praktikan.
10. Asisten wajib untuk mematikan LCD Projector dan komputer asisten/praktikan apabila tidak digunakan.
11. Asisten tidak diperkenankan menggunakan akses internet selain untuk kegiatan praktikum, seperti Youtube/Game/Medsos/Streaming Film di komputer praktikan.

1.4. LAIN-LAIN

1. Pada Saat Responsi Harus menggunakan Baju Kemeja untuk Laki-laki dan Perempuan untuk Praktikan dan Asisten.
2. Ketidakhadiran praktikum dengan alasan apapun dianggap INHAL.
3. Izin praktikum mengikuti aturan izin SIMERU/KULIAH.
4. Yang tidak berkepentingan dengan praktikum dilarang mengganggu praktikan atau membuat keributan/kegaduhan.

- 
5. Penggunaan lab diluar jam praktikum maksimal sampai pukul 21.00 dengan menunjukkan surat ijin dari Kepala Laboratorium Prodi Teknik Informatika.

Yogyakarta, 1 Agustus 2021

Kepala Laboratorium Teknik Informatika

Lisna Zahrotun, S.T., M.Cs.

NIY. 60150773



DAFTAR ISI

HAK CIPTA	1
KATA PENGANTAR.....	2
DAFTAR PENYUSUN.....	3
HALAMAN REVISI.....	4
HALAMAN PERNYATAAN.....	5
VISI DAN MISI PRODI TEKNIK INFORMATIKA	6
TATA TERTIB LABORATORIUM TEKNIK INFORMATIKA.....	7
DAFTAR ISI	10
DAFTAR GAMBAR	11
DAFTAR TABEL.....	12
SKENARIO PRAKTIKUM SECARA DARING	13
PRAKTIKUM 1: PERSIAPAN DAN PENGATURAN SYSTEM (UBUNTU)	14
PRAKTIKUM 2: PENGENALAN ROBOT OPERATING SYSTEM	27
PRAKTIKUM 3: KONSEP NODE, MESSAGE, DAN TOPIC PADA ROBOT OPERATING SYSTEM.....	37
PRAKTIKUM 4: KONSEP SERVICE PADA ROBOT OPERATING SYSTEM.....	49
PRAKTIKUM 5: KONSEP PACKAGE PADA ROBOT OPERATING SYSTEM.....	58
PRAKTIKUM 6: RVIZ PADA ROBOT OPERATING SYSTEM.....	69
PRAKTIKUM 7: PENGENALAN SIMULASI MIKROKONTROLLER.....	79
PRAKTIKUM 8: PENGENALAN SIMULASI STEPPER MOTOR.....	88
PRAKTIKUM 9: PENGENALAN SIMULASI MPU6050	99
PRAKTIKUM 10: PENGENALAN PENGGERAK SERVO	107
DAFTAR PUSTAKA.....	115

DAFTAR GAMBAR

DAFTAR TABEL



SKENARIO PRAKTIKUM SECARA DARING

Nama Mata Praktikum :

Jumlah Pertemuan :

TABEL SKENARIO PRAKTIKUM DARING

Pertemuan ke	Judul Materi	Waktu (Lama praktikum sampai pengumpulan posttest)	Skenario Praktikum dari pemberian pre-test, post-test dan pengumpulannya serta mencantumkan metode yang digunakan misal video, whatsapp group, Google meet atau lainnya
1	Persiapan Dan Pengaturan Sistem (Ubuntu)	1 minggu	Pre-test dan post-test diberikan dan dikumpulkan melalui e-learning (E-Learning UAD atau Google Classroom) sesuai waktu yang ditentukan.
2	Pengenalan Robot Operating System	1 minggu	Praktikum dilakukan menggunakan video tutorial dan whatsapp/forum di elearning untuk bertanya.
3	Konsep Node, Message, dan Topic Pada Robot Operating System	1 minggu	Pre-test diberikan di awal praktikum selama 15 menit, keterlambatan diberi waktu sampai waktu praktikum berakhir.
4	Konsep Service Pada Robot Operating System	1 minggu	Post-test dan laporan praktikum dikerjakan selama 1 minggu sampai praktikum berikutnya dimulai.
5	Konsep Package Pada Robot Operating System	1 minggu	Laporan praktikum berisi: Cover, Pendahuluan (berisi materi singkat ttg post test yang dikerjakan), Pre-test (berisi jawaban pre-test yang ada di modul praktikum), Praktikum (berisi listing kode jawaban post-test disertai penjelasannya dan screenshot hasil post test) dan Pembahasan (pembahasan singkat dari hasil post test)
6	Rviz Pada Robot Operating System	1 minggu	

PRAKTIKUM 1: PERSIAPAN DAN PENGATURAN SYSTEM (UBUNTU)

Pertemuan ke : 1

Total Alokasi Waktu : 90 menit

- Materi : 15 menit
- Pre-Test : 15 menit
- Praktikum : 45 menit
- Post-Test : 15 menit

Total Bobot Penilaian : 100%

- Pre-Test : 0 %
- Praktik : 100 %
- Post-Test : 0 %

Pemenuhan CPL dan CPMK:

CPL-08	Mampu memilih, membuat dan menerapkan teknik, sumber daya, penggunaan perangkat teknik modern dan implementasi teknologi informasi untuk memecahkan masalah
CPMK-05	Mampu mengimplementasikan low level control pada robot dan pemrogramannya, High Level control pada robot dan pemrogramannya, Pemanfaatan mesin Visi dan sistem cerdas pada Robot

1.1. DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

1. Menjelaskan bagaimana cara menggunakan ros pada virtual komputer maupun boot melalui device lain.

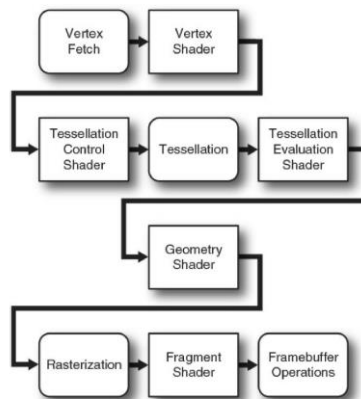
1.2. INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-08	CPMK-05	Mampu membangun dan mempresentasikan sistem robot
--------	---------	---

1.3. TEORI PENDUKUNG

Tuliskan teori pendukung disini. Contoh penulisan Gambar 1.1.



Gambar 1.1 Label Gambar.

1.4. HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

1. Komputer.
2. Sistem Operasi Ubuntu
3. Robotic Operating System

1.5. PRE-TEST

Jawablah pertanyaan berikut (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Skor
1.	-	-	-	

1.6. LANGKAH PRAKTIKUM

Persiapan Boot Media

Seluruh pelaksanaan praktikum robotika akan dilaksanakan menggunakan Robot Operating System yang terpasang pada System dengan Sistem Operasi Ubuntu. Peserta dapat memilih dua opsi dalam pengerjaan praktikum. Opsi pertama menggunakan USB bootable dan opsi kedua menggunakan virtual machine.

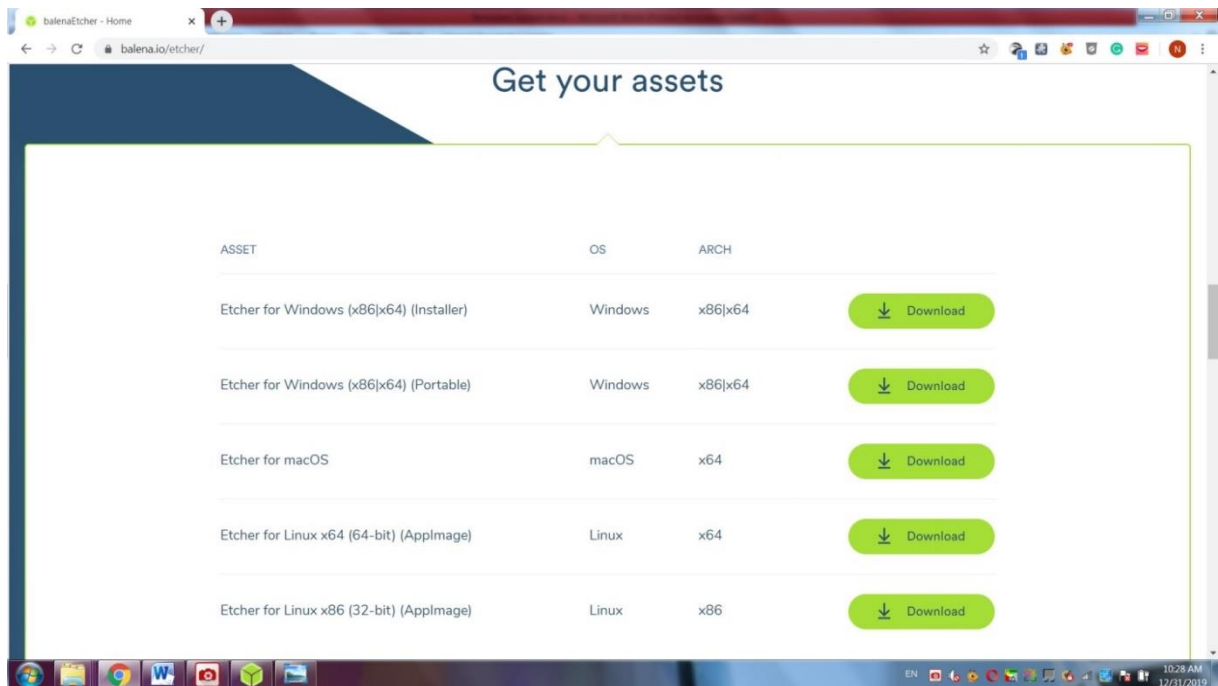
Persiapan Boot Media (Menggunakan Flash Disk)

Seluruh pelaksanaan praktikum robotika akan dilaksanakan menggunakan Robot Operating System yang terpasang pada System dengan Sistem Operasi Ubuntu. Peserta praktikum wajib menyiapkan:

1. Laptop dengan sekurangnya satu port USB 3.0 atau satu slot SD Card/microSD
2. Boot Media yang dapat berupa:
 - a. SSD SATA yang dilengkapi kabel USB 3.0 to SATA
 - b. Flash Disk 32GB USB 3.0
 - c. MicroSD Class 10 atau UHS-1 32GB

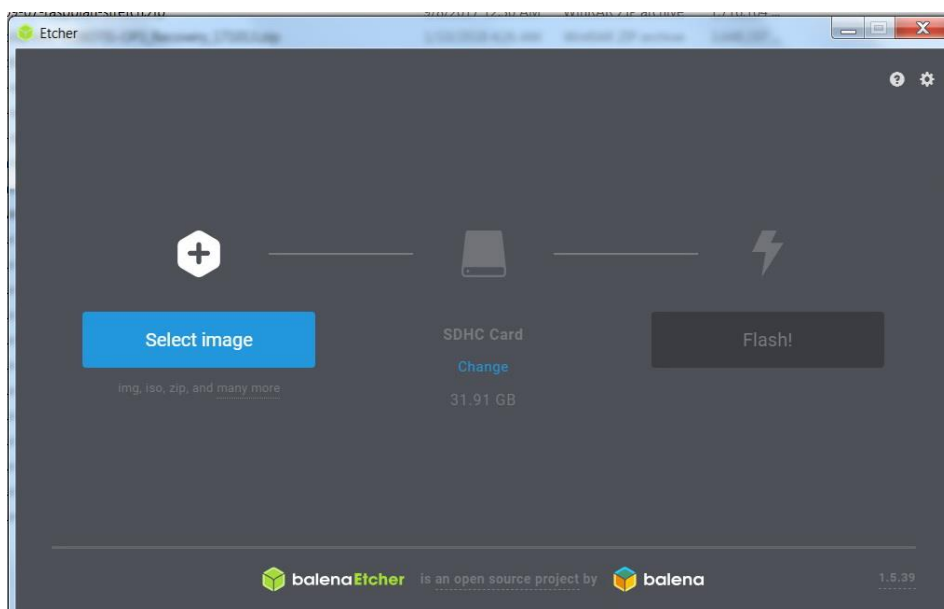
Tujuan dari unit praktikum ini adalah mengarahkan peserta praktikum untuk menyiapkan sistem yang diperlukan untuk praktikum dengan menggunakan Robot Operating System. Untuk memudahkan peserta, telah disediakan image OS yang sesuai(clone OS Ubuntu yang sudah diinstall ROS). File ini selanjutnya perlu untuk dipasangkan/diflash pada Boot Media agar selanjutnya

peserta melakukan booting dari Boot Media dengan OS Ubuntu yang dilengkapi ROS ini. Software yang diperlukan untuk melakukan proses ini adalah Balena etcher (<https://www.balena.io/etcher/>). Download dan install versi Balena etcher yang sesuai dengan Laptop yang anda gunakan (Windows).

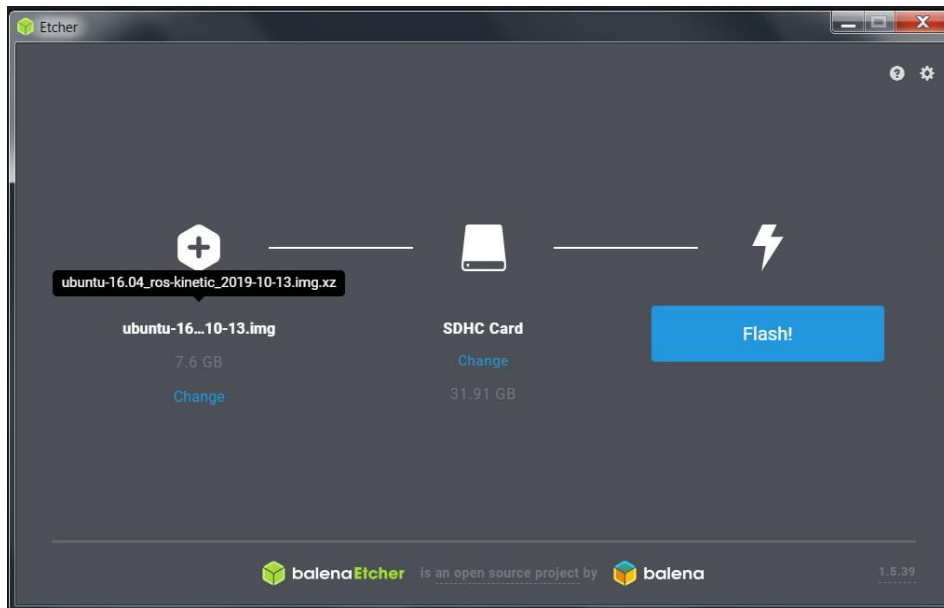


Selanjutnya pastikan bahwa image OS yang akan digunakan sudah berada pada Storage Internal Laptop anda. File Image yang akan digunakan memiliki nama file ubuntu-18.04_ros-melodic_2019-10-15.img.xz

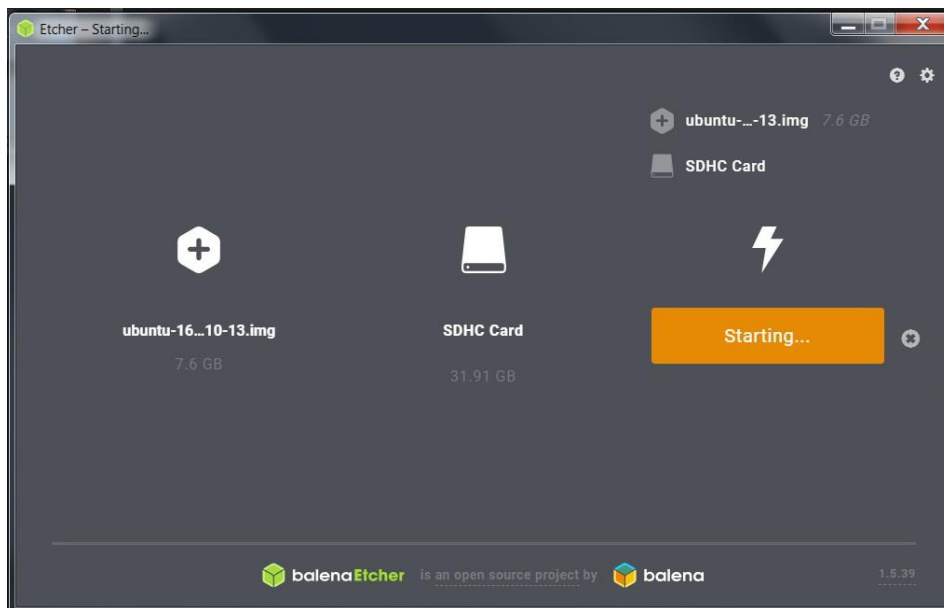
Pasang USB Flash Disk 3.0 pada port USB 3.0 atau pasang MicroSD pada slot yang tersedia. Buka software Balena Etcher, maka akan anda peroleh tampilan sebagai berikut



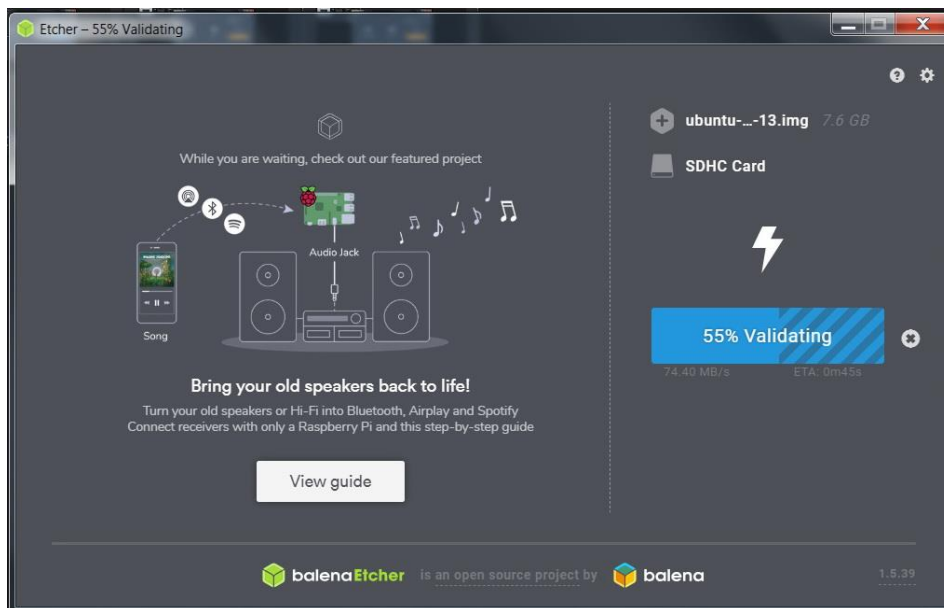
Pilih file image OS yang akan anda flash. Setelah dipilih tampilan akan berubah menjadi sebagai berikut:



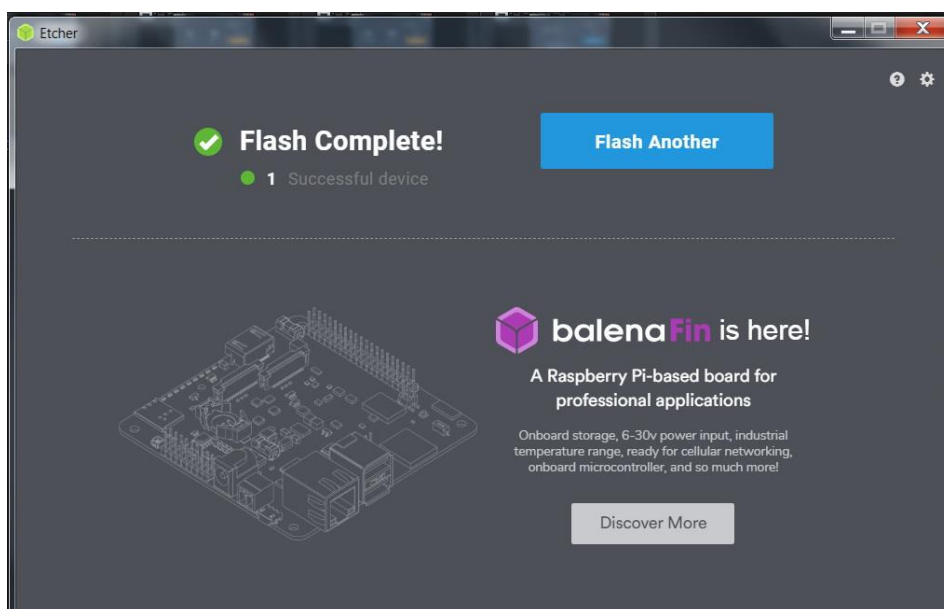
Pastikan removable storage yang akan diflash sudah sesuai. Jika semuanya sudah sesuai klik tombol Flash!. Jika muncul pop up dialog pilih Yes. Proses flash akan dimulai yang ditunjukkan dengan tampilan berikut



Pada Balena Etcher proses flashing secara otomatis diikuti proses validasi seperti tampak pada gambar berikut:



Tunggu hingga proses selesai yang ditunjukkan tampilan berikut



Setelah proses Flashing selesai matikan Laptop anda, nyalakan lagi laptop dengan FlashDisk atau microSD tetap terpasang dan atur agar masuk ke BIOS saat booting, atur agar FlashDisk atau removable media menjadi Boot Media pertama. Ikuti dan perhatikan panduan pengguna sesuai laptop anda masing-masing untuk pengaturan urutan Boot Media.

Persiapan Boot Media (Menggunakan Virtual BOX)

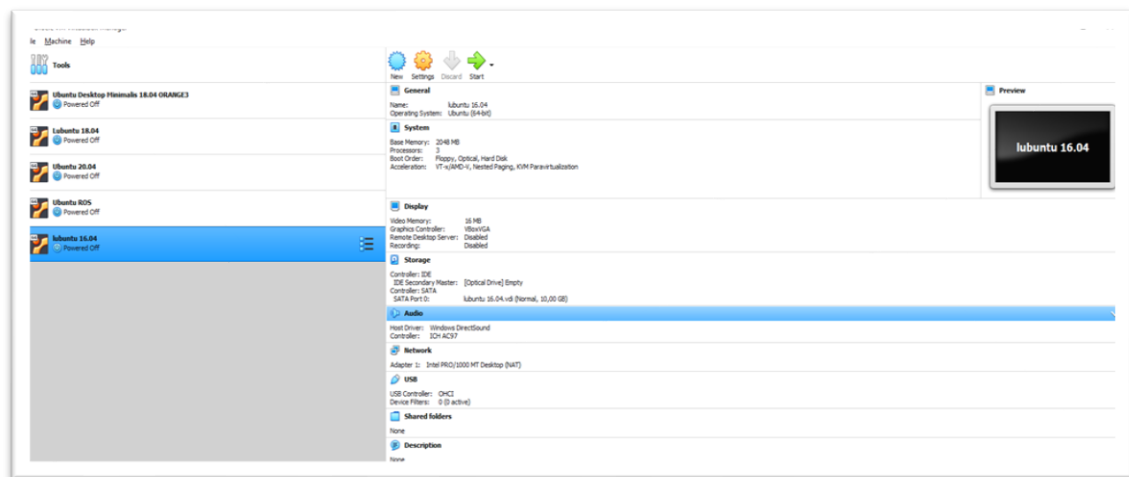
Tahapan dalam penggunaan Virtual Box untuk praktikum robotika dibagi menjadi 2 bagian. Pada bagian pertama adalah mengubah file format *.img.xz menjadi *.vdi. Pada bagian kedua adalah memasukkan *image* tersebut dalam virtual box.

Bagian 1 (Mengubah jenis file)

1. Unduh virtual box (link) dan install.
2. Ekstrak file *.img.xz sehingga terdapat file *.img

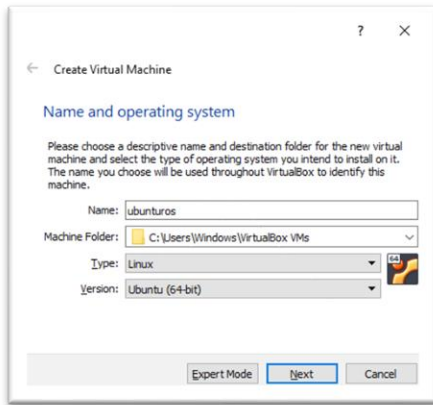
3. Copy file hasil ekstrak pada tempat instalasi vmware
 - a. Contoh “C:\Program Files\Oracle\VirtualBox”
4. Buka commad prompt dengan “run as admistrator”
5. Jalankan perintah
 - a. cd C:\Program Files\Oracle\VirtualBox
 - b. vboxmanage convertdd *.img *.vdi
6. Wajib Pindahkan file *.vdi hasil ekstrak ketempat yang dipilih.

Bagian 2 (Aplikasi pada virtual box)

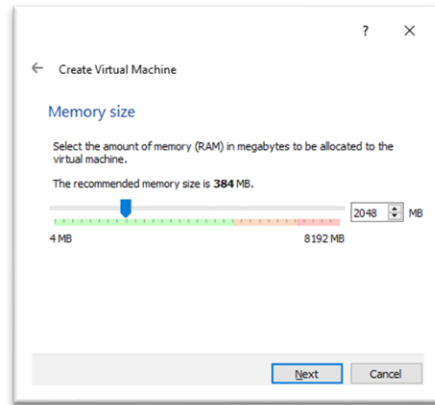


Gambar 1 Tampilan Virtual Box

1. Buka VirtualBox seperti terlihat pada Gambar 1.
2. Buat Virtual Machine baru dengan klik “new” akan terlihat Gambar 2.
 - a. Isikan nama sistem yang digunakan.
 - b. Ubah type “Linux” karena menggunakan OS linux.
 - c. Pilih versi ubuntu (64-bit) sesuai sistem yang akan digunakan.
 - d. Klik “next”.

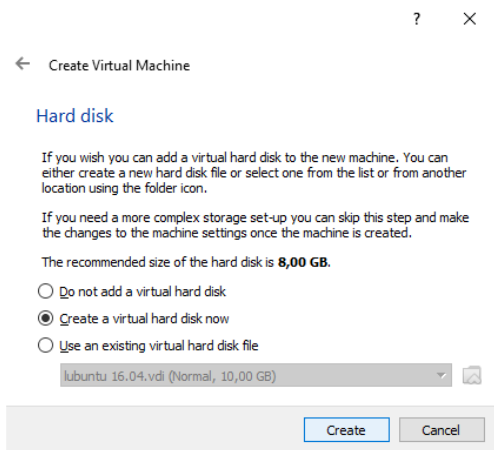


Gambar 2 Membuat Virtual Machine

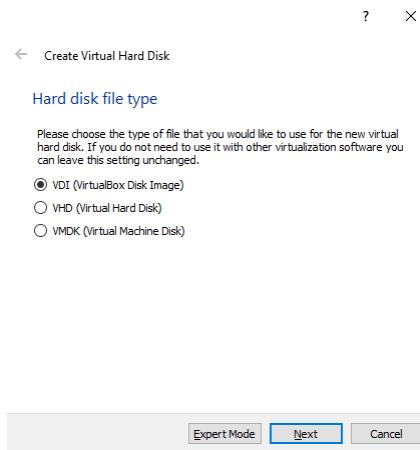


Gambar 3 Atur RAM sistem

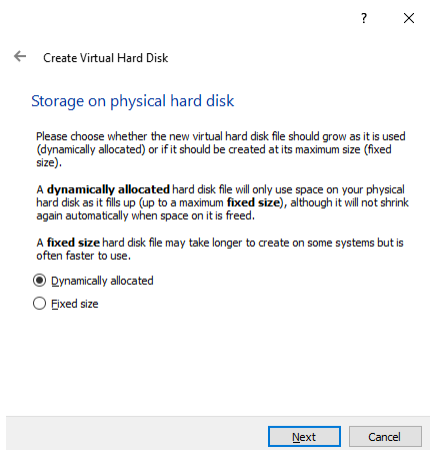
3. Sesuaikan ukuran memori RAM yang akan digunakan pada sistem virtual seperti pada Gambar 3. Penggunaan RAM minimal yang dibutuhkan sistem ini adalah 2 GB atau 2048 MB. Penggunaan RAM lebih besar akan lebih baik, namun juga disesuaikan dengan kapasitas RAM fisik pada komputer masing- masing.
4. Membuat virtual hardisk. Pada mesin virtual ini, kita juga diharuskan mengalokasikan ruang penyimpanan untuk sistem.
 - a. Silahkan pilih “create a virtual hardisk now” seperti Gambar 4.
 - b. Pilih hardisk type vdi seperti Gambar 5.
 - c. Pilih sifat penyimpanan dynamic seperti Gambar 6.
 - d. Atur alokasi penyimpanan, sesuaikan dengan kapasitas penyimpanan masing- masing. Atur seperti pada Gambar 7.



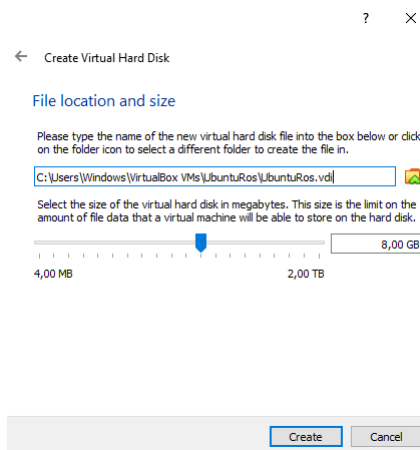
Gambar 4 Membuat *virtual hardisk*



Gambar 5 Memilih tipe penyimpanan

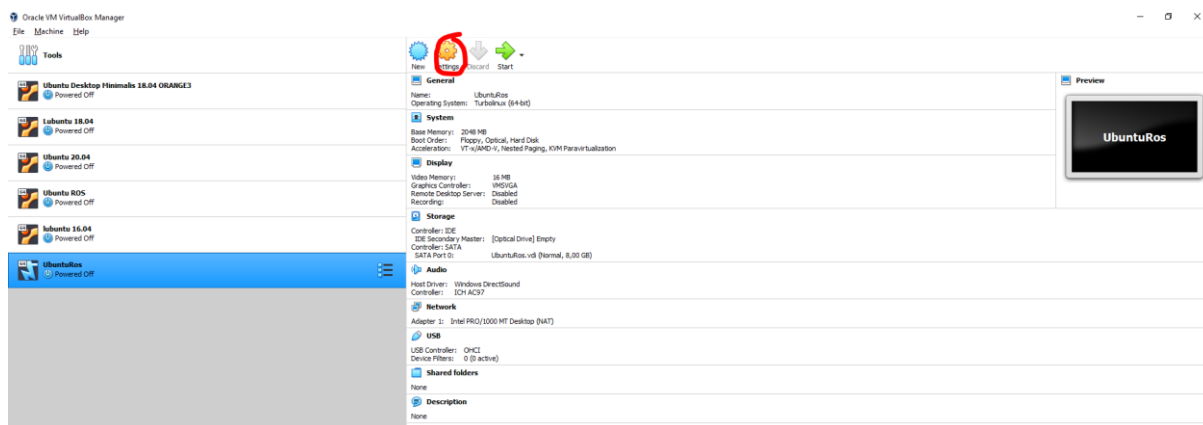


Gambar 6 Memilih sifat penyimpanan



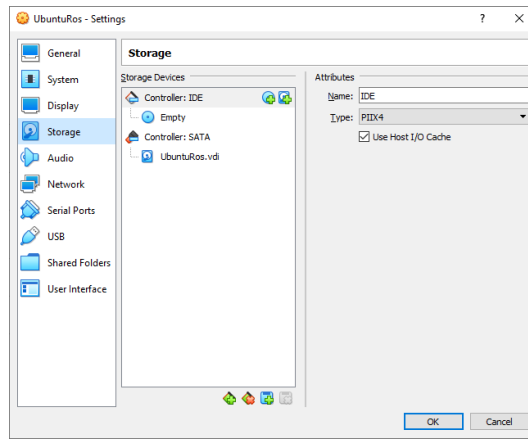
Gambar 7 Mengatur alokasi penyimpanan

5. Jika sudah, pilih virtual machine yang dibuat dan klik setting diatas. Seperti pada Gambar 8.



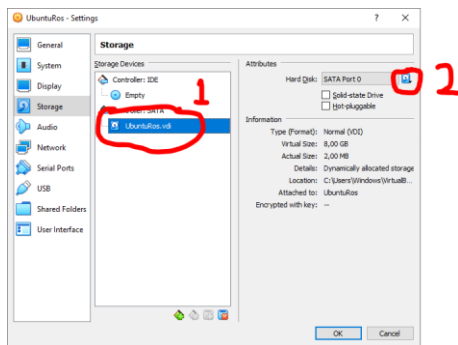
Gambar 8 Menu setting

6. Pilih menu “storage” seperti pada Gambar 9.

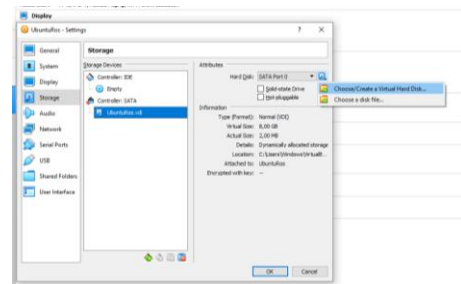


Gambar 9 Menu penyimpanan

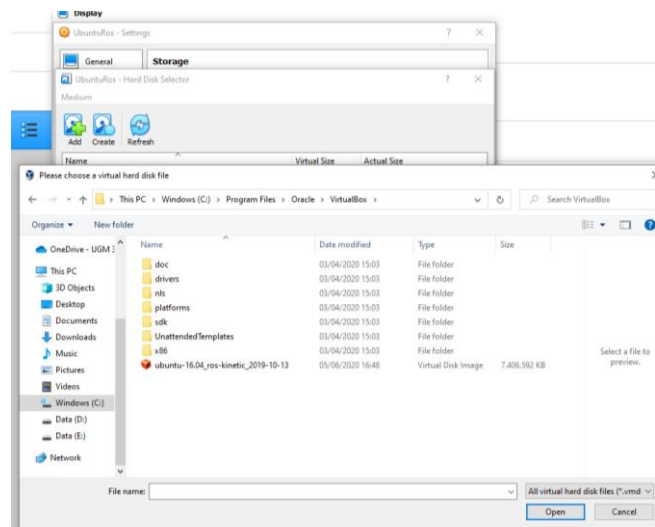
7. Pilih “namavirtual.vdi” Pilih file .vdi yang tadi di ekstrak.
 - a. Klik menu pada Gambar 10.
 - b. Pilih create virtual hardisk seperti pada Gambar 11.
 - c. Pilih file yang telah diekstrak tadi seperti pada Gambar 12.



Gambar 10 Klik *.vdi

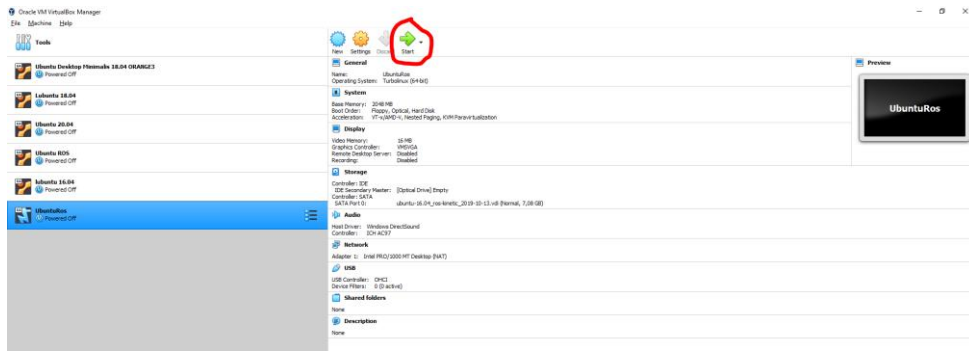


Gambar 11 pilih dari hardisk



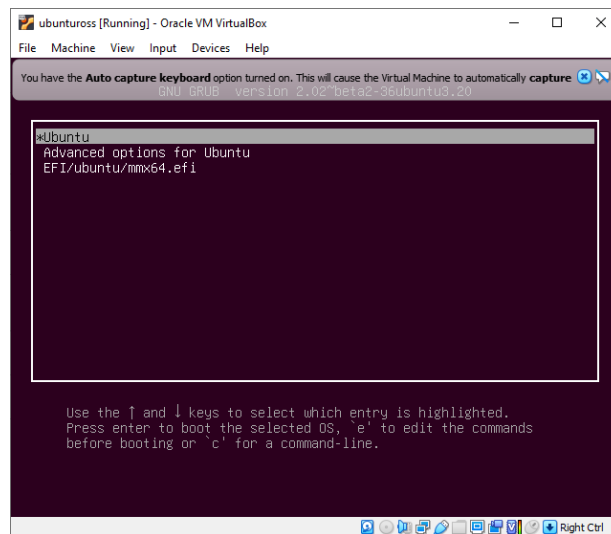
Gambar 12 Pilih file *.vdi yang telah di ekstrak sebelumnya

8. Jalankan virtual machine dengan klik start seperti pada Gambar 13.



Gambar 13 Start machine

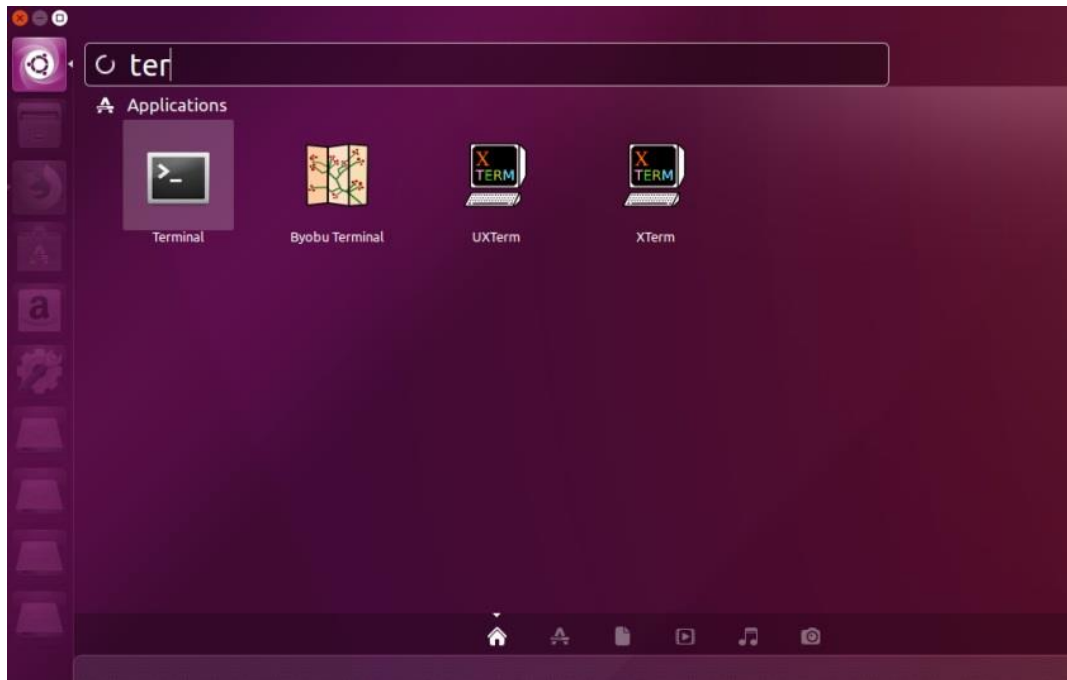
9. Virtual machine akan berjalan seperti pada Gambar 14.



Gambar 14 Virtual machine berjalan

Pengaturan Operating System Ubuntu

Secara default hostname pada image yang digunakan adalah ubuntu dan password adalah ubuntu, agar bisa menunjukkan identitas mahasiswa peserta praktikum hostname ini harus diganti menjadi sesuai dengan NIM masing-masing. Langkah-langkah yang harus dilakukan untuk penggantian hostname adalah sebagai berikut. Buka Terminal baru dengan cara klik sudut kiri desktop Ubuntu dan ketikkan terminal pada search box.



Pada Terminal yang muncul lakukan urutan perintah-perintah berikut agar hostname mengikuti ketentuan ketentuan huruf m diikuti nomor NIM anda, misalnya nomor NIM anda 1600022150 maka menjadi `m1600022150`

1. Edit file `/etc/hostname` dan ubah menjadi sama dengan langkah sebelumnya, dengan cara ketikkan perintah berikut di Terminal

```
$ sudo gedit /etc/hostname
```

Ganti `Ubuntu` dengan `m1600022150`

Save dan Close file tersebut.

2. Berikutnya, edit file `/etc/hosts` dan sesuaikan agar hostname sesuai dengan ketentuan sebelumnya dengan cara ketikkan

```
$ sudo gedit /etc/hosts
```

Ubahlah baris :

```
127.0.1.1 ubuntu
```

Menjadi:

```
127.0.1.1 m1600022150
```

Save dan Close file tersebut.

3. Ketik perintah berikut pada Terminal:

```
$ sudo hostname m1600022150
```

Tutup semua terminal dan buka kembali terminal pastikan hostname sudah berubah sesuai ketentuan.



Aturan Penilaian (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-03	CPMK-01	Selesaikan langkah praktikum	Hasil praktikum langkah	

POST TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	-	-		

1.7. HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-03	CPMK-01			
2.	Praktik	CPL-03	CPMK-01	100%		
3.	Post-Test	CPL-03	CPMK-01			
Total Nilai						



PRAKTIKUM 2: PENGENALAN ROBOT OPERATING SYSTEM

Pertemuan ke : 2

Total Alokasi Waktu : 90 menit

- Materi : 15 menit
- Pre-Test : 15 menit
- Praktikum : 45 menit
- Post-Test : 15 menit

Total Bobot Penilaian : 100%

- Pre-Test : 20 %
- Praktik : 30 %
- Post-Test : 50 %

Pemenuhan CPL dan CPMK:

CPL-08	Mampu memilih, membuat dan menerapkan teknik, sumber daya, penggunaan perangkat teknik modern dan implementasi teknologi informasi untuk memecahkan masalah
CPMK-05	Mampu mengimplementasikan low level control pada robot dan pemrogramannya, High Level control pada robot dan pemrogramannya, Pemanfaatan mesin Visi dan sistem cerdas pada Robot

2.1. DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

1. Menjelaskan apa itu *robot operating system*.
2. Menerapkan penggunaan ROS pada komputer.

2.2. INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-08	CPMK-05	Mampu membangun dan mempresentasikan sistem robot
--------	---------	---

2.3. TEORI PENDUKUNG

Robot Operating System(ROS) adalah sebuah framework untuk menyusun perangkat lunak bagi robot. ROS merupakan kumpulan dari tool, library dan convention yang bertujuan untuk menyederhanakan proses penyusunan perilaku robot yang kompleks dan handal pada berbagai platform robot. ROS saat ini telah berbentuk menjadi suatu proyek besar dengan banyak perintis dan kontributor. ROS bermula dari munculnya kesadaran adanya kebutuhan framework kerjasama yang bersifat terbuka di antara komunitas peneliti robot. Pada mulanya ROS dilatarbelakangi berbagai proyek robot di Universitas Stanford dilanjutkan oleh Willow Garage, suatu kelompok pengembang robot yang secara signifikan melanjutkan konsep yang dibentuk menjadi sebuah implementasi yang teruji. Usaha ini juga didorong oleh banyak peneliti yang berkontribusi pada proyek ROS ini sehingga setelah melalui pengembangan yang bersifat open source ROS menjadi sebuah platform yang digunakan secara luas pada komunitas peneliti robot.

Sebagai Sebuah platform atau framework tentunya ROS tidak terlepas dari pola pikir dan filosofi yang dianut oleh para pengembang nya- Secara Umum ROS mengikuti filosofi yang sama dengan UNIX dalam beberapa aspek. Berikut ini adalah beberapa aspek filosofis Yang digunakan pada ROS.

Peer to peer

Sistem yang dianut oleh Ros adalah adanya sejumlah program yang ter hubung Satu dengan lainnya dan secara kontinyu saling bertukar informasi yang dalam istilah Ros disebut sebagai Messages. Sejumlah Messages ini berpindah secara langsung dari satu program ke program lainnya.

Tool Based

Sebagaimana yang ditunjukkan oleh arsitektur UNIX yang bertahan sepanjang masa, sebuah sistem perangkat lunak yang kompleks dapat disusun dari sejumlah program kecil yang sederhana. Berbeda dengan kebanyakan frame work" perangkat lunak robot yang lain' Ros tidak memiliki sebuah lingkungan pengembangan yang khusus. Untuk menjelajahi atau menelusuri Source code' memvisualisasikan interkoneksi antar Sistem, melakukan plotting data secara grafis, membuat data log maupun menyusun dokumentasi dilakukan dengan program yang terpisah. Hal ini mendorong adanya kreasi baru, Implementasi yang selalu diperbaiki sepanjang waktu karena adanya potensi untuk Saling bertukar implementasi yang paling sesuai untuk tujuan tertentu.

Multilingual

Beberapa tugas perangkat lunak dapat diselesaikan dengan lebih mudah jika dilakukan menggunakan bahasa program yang sederhana semacam Python dan Ruby. Namun demikian ada kalanya Karena tuntutan Kinerja diperlukan penggunaan bahasa pemrograman yang lebih cepat Semisal C++. Ada kalanya situasi lebih sesuai jika digunakan bahasa pemrograman semacam LISP atau Matlab. Dengan memperhatikan fenomena ini, Yaitu bahwa setiap bahasa pemrograman memiliki teguran yang berbeda pada Konteks Yang berbeda dan bahwa Setiap programmer memiliki latar belakang yang berbeda dalam memilih bahasa pemrograman maka Ros memilih pendekatan multilingual. Model perangkat lunak ROS dapat ditulis dalam berbagai bahasa pemrograman yang telah tersedia client library nya dalam ROS. Saat ini telah tersedia client library dalam bahasa C++' Python, LISP Java, Javascript MATLAB Ruby' Haskell dan beberapa bahasa program lainnya. Dalam buku petunjuk praktikum ini akan digunakan bahasa pemrograman C++ dan Python.

Gratis dan Open Source

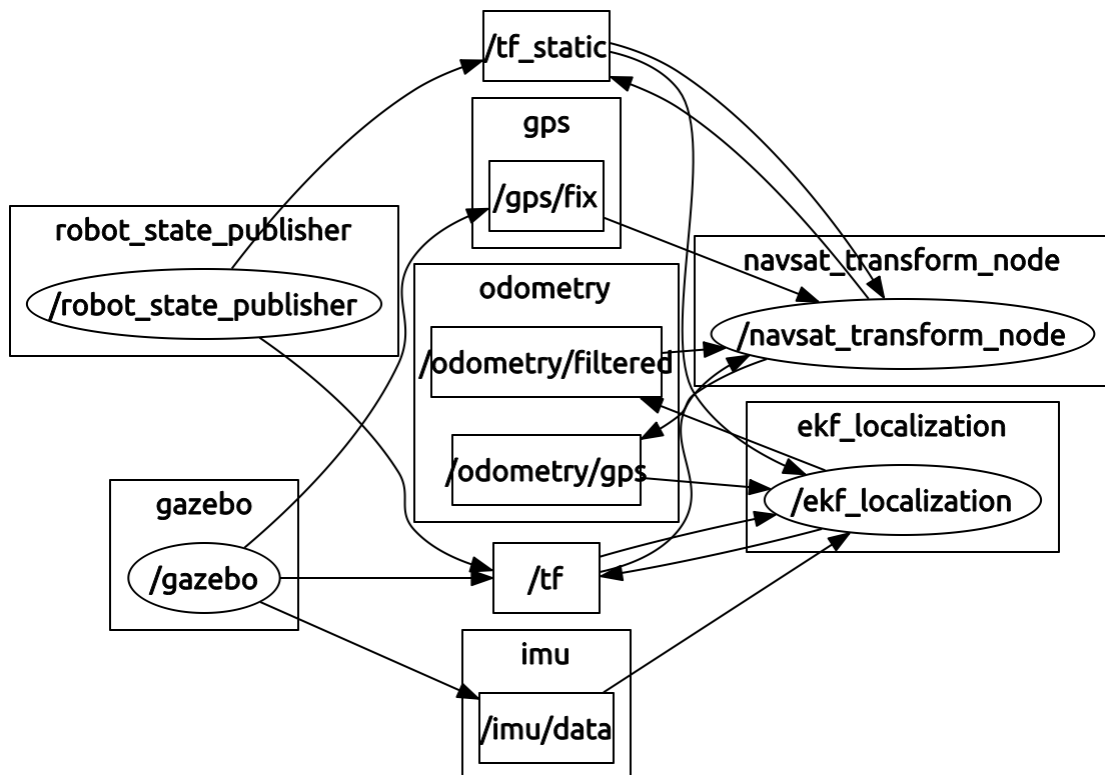
Bagian Utama ROS direlease dengan lisensi BSD yang memungkinkan penggunaan Komersial maupun non komersial. ROS menyalurkan data antar modul ROS dengan menggunakan IPCC Inter Process Communication Yang berarti Setiap sistem yang dibangun dengan ROS dapat memiliki pengaturan lisensi yang detil dan independen pada berbagai Komponennya. Sebagai Contoh sebuah sistem yang bersifat komersial(dari sisi lisensi maupun aplikasi)' seringkali memiliki sejumlah modul yang bersifat Closed Source yang berkomunikasi dengan modul-modul yang bersifat open Source.

ROS Graph dan Node

Salah Satu topik permasalahan yang mendasari pengembangan ROS adalah " fetch an item". Kita dapat membayangkan "fetch an item" ini dalam kasus suatu robot yang dilengkapi sejumlah kamera dan pemindailaser Satu lengan manipulator dan badan beroda. Untuk menyelesaikan permasalahan "fetch an item" robot tersebut harus bernavigasi dalam Sebuah lingkungan yang menyerupai rumah atau Kantor, mencari dan menemukan benda yg dibutuhkan, dan mengantarkannya pada lokasi yg ditetapkan. Tugas Semacam ini sebagaimana tugas- tugas robot lainnya, mendorong para peneliti robot dalam mengembangkan ROS.

1. Penyelesaian tugas dibagi dalam beberapa subsistem yang independen, diantaranya: navigasi, Computer vision, grasping(memegang obyek) dan beberapa sub sistem lainnya.
2. Subsistem yang dikembangkan dapat digunakan sebagai bagian dari tugas lain yang berbeda, misalnya patroli keamanan, pembersihan lingkungan, pengantaran berkas dan lainnya.
3. Dengan layer abstraksi geometri dan hardware yang sesuai sebagian besar bagian dari perangkat lunak akan dapat dijalankan pada berbagai robot.

Tiga hal ini direpresentasikan secara mendasar pada sebuah konsep pada Ros yang disebut graph. Sebuah Sistem ROS terbangun dari berbagai program berbeda yang berjalan bersama-sama dan saling berkomunikasi antara Satu dan lainnya dengan saling bertukar informasi dalam suatu hal yang dalam ROS disebut sebagai messages. Mengingat hal ini, maka Sangatlah cocok jika konsep ini direpresentasikan dalam sebuah graph yang menunjukkan sekumpulan program dan sejumlah Messages. Masing - masing program merupakan node, yang terhubung dengan node yang lain melalui garis anak panah, contoh ROS graph dapat dilihat pada gambar berikut. Detail dari graph ini tidaklah penting, yang utama adalah bahwa graph ini dapat menyajikan atau mendeskripsikan konsep umum pada ROS, Yaitu bahwa Sistem ROS terdiri dari sekumpulan program (nodes) yang saling bertukar messages. Node adalah suatu proses yang melakukan komputasi. Pada dasarnya suatu node adalah suatu subprogram yang bisa dieksekusi/dijalankan yang merupakan bagian dari sebuah aplikasi ROS. Kombinasi sejumlah node membentuk graph dan berkomunikasi antara satu dengan lainnya dengan menggunakan aliran topic. Masing-masing node beroperasi pada level yang detail, suatu sistem kendali robot umumnya terdiri dari beberapa node. Secara grafis Konsep ini diilustrasikan pada gambar berikut.

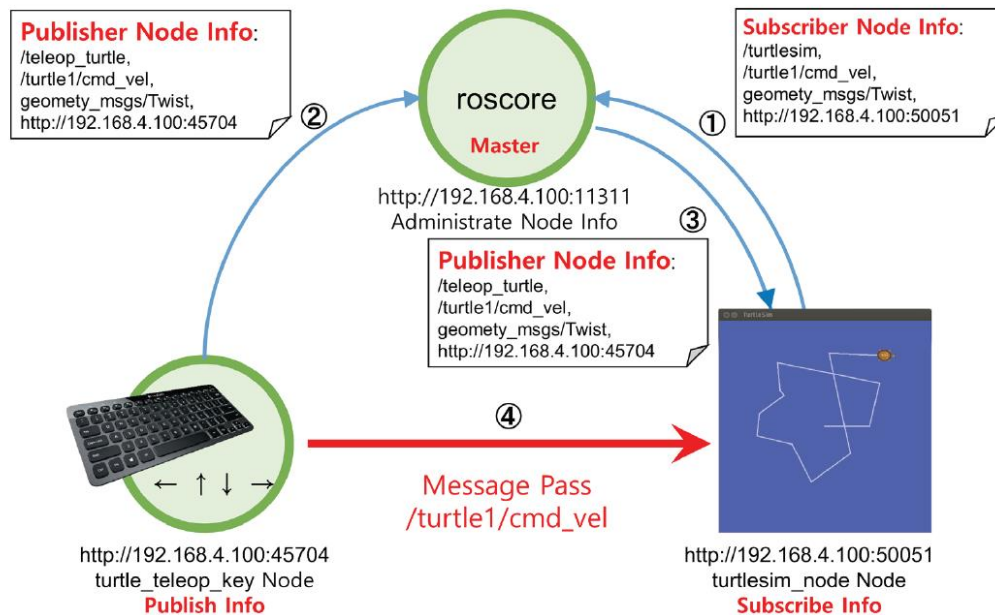


roscore

roscore adalah layanan (Services) yang menyediakan informasi tentang koneksi antar node, sehingga node-node tersebut dapat saling berkirir Messages. Setiap node terhubung ke roscore pada Saat Startup Untuk mencatatkan bagaimana node tersebut mengirimkan messages dan bagaimana node tersebut menerima messages. Ketika sebuah node baru muncul, roscore memberikan kepada node baru tersebut informasi yang diperlukan agar node baru tersebut bisa terhubung dan berkomunikasi dengan node lain Yang sudah ada, khususnya pada topic yang

sesuai. Setiap Sistem ROS memerlukan roscore karena tanpa roscore, masing-masing node tidak akan bisa menemukan node yang lain.

Namun demikian ada hal penting terkait konsep dasar ROS Yang harus Selalu diingat, yaitu bahwa Setiap node berkomunikasi secara langsung(peer-to-peer) dengan node lainnya. roscore hanya dibutuhkan oleh node untuk menemukan node Yang akan menjadi pasangan(Peer bagi node tersebut). Secara grafis Konsep ini diilustrasikan pada gambar berikut.



2.4. HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

1. Laptop Ubuntu dengan ROS Kinetic

2.5. PRE-TEST

Jawablah pertanyaan berikut (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-08	CPMK-05	Apa itu ROS?	50
2.	CPL-08	CPMK-05	Sebutkan dan jelaskan aspek filosofis Yang digunakan pada ROS?	50

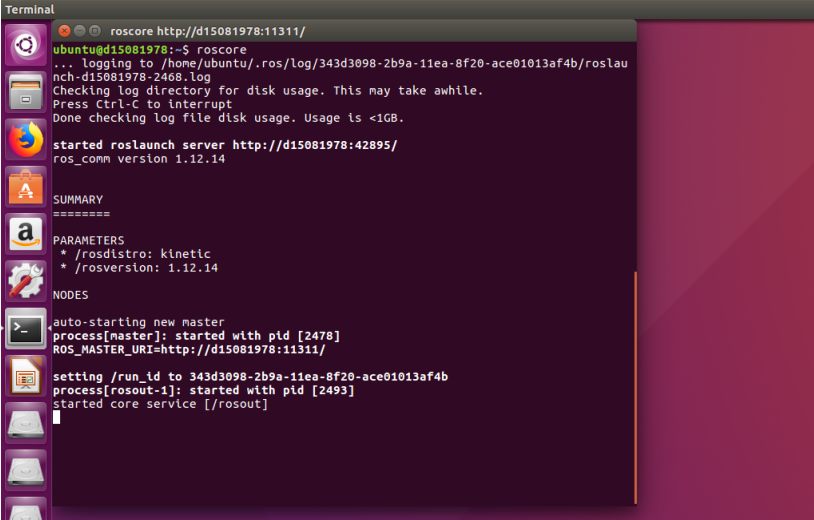
2.6. LANGKAH PRAKTIKUM

1. Memeriksa Instalasi ROS

Buka Terminal baru dan ketikkan perintah berikut :

```
$ roscore
```


Tekan tombol Enter. Jika Sistem sudah terinstall dengan benar maka akan diperoleh tampilan sebagai berikut:



```
Terminal
roscore http://d15081978:11311/
ubuntu@d15081978:~$ roscore
... logging to /home/ubuntu/.ros/log/343d3098-2b9a-11ea-8f20-ace01013af4b/roslau
nch-d15081978-2468.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.
started roslaunch server http://d15081978:42895/
ros_comm version 1.12.14

SUMMARY
=====
PARAMETERS
* /rostdistro: kintetic
* /rosversion: 1.12.14

NODES

auto-starting new master
process[master]: started with pid [2478]
ROS_MASTER_URI=http://d15081978:11311/

setting /run_id to 343d3098-2b9a-11ea-8f20-ace01013af4b
process[rosout-1]: started with pid [2493]
started core service [/rosout]
```

Tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit2-tugas01.

2. Mencoba menjalankan suatu Node ROS

Tetap biarkan terminal pada bagian sebelumnya (no.1) terbuka agar roscore tetap berjalan. Buka Terminal baru dan ketikkan perintah berikut

```
$ rosrunc turtlesim turtlesim_node
```

Tekan tombol Enter. Perhatikan tampilan yang anda peroleh, tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit2-tugas02.

3. Mencoba menggerakkan robot dengan Node draw_square

Tetap biarkan terminal pada bagian sebelumnya (no.1 dan no.2) terbuka agar roscore dan node turtlesim_node tetap berjalan. Buka terminal baru dan ketikkan:

```
$ rosrunc turtlesim draw_square
```

Tekan tombol Enter. Perhatikan tampilan yang anda peroleh, tunggu sampai diperoleh sebuah bentuk persegi(bujur sangkar) yang utuh, tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit2-tugas03.

4. Mencoba menggerrakkan robot dengan Node turtle_teleop_key

Tutup terminal yang anda peroleh pada no.2 dan no.3, biarkan terminal pada no.1 tetap terbuka. Tutup terminal dengan menekan tombol Ctrl+c. diikuti mengetikkan perintah

```
$ exit
```

Tekan Enter.

Selanjutnya buka terminal baru dan ketikkan:

```
$ rosrun turtlesim turtlesim_node
```

Tekan tombol Enter. Buka terminal baru yang lain dan ketikkan

```
$ rosrun turtlesim turtle_teleop_key
```

Tekan tombol Enter. Coba tekan tombol panah pada keyboard untuk menggerakkan robot. Perhatikan tampilan yang anda peroleh, tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit2-tugas04.

5. Menampilkan graph ROS

Biarkan terminal-terminal sebelumnya tetap terbuka. Selanjutnya buka terminal baru dan ketikkan:

```
$ rqt_graph
```

Perhatikan tampilan yang anda peroleh, tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit2-tugas05.

Catatan: Setelah selesai praktikum gunakan tombol ctrl+c untuk menghentikan roscore dan ketikkan perintah exit untuk menutup terminal. Pastikan anda shutdown sepenuhnya sebelum mencabut flashdisk/microSd/SSD anda.

Aturan Penilaian (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-08	CPMK-05	Selesaikan langkah praktikum	Hasil praktikum langkah	100

2.7. POST TEST

Jawablah pertanyaan berikut (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-08	CPMK-05	Buat gerakan berupa lingkaran	100

2.8. HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-03	CPMK-01	20%		
2.	Praktik	CPL-03	CPMK-01	50%		
3.	Post-Test	CPL-03	CPMK-01	30%		
Total Nilai						



PRAKTIKUM 3: KONSEP NODE, MESSAGE, DAN TOPIC PADA ROBOT OPERATING SYSTEM

Pertemuan ke : 3

Total Alokasi Waktu : 90 menit

- Materi : 15 menit
- Pre-Test : 15 menit
- Praktikum : 45 menit
- Post-Test : 15 menit

Total Bobot Penilaian : 100%

- Pre-Test : 20 %
- Praktik : 30 %
- Post-Test : 50 %

Pemenuhan CPL dan CPMK:

CPL-08	Mampu memilih, membuat dan menerapkan teknik, sumber daya, penggunaan perangkat teknik modern dan implementasi teknologi informasi untuk memecahkan masalah
CPMK-05	Mampu mengimplementasikan low level control pada robot dan pemrogramannya, High Level control pada robot dan pemrogramannya, Pemanfaatan mesin Visi dan sistem cerdas pada Robot

3.1. DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

3. Menjelaskan konsep *node*, *message*, dan *topic* pada *robot operating system*.
4. Menerapkan konsep *node*, *message*, dan *topic* pada *robot operating system*.

3.2. INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-08	CPMK-05	Mampu membangun dan mempresentasikan sistem robot
--------	---------	---

3.3. TEORI PENDUKUNG

roscd

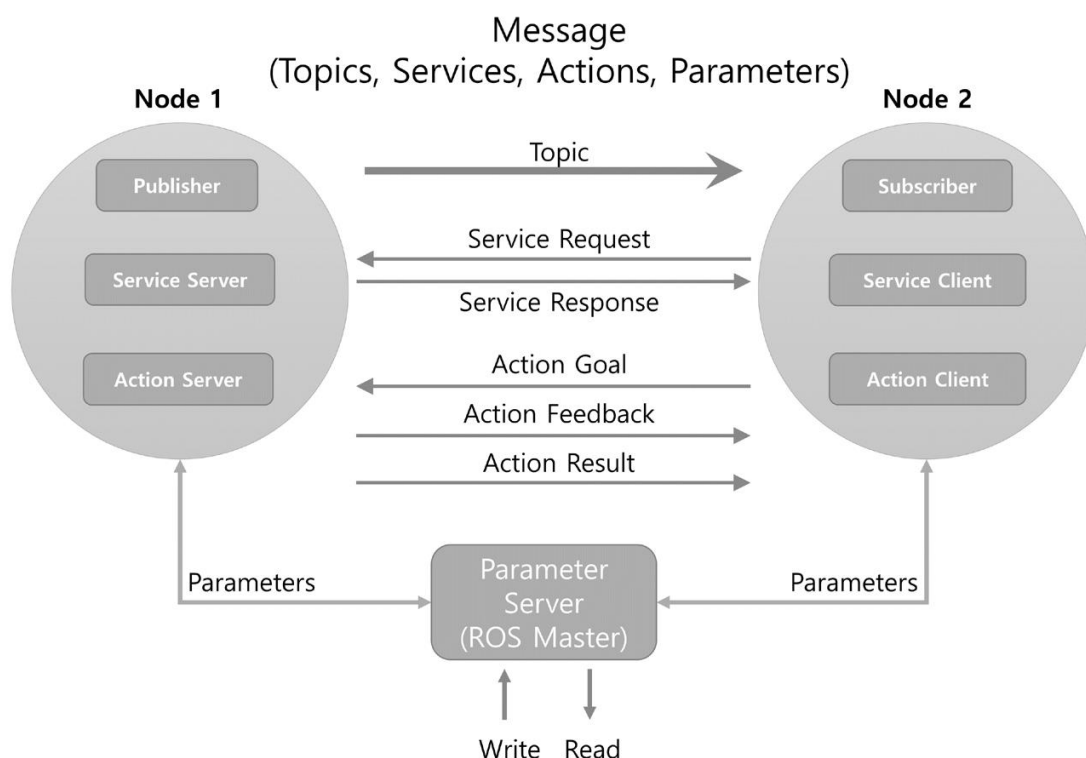
Pada praktikum sebelumnya telah dicoba perintah untuk menjalankan sebuah node dengan menggunakan perintah `roscd`. Terdapat sebuah command-line tool bernama `roscd` yang juga berkaitan dengan pengaturan node. `roscd` menyediakan beberapa perintah yang bisa digunakan ketika bekerja dengan node.

Berikut ini adalah daftar perintah yang disediakan beserta deskripsi singkatnya dalam bahasa Inggris.

```
rosnode info    print information about node
rosnode kill    kill a running node
rosnode list    list active nodes
rosnode machine list nodes running on a particular machine or list machines
rosnode ping    test connectivity to node
rosnode cleanup purge registration information of unreachable nodes
```

Message Communication

Sebagaimana telah disebutkan pada kesempatan sebelumnya, sebuah sistem dalam ROS dibangun oleh sejumlah node yang saling bertukar informasi/messages. Ada tiga metode pertukaran informasi antar node ini. Pertama adalah pertukaran yang disebut topic yang menyediakan jalur transmisi satu arah, metode yang kedua disebut services yang menyediakan jalur komunikasi dua arah dalam bentuk request/response serta metode ketiga yang disebut action yang menyediakan jalur komunikasi dua arah yang berupa goal/result/feedback. Secara grafis ketiga metode komunikasi ini diilustrasikan pada gambar berikut

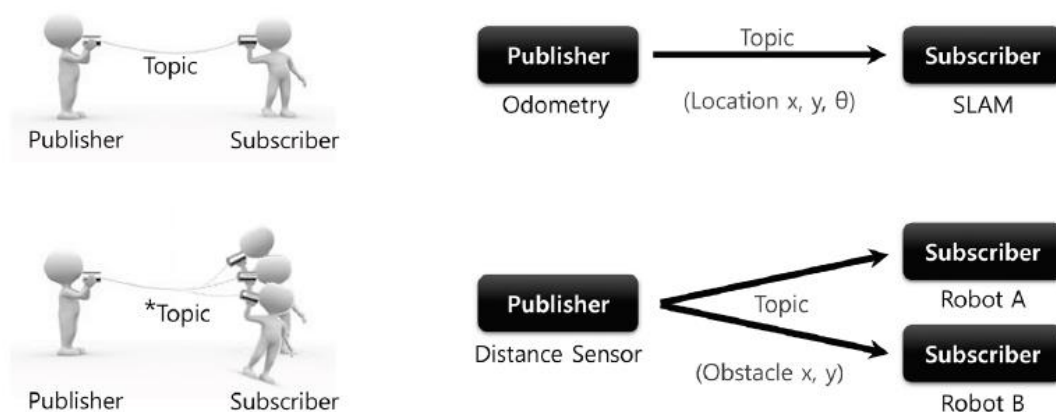


ROS Topic, Publisher dan Subscriber

ROS Topic atau untuk selanjutnya kita sebut sebagai topic adalah jalur komunikasi antar node yang diberi nama tertentu dan digunakan antar node untuk bertukar informasi(messages). Dalam komunikasi melalui topic ini terdapat node yang berperan sebagai publisher dan ada node

yang berperan sebagai subscriber. Sesuai dengan namanya publisher node adalah node yang memancarkan/mengirimkan informasi dalam bentuk message dan node subscriber adalah node yang menerima/mendengarkan informasi tersebut. Salah satu hal penting yang perlu dicatat adalah bahwa hubungan antara publisher dan subscriber bersifat anonymous, atau dengan kata lain tidak saling mengenal. Hal yang diperhatikan oleh sebuah publisher bukanlah dengan siapa node tersebut berhubungan tetapi pada apa yang node tersebut publish melalui sebuah topic, sedangkan pada subscriber, node ini juga tidak memperhatikan dengan siapa dia tersambung tetapi node ini memperhatikan pada topic apa dia tersambung. Ketika berkomunikasi melalui topic sifat komunikasinya adalah satu arah, yaitu dari publisher ke subscriber.

Setiap topic memiliki tipe data tertentu. Master atau roscore tidak mengatur konsistensi tipe data antara sejumlah publisher node namun demikian subscriber node hanya akan menerima message jika memiliki kesamaan tipe data dengan publisher yang tersambung. Gambar berikutnya menjelaskan konsep tersebut.



*Topic not only allows 1:1 Publisher and Subscriber communication, but also supports 1:N, N:1 and N:N depending on the purpose.

Topic Transports

Saat ini ROS mendukung transfer/transport message berbasis TCP/IP dan UDP. Transfer berbasis TCP/IP disebut TCPROS dan menyalurkan data message melalui koneksi TCP/IP persisten. TCPROS adalah cara transfer default yang digunakan pada ROS dan setiap library untuk client hanya wajib dibuat untuk mendukung jenis transfer ini. Transfer berbasis UDP disebut sebagai UDPROS saat ini hanya tersedia dalam versi roscpp. Jenis transfer ini lebih sesuai untuk teleoperation (kendali jarak jauh) karena bersifat low-latency, lossy transport yang selalu diperbaiki sepanjang waktu karena adanya potensi untuk saling bertukar implementasi yang paling sesuai untuk tujuan tertentu.

rostopic command-line tool

Salah satu tool yang sering digunakan ketika berurusan dengan topic adalah `rostopic`. Perintah ini dapat digunakan untuk menampilkan berbagai informasi tentang topic. Perintah ini dapat menampilkan daftar topic yang sedang aktif, nama publisher dan subscriber pada suatu topic tertentu, publishing rate suatu topic, bandwidth suatu topic dan message yang terpublish pada suatu topic. Berikut ini jenis perintah terkait rostopic yang tersedia saat ini


```

rostopic bw      display bandwidth used by topic
rostopic delay  display delay for topic which has header
rostopic echo   print messages to screen
rostopic find   find topics by type
rostopic hz     display publishing rate of topic
rostopic info   print information about active topic
rostopic list   print information about active topics
rostopic pub    publish data to topic
rostopic type   print topic type

```

3.4. HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

1. Laptop Ubuntu dengan ROS Kinetic

3.5. PRE-TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-08	CPMK-05	Apa yang dimaksud dengan <i>node</i> , <i>message</i> , dan <i>topic</i> pada <i>robot operating system</i> ?	100

3.6. LANGKAH PRAKTIKUM

1. Menggunakan perintah `rostopic list`

Buka Terminal baru dan ketikkan perintah berikut :

```
$ roscore
```

Tekan tombol Enter.

Buka Terminal baru dan ketikkan perintah berikut

```
$ rostopic list
```

Tekan tombol Enter.

Tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit3-tugas01a.

Buka Terminal baru dan ketikkan perintah berikut

```
$ roslaunch turtlesim turtlesim_node
```

Tekan tombol Enter. Kembali ke terminal yang sebelumnya, ulangi perintah

```
$ rostopic list
```

Tekan tombol Enter.

Tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit3-tugas01b.

2. Menggunakan perintah `roscd`

Tetap biarkan terminal pada bagian sebelumnya (no.1) terbuka semua. Buka Terminal baru dan ketikkan perintah berikut

```
$ roscd /turtlesim
```

Tekan tombol Enter. Atur ukuran terminal agar semua informasi terlihat, perhatikan tampilan yang anda peroleh, tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit3-tugas02.

3. Menggunakan perintah `roscd`

Tetap biarkan terminal pada bagian sebelumnya (no.1 dan no.2) terbuka agar roscore dan node `turtlesim_node` tetap berjalan. Buka terminal baru dan ketikkan:

```
$ roscd /turtlesim
```

Tekan tombol Enter. Perhatikan tampilan yang anda peroleh, tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit3-tugas03.

4. Menggunakan perintah `roscd`

Biarkan terminal yang masih terbuka, buka terminal baru dan ketikkan:

```
$ roscd /turtlesim
```

Tekan tombol Enter. Buka terminal baru yang lain dan ketikkan

```
$ roscd /turtlesim
```

Tekan tombol Enter. Coba tekan tombol panah pada keyboard untuk menggerakkan robot. Perhatikan tampilan yang anda peroleh, tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit3-tugas04.

5. Menggunakan perintah `roscd`

Biarkan terminal-terminal sebelumnya tetap terbuka. Selanjutnya buka terminal baru dan ketikkan:

```
$ rostopic info /turtle1/cmd_vel
```

Tekan tombol Enter.

Perhatikan tampilan yang anda peroleh, atekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit3-tugas05.

6. Menggunakan perintah rostopic echo

Biarkan terminal-terminal sebelumnya tetap terbuka. Selanjutnya buka terminal baru dan ketikkan:

```
$ rostopic echo /turtle1/cmd_vel
```

Tekan tombol Enter.

Atur agar terminal ini mudah terlihat, selalu perhatikan terminal ini. Buka terminal baru dan ketikkan

```
$ rosrun turtlesim draw_square
```

Tekan tombol Enter, tunggu sampai terbentuk sebuah persegi utuh, tekan ctrl+c pada terminal `draw_square` sambil tetap memperhatikan terminal sebelumnya. Atur ukuran terminal agar informasi pada terminal sebelumnya terlihat dengan utuh. Tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit3-tugas06.

7. Menggunakan perintah rostopic pub

8. Tutup semua terminal kecuali terminal roscore. Selanjutnya buka terminal baru dan ketikkan:

```
$ rosrun turtlesim turtlesim_node
```

Buka satu terminal baru dan ketikkan

```
$ rostopic pub /turtle1/cmd_vel geometry_msgs/Twist "linear:  
  x: 1.0  
  y: 0.0  
  z: 0.0
```

```
angular:
  x: 0.0
  y: 0.0
  z: 0.0"
```

catatan: untuk memudahkan mengetikkan perintah tersebut dapat digunakan bantuan tombol tab. Sehingga untuk mengetikkan perintah di atas urutan pengetikannya

```
$ rostopic pub /turtle1/cmd_vel geometry_msgs/Twist (Tab) (Tab)
```

Akan diperoleh

```
$ rostopic pub /turtle1/cmd_vel geometry_msgs/Twist "linear:
  x: 0.0
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 0.0"
```

Gunakan tombol panah kanan kiri atas bawah dan ganti agar `` Linear:

```
x: 1.0
```

Tekan tombol Enter.

Perhatikan tampilan yang anda peroleh, tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit3-tugas07a. Tekan tombol ctrl-c.

9. Selanjutnya pada terminal yang sama ketikkan perintah berikut

```
$ rostopic pub /turtle1/cmd_vel geometry_msgs/Twist "linear:
  x: 1.0
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
```

```
z: 1.0"
```

Gunakan cara bantuan tombol Tab atau tombol up(panah ke atas)lakukan perubahan yang diperlukan.

Tekan tombol Enter.

Perhatikan tampilan yang anda peroleh, tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit3-tugas07b. Tekan tombol ctrl-c.

10. Masih pada terminal yang sama ketikkan perintah berikut

```
rostopic pub -1 /turtle1/cmd_vel geometry_msgs/Twist "linear:  
  x: 1.0  
  y: 0.0  
  z: 0.0  
angular:  
  x: 0.0  
  y: 0.0  
  z: 1.0"
```

Gunakan cara bantuan tombol Tab atau tombol up(panah ke atas), lakukan perubahan yang diperlukan.

Tekan tombol Enter.

Perhatikan tampilan yang anda peroleh, tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit3-tugas07c.

11. Tetap gunakan terminal yang sama, ketikkan perintah berikut

```
rostopic pub -r 1 /turtle1/cmd_vel geometry_msgs/Twist "linear:  
  x: 1.0  
  y: 0.0  
  z: 0.0  
angular:  
  x: 0.0  
  y: 0.0  
  z: 1.0"
```

Gunakan cara bantuan tombol Tab atau tombol up(panah ke atas), lakukan perubahan yang diperlukan.

Tekan tombol Enter.

Perhatikan tampilan yang anda peroleh, tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit3-tugas07d.

12. Mengatur penamaan node

Tutup semua terminal kecuali yang menjalankan roscore, buka terminal baru dan ketikkan

```
$ rosrun turtlesim turtlesim_node __name:=penyusatu
```

Tekan Enter

Buka terminal baru dan ketikkan

```
$ rosrun turtlesim draw_square
```

Tekan Enter.

Perhatikan tampilan yang anda peroleh, tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit3-tugas08a.

Biarkan terminal-terminal sebelumnya tetap terbuka. Selanjutnya buka terminal baru dan ketikkan:

```
$ rqt_graph
```

Tekan Enter.

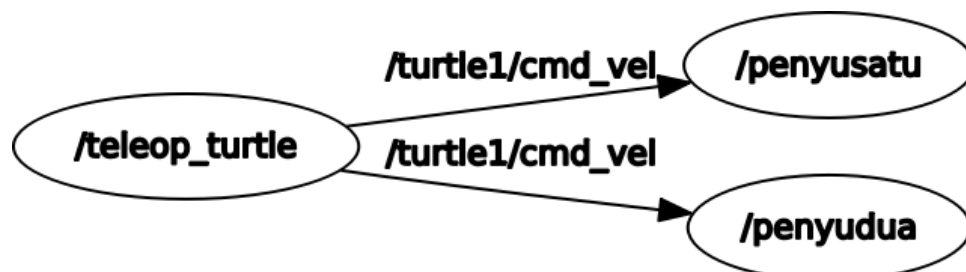
Perhatikan tampilan yang anda peroleh, tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit3-tugas08b.

Aturan Penilaian (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-08	CPMK-05	Selesaikan langkah praktikum	Hasil praktikum langkah	100

3.7. POST TEST

Jalankan atau buka node yang diperlukan agar diperoleh hasil rqt_graph sebagai berikut



Atur agar semua terminal dan rqt_graph dapat terlihat, gunakan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit3-tugasakhir.

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-08	CPMK-05	Lihat soal diatas	100

3.8. HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-03	CPMK-01	20%		
2.	Praktik	CPL-03	CPMK-01	30%		
3.	Post-Test	CPL-03	CPMK-01	50%		
Total Nilai						



PRAKTIKUM 4: KONSEP SERVICE PADA ROBOT OPERATING SYSTEM

- Pertemuan ke** : 4
- Total Alokasi Waktu** : 90 menit
- Materi : 15 menit
 - Pre-Test : 15 menit
 - Praktikum : 45 menit
 - Post-Test : 15 menit
- Total Bobot Penilaian** : 100%
- Pre-Test : 20 %
 - Praktik : 30 %
 - Post-Test : 50 %

Pemenuhan CPL dan CPMK:

CPL-08	Mampu memilih, membuat dan menerapkan teknik, sumber daya, penggunaan perangkat teknik modern dan implementasi teknologi informasi untuk memecahkan masalah
CPMK-05	Mampu mengimplementasikan low level control pada robot dan pemrogramannya, High Level control pada robot dan pemrogramannya, Pemanfaatan mesin Visi dan sistem cerdas pada Robot

4.1. DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

1. Menjelaskan konsep service pada *robot operating system*.
2. Menerapkan konsep service pada *robot operating system*.

4.2. INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-08	CPMK-05	Mampu membangun dan mempresentasikan sistem robot
--------	---------	---

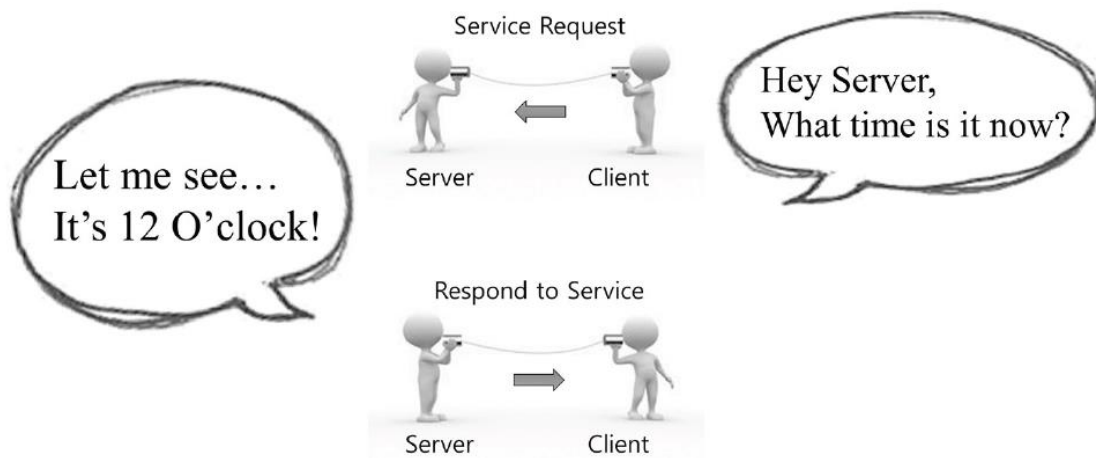
4.3. TEORI PENDUKUNG

Service

Pada praktikum sebelumnya telah disinggung tentang tiga metode komunikasi antar node pada ROS. Service merupakan salah satu dari ketiga metode tersebut. Komunikasi yang terjadi pada suatu service adalah Suatu komunikasi sinkron bersifat dua arah antara suatu service client dan Service server. Service client meminta request sebuah Service kepada Ser vice server yang akan

merespon permintaan tersebut. Konsep "publish" dan "subscribe" yang ada pada komunikasi topic lebih Sesuai untuk pengiriman data yang bersifat periodik- untuk kondisi tertentu yang dibutuhkan adalah komunikasi sinkron yang berupa suatu permintaan dan respon. Metode komunikasi ini dalam Ros disebut service.

Suatu komunikasi dengan metode service terdiri dari suatu service server yang hanya akan merespon ketika ada request /permintaan dari suatu service client. Tidak Seperti topic komunikasi dalam bentuk service adalah one time communication. Oleh karena ketika proses request-response selesai koneksi antara node yang berkomunikasi akan terputus. Pada umumnya service digunakan untuk memerintahkan robot untuk melakukan suatu tindakan tertentu atau memerintahkan suatu node untuk melakukan hal tertentu dengan persyaratan tertentu. Konsep service dapat dilihat pada gambar berikut



rosservice command-line tool

Salah satu tool yang berkaitan dengan service adalah `rosservice`. Berikut ini jenis perintah terkait `rosservice` yang tersedia saat ini

```
rosservice call call the service with the provided args
rosservice find find services by service type
rosservice info print information about service
rosservice list list active services
rosservice type print service type
rosservice uri print service ROSRPC uri
```

4.4. HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

1. Laptop Ubuntu dengan ROS Kinetic

4.5. PRE-TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-08	CPMK-05	Jelaskan konsep service pada <i>robot operating system</i> !	100

4.6. LANGKAH PRAKTIKUM

1. Menggunakan perintah `rosservice list`

Buka Terminal baru dan ketikkan perintah berikut :

```
$ roscore
```

Tekan tombol Enter.

Buka Terminal baru dan ketikkan perintah berikut

```
$ rosservice list
```

Tekan tombol Enter.

Tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit4-tugas01a.

Buka Terminal baru dan ketikkan perintah berikut

```
$ rosrunc turtlesim turtlesim_node
```

Tekan tombol Enter. Kembali ke terminal yang sebelumnya, ulangi perintah

```
$ rosservice list
```

Tekan tombol Enter.

Tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit4-tugas01b.

2. Menggunakan perintah `rosservice info`

Tetap biarkan terminal pada bagian sebelumnya (no.1) terbuka semua. Buka Terminal baru dan ketikkan perintah berikut

```
$ rosservice info /clear
```

Tekan tombol Enter. Atur ukuran terminal agar semua informasi hasil perintah ini terlihat, tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit4-tugas02a.

Pada terminal yang sama ketikkan

```
$ rosservice info /reset
```

Tekan tombol Enter. Atur ukuran terminal agar semua informasi hasil perintah ini terlihat, tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit4-tugas02b.

Pada terminal yang sama ketikkan

```
$ rosservice info /turtle1/teleport_relative
```

Tekan tombol Enter. Atur ukuran terminal agar semua informasi hasil perintah ini terlihat, tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit4-tugas02c.

Pada terminal yang sama ketikkan

```
$ rosservice info /turtle1/teleport_absolute
```

Tekan tombol Enter. Atur ukuran terminal agar semua informasi hasil perintah ini terlihat, tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit4-tugas02d.

Pada terminal yang sama ketikkan

```
$ rosservice info /turtle1/set_pen
```

Tekan tombol Enter. Atur ukuran terminal agar semua informasi hasil perintah ini terlihat, tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit4-tugas02e.

3. Menggunakan perintah rosservice call

Tetap biarkan terminal pada bagian sebelumnya (no.1 dan no.2) terbuka agar roscore dan node turtlesim_node tetap berjalan. Buka terminal baru dan ketikkan:

```
$ rosrunc turtlesim turtle_teleop_key
```

4. Tekan tombol Enter. Gerakkan robot sesuai keinginan anda dengan menggunakan keyboard/tombol panah. Perhatikan tampilan yang anda peroleh, tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit4-tugas03a.

Buka Terminal baru dan ketikkan perintah:

```
$ rosservice call /clear
```

5. Tekan tombol Enter. Perhatikan tampilan yang anda peroleh, tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit4-tugas03b.

Pada Terminal rosservice ketikkan perintah:

```
$ rosservice call /reset
```

6. Tekan tombol Enter. Perhatikan tampilan yang anda peroleh, tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit4-tugas03c.

Pada Terminal rosservice ketikkan perintah:

```
$ rosservice call /turtle1/teleport_absolute 1.0 1.0 0.0
```

7. Tekan tombol Enter. Perhatikan tampilan yang anda peroleh, tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit4-tugas03d. Catatan: ada bisa menggunakan tombol tab untuk memudahkan mengetikkan perintah di atas.

Pada Terminal rosservice ketikkan perintah:

```
$ rosservice call /turtle1/teleport_relative 1.0 0.0
```

8. **Tekan tombol Enter. Perhatikan tampilan yang anda peroleh, tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit4-tugas03e. Catatan: ada bisa menggunakan tombol tab untuk memudahkan mengetikkan perintah di atas.**

Pastikan terminal `turtle_teleop_key` masih aktif, pada terminal `rosservice` ketikkan

```
$ rosservice call /turtle1/set_pen 255 0 0 5 0
```

9. **Tekan tombol Enter. Gerakkan robot sesuai keinginan anda dengan menggunakan keyboard/tombol panah. Perhatikan tampilan yang anda peroleh, tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit4-tugas03f.**

Pastikan terminal `turtle_teleop_key` masih aktif, pada terminal `rosservice` ketikkan

```
$ rosservice call /turtle1/set_pen 255 0 0 5 1
```

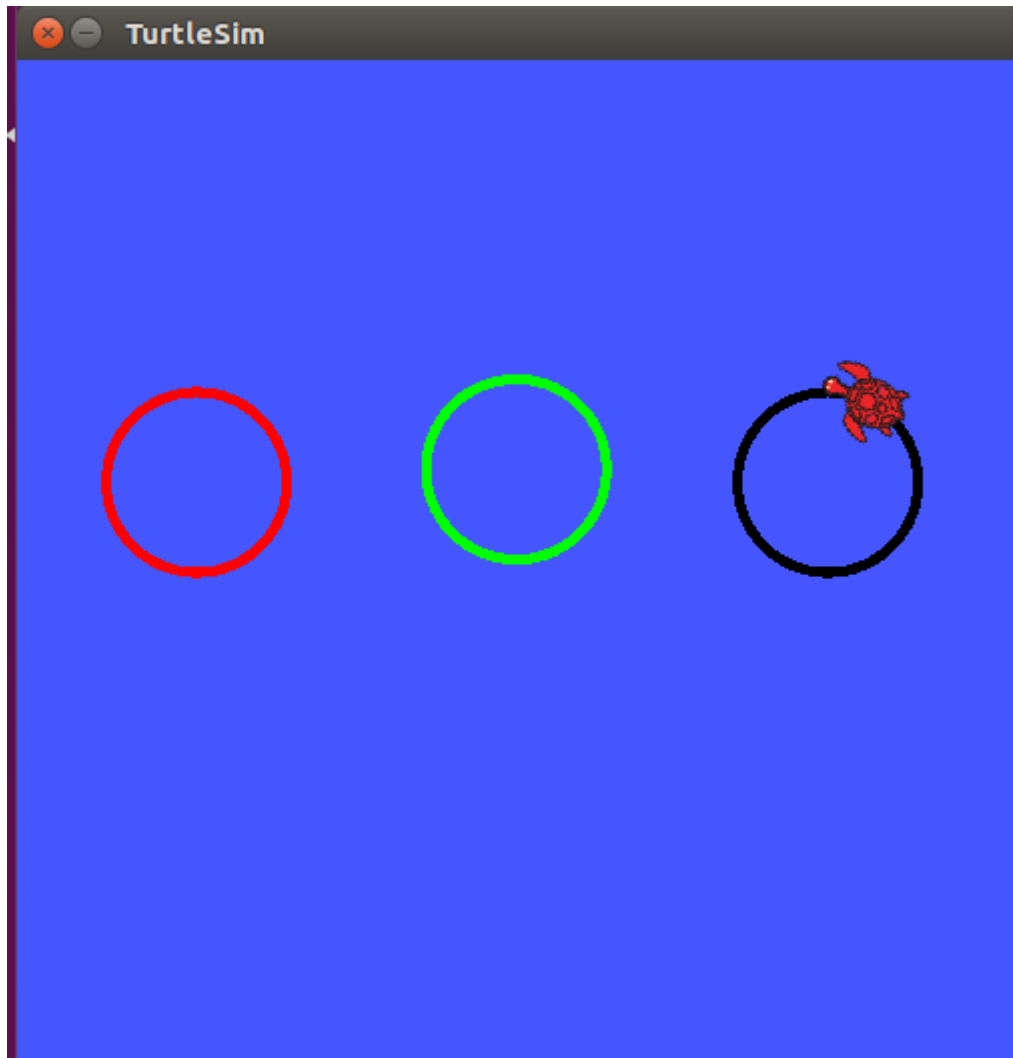
10. **Tekan tombol Enter. Gerakkan robot sesuai keinginan anda dengan menggunakan keyboard/tombol panah. Perhatikan tampilan yang anda peroleh, tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit4-tugas03g.**

Aturan Penilaian (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-08	CPMK-05	Selesaikan langkah praktikum	Hasil praktikum langkah	100

4.7. POST TEST

Dengan menggunakan perintah-perintah yang telah anda pelajari pada unit ini dan unit-unit sebelumnya buatlah agar diperoleh jejak/lintasan robot turtle berbentuk lingkaran sebanyak tiga buah dengan warna merah, hijau, dan hitam. Gambar berikut menunjukkan contoh hasil yang diminta. Antara satu lingkaran dengan lingkaran yang lain saling terpisah, tidak diperbolehkan saling tindih/saling berpotongan. Ukuran lingkaran bebas. Gunakan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit4-tugasakhir.



No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-08	CPMK-05	Lihat pertanyaan diatas	100

4.8. HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-03	CPMK-01	20%		
2.	Praktik	CPL-03	CPMK-01	30%		
3.	Post-Test	CPL-03	CPMK-01	50%		
Total Nilai						



LEMBAR JAWABAN PRE-TEST DAN POST-TEST PRAKTIKUM

Nama : NIM :	Asisten: Paraf Asisten:	Tanggal: Nilai:
-------------------------------	--	----------------------------------

--



PRAKTIKUM 5: KONSEP PACKAGE PADA ROBOT OPERATING SYSTEM

Pertemuan ke : 5

Total Alokasi Waktu : 90 menit

- Materi : 15 menit
- Pre-Test : 15 menit
- Praktikum : 45 menit
- Post-Test : 15 menit

Total Bobot Penilaian : 100%

- Pre-Test : 20 %
- Praktik : 30 %
- Post-Test : 50 %

Pemenuhan CPL dan CPMK:

CPL-08	Mampu memilih, membuat dan menerapkan teknik, sumber daya, penggunaan perangkat teknik modern dan implementasi teknologi informasi untuk memecahkan masalah
CPMK-05	Mampu mengimplementasikan low level control pada robot dan pemrogramannya, High Level control pada robot dan pemrogramannya, Pemanfaatan mesin Visi dan sistem cerdas pada Robot

5.1. DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

3. Menjelaskan konsep package pada *robot operating system*.
4. Menerapkan konsep package pada *robot operating system*.

5.2. INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-08	CPMK-05	Mampu membangun dan mempresentasikan sistem robot
--------	---------	---

5.3. TEORI PENDUKUNG

Package

Suatu package adalah Satuan dasar(basic unit) yang terdapat pada ROS. Aplikasi pada ROS pada Umumnya dikembangkan berbasis package yang didalamnya(package) berisi baik suatu file konfigurasi untuk memanggil package lain ataupun sejumlah node. Suatu package juga menyimpan semua file yang diperlukan untuk menjalankan package tersebut, termasuk library ROS yang

diperlukan untuk menjalankan berbagai proses datasets, dan file konfigurasi. Saat ini terdapat sekitar 1600 package untuk ROS Kinetic.

Meta package

Suatu metapackage adalah sekumpulan package yang memiliki kegunaan yang serupa. Sebagai contoh Metapackage untuk navigasi robot terdiri dari 10 package yang didalamnya termasuk package AMCL, DUA EKF dan map_server.

Roslaunch

Jika `roslaunch` adalah suatu perintah yang digunakan untuk menjalankan sebuah node maka `roslaunch` adalah perintah yang digunakan untuk memanggil atau menjalankan beberapa node secara bersamaan sekaligus. `roslaunch` adalah perintah khusus pada ROS yg berkaitan dengan eksekusi node dengan beberapa tambahan fungsi misalnya mengubah parameter package atau merubah nama node atau mengkonfigurasi node serta kemampuan/fasilitas untuk mengubah variable yang digunakan ketika menjalankan node atau sejumlah node.

5.4. HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

1. Laptop Ubuntu dengan ROS Kinetic

5.5. PRE-TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-08	CPMK-05	Jelaskan konsep package pada <i>robot operating system</i> .	100

5.6. LANGKAH PRAKTIKUM

1. Pengaturan Workspace Folder

Buka Terminal baru dan ketikkan perintah berikut :

```
$ mkdir -p ~/catkin_ws/src
```

Tekan tombol Enter. Masih pada terminal yang sama lanjutkan ketik perintah berikut :

```
$ cd ~/catkin_ws/src
```

Tekan tombol Enter. Masih pada terminal yang sama lanjutkan ketik perintah berikut:

```
$ catkin_init_workspace
```

Tekan tombol Enter.

Setelah rangkaian perintah tersebut dijalankan, sebuah catkin workspace folder telah anda buat dan selanjutnya anda dapat bekerja pada folder tersebut. Untuk memastikan semua telah sesuai. Masih pada terminal yang sama ketikkan perintah berikut.

```
$ cd ~/catkin_ws/
```

Tekan tombol Enter. Masih pada Terminal tersebut ketikkan perintah berikut

```
$ catkin_make
```

Tekan tombol Enter.

Jika semua telah anda atur dengan benar maka akan terjadi kompilasi tanpa error.

Lanjutkan pada terminal yang sama perintah berikut:

```
$ ls
```

Tekan tombol Enter. Pastikan bahwa anda memperoleh keluaran yang menunjukkan directory `catkin_ws` berisi folder `build`, `devel` dan `src`.

2. Pengaturan directory kerja ROS

Agar direktori kerja yang baru saja dibuat dapat dikenali oleh ROS maka perlu dilakukan pengaturan. Ikuti rangkaian perintah berikut untuk melakukan pengaturan tersebut. Buka Terminal baru dan ketikkan perintah berikut

```
$ gedit ~/.bashrc
```

Pada file yang terbuka cari baris yang bertuliskan

```
source /opt/ros/kinetic/devel/setup.bash
```

Tambahkan dibawahnya

```
source ~/catkin_ws/devel/setup.bash
```

Simpan perubahan dan tutup file tersebut. Selanjutnya pada terminal yang terbuka ketikkan perintah

```
$ source ~/.bashrc
```

3. Membuat package ROS

Buka terminal baru dan ketikkan:

```
$ cd ~/catkin_ws/src
```

a. Tekan tombol Enter.

Masih pada Terminal yang sama ketikkan perintah:

```
$ catkin_create_pkg ros_tutorials_topic message_generation std_msgs roscpp
```

b. Tekan tombol Enter.

Pada Terminal yang sama ketikkan perintah:

```
$ cd ros_tutorials_topic
```

c. Tekan tombol Enter.

Pada Terminal yang sama ketikkan perintah:

```
$ ls
```

d. Tekan tombol Enter. Pastikan anda memperoleh keluaran berupa daftar yang terdiri dari include, src, CmakeLists.txt dan package.xml.

Pada Terminal yang sama ketikkan perintah:

```
$ gedit package.xml
```

e. Tekan tombol Enter. Pada file yang terbuka lakukan perubahan agar file tersebut berisi tulisan berikut

```
<?xml version="1.0"?>
<package>
<name>ros_tutorials_topic</name>
<version>0.1.0</version>
<description>ROS tutorial package to learn the topic</descriptio
n>
<license>Apache License 2.0</license>
<author email="NIM@uad.ac.id">Nama_anda</author>
<maintainer email="NIM@uad.ac.id">Nama_anda</maintainer>
<url type="website">http://uad.ac.id</url>
<buildtool_depend>catkin</buildtool_depend>
<build_depend>roscpp</build_depend>
<build_depend>std_msgs</build_depend>
<build_depend>message_generation</build_depend>
<run_depend>roscpp</run_depend>
<run_depend>std_msgs</run_depend>
<run_depend>message_runtime</run_depend>
<export></export>
</package>
```

Simpan perubahan dan tutup file tersebut.

f. Buka terminal baru dan ketikkan perintah berikut

```
$ gedit CMakeList.txt
```

- g. Tekan tombol Enter. Pada file yang terbuka lakukan perubahan agar file tersebut berisi tulisan berikut.

```
cmake_minimum_required(VERSION 2.8.3)
project(ros_tutorials_topic)
find_package(catkin REQUIRED COMPONENTS message_generation std_msgs ros
  cpp)
add_message_files(FILES MsgTutorial.msg)
generate_messages(DEPENDENCIES std_msgs)
catkin_package(
  LIBRARIES ros_tutorials_topic
  CATKIN_DEPENDS std_msgs roscpp
)
include_directories(${catkin_INCLUDE_DIRS})
add_executable(topic_publisher src/topic_publisher.cpp)
add_dependencies(topic_publisher ${${PROJECT_NAME}_EXPORTED_TARGETS}
  ${catkin_EXPORTED_TARGETS})
target_link_libraries(topic_publisher ${catkin_LIBRARIES})
add_executable(topic_subscriber src/topic_subscriber.cpp)
add_dependencies(topic_subscriber ${${PROJECT_NAME}_EXPORTED_TARGETS}
  ${catkin_EXPORTED_TARGETS})
target_link_libraries(topic_subscriber ${catkin_LIBRARIES})
```

Simpan perubahan dan tutup file tersebut.

- h. Pada terminal yang sama ketikkan perintah berikut.

```
$ cd ~/catkin_ws/src/ros_tutorials_topic
```

- i. Tekan Enter. Pada terminal yang sama ketikkan perintah

```
$ mkdir msg
```

- j. Tekan Enter. Pada terminal yang sama ketikkan perintah

```
$ cd msg
```

- k. Tekan tombol Enter. Pada Terminal yang sama ketikkan perintah:

```
$ gedit MsgTutorial.msg
```

- l. Tekan tombol Enter. Pada file yang terbuka lakukan perubahan agar file tersebut berisi tulisan berikut.

```
time stamp
int32 data
```

Simpan perubahan dan tutup file tersebut.

m. Pada Terminal yang sama ketikkan perintah:

```
$ cd ~/catkin_ws/src/ros_tutorials_topic/src
```

n. Tekan tombol Enter. Pada Terminal yang sama ketikkan perintah:

```
$ gedit topic_publisher.cpp
```

o. Tekan tombol Enter. Pada file yang terbuka lakukan perubahan agar file tersebut berisi tulisan sesuai/sama dengan `topic_publisher.cpp` terlampir.

Simpan perubahan dan tutup file tersebut.

p. Pada Terminal yang sama ketikkan perintah:

```
$ cd ~/catkin_ws/src/ros_tutorials_topic/src
```

q. Tekan tombol Enter. Pada Terminal yang sama ketikkan perintah:

```
$ gedit topic_subscriber.cpp
```

r. Tekan tombol Enter. Pada file yang terbuka lakukan perubahan agar file tersebut berisi tulisan sesuai/sama dengan `topic_subscriber.cpp` terlampir.:

Simpan perubahan dan tutup file tersebut.

s. Pada Terminal yang sama ketikkan perintah:

```
$ cd ~/catkin_ws/
```

t. Tekan tombol Enter. Pada Terminal yang sama ketikkan perintah:

```
$ catkin_make
```

Tunggu hingga kompilasi selesai dengan tanpa timbul error.

u. Buka Terminal baru dan ketikkan perintah berikut :

```
$ roscore
```

v. Tekan tombol Enter. Buka Terminal baru dan ketikkan perintah berikut:

```
$ rosruncatkin_ws/src/ros_tutorials_topic/topic_publisher
```


w. SdsTekan tombol Enter. Tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit5-tugas01w. Buka Terminal baru dan ketikkan perintah berikut:

```
$ rosruncatkin_ws/src/ros_tutorials_topic/topic_subscriber
```

x. Tekan tombol Enter. Tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit5-tugas01x. Buka Terminal baru dan ketikkan perintah berikut:

```
$ rqt_graph
```

y. Tekan tombol Enter. Tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit5-tugas01y.

Catatan: Jika saat menjalankan perintah v), w) , x) dan y) terdapat kendala coba terlebih dahulu ketikkan pada setiap terminal yang baru dibuka sebelum menjalankan perintah pada v), w) , x) dan y).

```
$ source ~/catkin_ws/devel/setup.bash
```

4. Membuat dan menggunakan roslaunch

Tutup semua terminal sebelumnya. Buka terminal baru dan ketikkan perintah berikut

```
$ cd ~/catkin_ws/src/ros_tutorials_topic
```

Tekan tombol Enter. Pada Terminal yang sama ketikkan perintah:

```
$ mkdir launch
```

Tekan tombol Enter. Pada Terminal yang sama ketikkan perintah:

```
$ cd launch
```

Tekan tombol Enter. Pada Terminal yang sama ketikkan perintah:

```
$ gedit launch
```

Tekan tombol Enter. Pada file yang terbuka lakukan perubahan agar file tersebut berisi tulisan berikut.

```
<launch>
<group ns="ns1">
<node pkg="ros_tutorials_topic" type="topic_publisher" name="
topic_publisher"/>
<node pkg="ros_tutorials_topic" type="topic_subscriber" name=
```

```

"topic_subscriber"/>
</group>
<group ns="ns2">
<node pkg="ros_tutorials_topic" type="topic_publisher" name="
topic_publisher"/>
<node pkg="ros_tutorials_topic" type="topic_subscriber" name=
"topic_subscriber"/>
</group>
</launch>

```

Simpan perubahan dan tutup file tersebut.

Buka terminal baru dan ketikkan perintah berikut;

```
$ roscore
```

Tekan tombol Enter(pastikan hanya ada satu roscore yang dijalankan/dipanggil). Buka terminal baru dan ketikkan

```
$ roslaunch ros_tutorials_topic union.launch --screen
```

Tekan tombol Enter(pastikan hanya ada satu roscore yang dijalankan/dipanggil). Buka terminal baru dan ketikkan

```
$ rqt_graph
```

Tekan tombol Enter. Tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit5-tugas02.

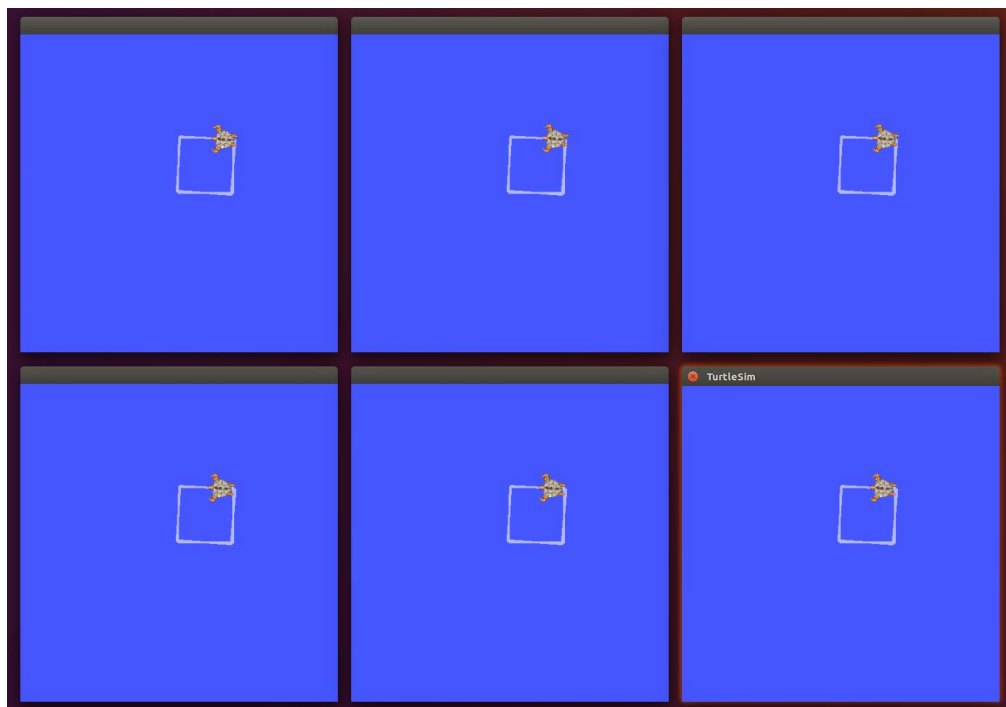
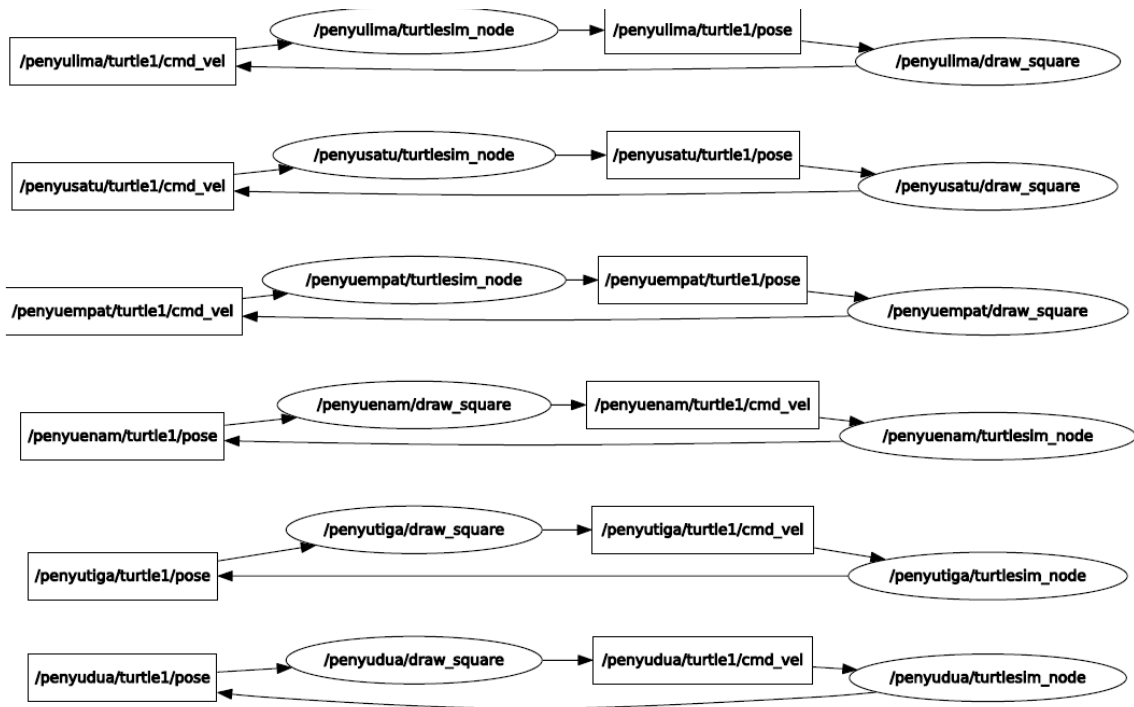
Aturan Penilaian (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-08	CPMK-05	Selesaikan langkah praktikum	Hasil praktikum langkah	100

5.7. POST TEST

1. Dengan menggunakan perintah-perintah yang telah anda pelajari pada unit ini dan unit-unit sebelumnya buatlah suatu package ROS yang terdiri dari node "pembicara" sebagai publisher dan node "pendengar" sebagai subscriber dengan message yang dikirimkan adalah tulisan "Nama dan NIM anda". (sesuaikan dengan nama dan NIM anda).
2. Dengan menggunakan roslaunch buatlah suatu file launch pada folder /catkin_ws/src dan jalankan melalui dari folder tersebut agar diperoleh enam turtlesim yang masing-masing


digerakkan oleh node draw_square.. Dengan menggunakan rqt_graph tunjukkan bahwa koneksi sebagaimana gambar berikut telah terjadi.



No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-08	CPMK-05	Silahkan kerjakan soal diatas	100

5.8. HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.



No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-03	CPMK-01	20%		
2.	Praktik	CPL-03	CPMK-01	30%		
3.	Post-Test	CPL-03	CPMK-01	50%		
Total Nilai						

PRAKTIKUM 6: RVIZ PADA ROBOT OPERATING SYSTEM

Pertemuan ke : 6

Total Alokasi Waktu : 90 menit

- Materi : 15 menit
- Pre-Test : 15 menit
- Praktikum : 45 menit
- Post-Test : 15 menit

Total Bobot Penilaian : 100%

- Pre-Test : 20 %
- Praktik : 30 %
- Post-Test : 50 %

Pemenuhan CPL dan CPMK:

CPL-08	Mampu memilih, membuat dan menerapkan teknik, sumber daya, penggunaan perangkat teknik modern dan implementasi teknologi informasi untuk memecahkan masalah
CPMK-05	Mampu mengimplementasikan low level control pada robot dan pemrogramannya, High Level control pada robot dan pemrogramannya, Pemanfaatan mesin Visi dan sistem cerdas pada Robot

6.1. DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

5. Menjelaskan rviz pada *robot operating system*.
6. Menerapkan rviz pada *robot operating system*.

6.2. INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-08	CPMK-05	Mampu membangun dan mempresentasikan sistem robot
--------	---------	---

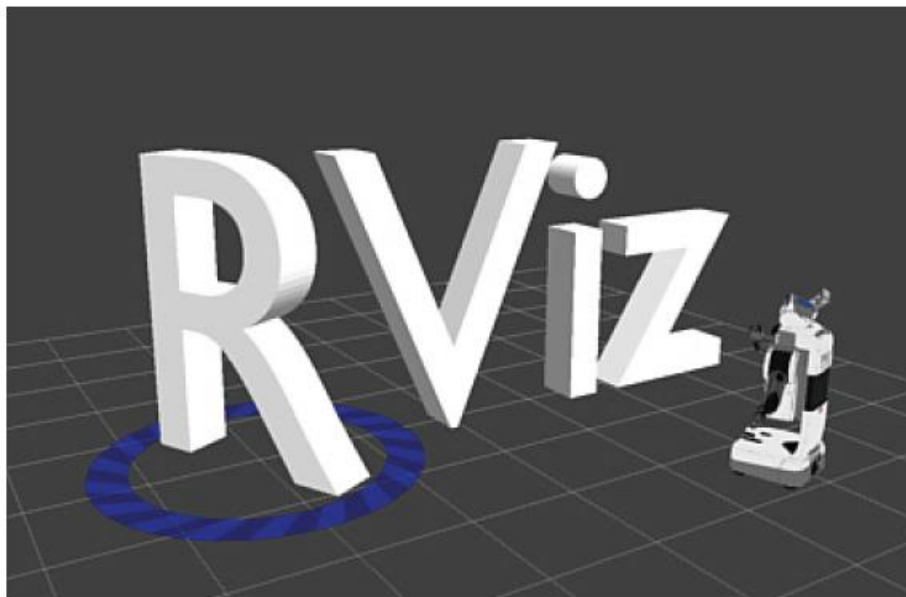
6.3. TEORI PENDUKUNG

RViz

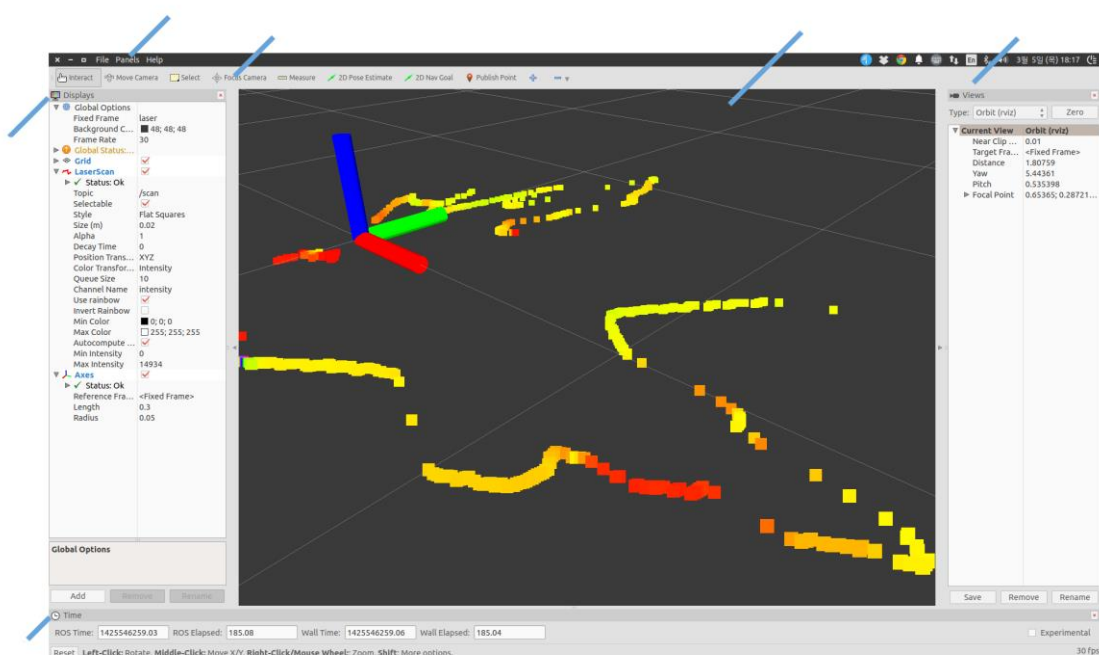
Rviz adalah perangkat visualisasi 3D yang terdapat pada ROS. Kegunaan utama Rviz adalah untuk menampilkan message(informasi dalam ROS) dalam bentuk 3D sehingga memungkinkan pengguna untuk memverifikasi data informasi tersebut secara visual. Rviz dapat digunakan untuk memvisualisasikan informasi jarak yang diperoleh dari pengukuran menggunakan Laser Distance Sensor(LDS), dapat juga memvisualisasikan data Point Cloud yang diperoleh dari suatu kamera 3D

atau sensor 3D semisal Realsense, Kinect ataupun Xtion. Demikian Halnya data Visual /Citra yang diperoleh dari suatu kamera dapat divisualisasi oleh Rviz tanpa perlu adanya pengembangan software tersendiri di luar Rviz.

Kegunaan lain dari Rviz yang tidak kalah penting adalah kemampuannya untuk memvisualisasikan suatu tanda interactive yang diperoleh dari masukan oleh pengguna yang Selanjutnya dapat digunakan sebagai dasar gerakan tindakan tertentu oleh robot. Selain itu Rviz dapat digunakan untuk memvisualisasikan bentuk 3D suatu robot dengan menggunakan Unified Robot Description Format(URDF), tidak hanya Untuk visualisasi model ini juga dapat digunakan Sebagai media interaksi antara pengguna dengan robot dikarenakan bagian ataupun sendi yang terdapat pada model robot berbasis URDF dapat digerakkan secara interaktif. Tampilan Awal Rviz dapat dilihat pada berikut.



Layout Rviz setelah dibuka akan seperti gambar berikut



3Dview: area hitam yang berada di tengah layar /tampilan Yang merupakan tampilan Utama yg memvisualisasikan berbagai data dalam bentuk 3D. Warna latar belakang, Ukuran grid(Yang mewakili jarak tertentu) dapat diatur Sesuai kebutuhan pengguna.

Display: merupakan Suatu panel yg berada pada bagian kiri dari .tampilan dan digunakan untuk memilih data mana yang akan ditampilkan

Menu: digunakan untuk menyimpan atau membuka setting tampilan View panel ini digunakan untuk mengatur arah/ Sudut pandang dalam 3D. Mengubah variable yang digunakan ketika menjalankan suatu node atau sejumlah node.

6.4. HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

1. Laptop Ubuntu dengan ROS Kinetic

6.5. PRE-TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-03	CPMK-01		
2.	Dst			

6.6. LANGKAH PRAKTIKUM

1. Menampilkan Robot menggunakan RViz

Buka Terminal baru dan ketikkan perintah berikut :

```
$ cd Downloads/
```

Tekan tombol Enter. Masih pada terminal yang sama lanjutkan ketik perintah berikut (pastikan laptop anda terkoneksi internet):

```
$ git clone https://github.com/ROBOTIS-GIT/turtlebot3.git
```

Tekan tombol Enter. Tunggu sampai proses selesai (100%). Masih pada terminal yang sama lanjutkan ketik perintah berikut:

```
$ cp -R ~/Downloads/turtlebot3/turtlebot3_description ~/catkin_ws/src/turtlebot3_description
```

Tekan tombol Enter.

Setelah rangkaian perintah tersebut dijalankan, sebuah folder bernama turtlebot3_description akan muncul pada direktori catkin_ws/src. Masih pada terminal yang sama ketikka perintah berikut.

```
$ cd ~/catkin_ws/
```

Tekan tombol Enter. Masih pada Terminal tersebut ketikkan perintah berikut

```
$ catkin_make
```


Tekan tombol Enter.

Jika semua telah anda atur dengan benar maka akan terjadi kompilasi tanpa error.

Lanjutkan pada terminal yang sama perintah berikut:

```
$ cd ~/catkin_ws/src
```

Tekan tombol Enter. Masih pada Terminal tersebut ketikkan perintah berikut

```
$ gedit tb3_burger_rviz.launch
```

Tekan tombol Enter. Pada file yang terbuka lakukan perubahan agar file tersebut berisi tulisan berikut.

```
<launch>

<arg name="model" default="$(find turtlebot3_description)/urdf/
turtlebot3_burger.urdf.xacro"/>
<arg name="gui" default="true" />
<arg name="rvizconfig" default="$(find turtlebot3_description)/
rviz/model.rviz" />

<param name="robot_description" command="$(find xacro)/xacro --
inorder $(arg model)" />
<param name="use_gui" value="$(arg gui)"/>

<node name="joint_state_publisher" pkg="joint_state_publisher"
type="joint_state_publisher" />
<node name="robot_state_publisher" pkg="robot_state_publisher"
type="state_publisher" />
<node name="rviz" pkg="rviz" type="rviz" args="-d $(arg rvizcon
fig)" required="true" />

</launch>
```

Simpan perubahan dan tutup file tersebut.

Buka terminal baru dan ketikkan perintah berikut

```
$ roscore
```

Tekan Enter. Buka terminal baru dan ketikkan perintah berikut

```
$ cd ~/catkin_ws/src
```

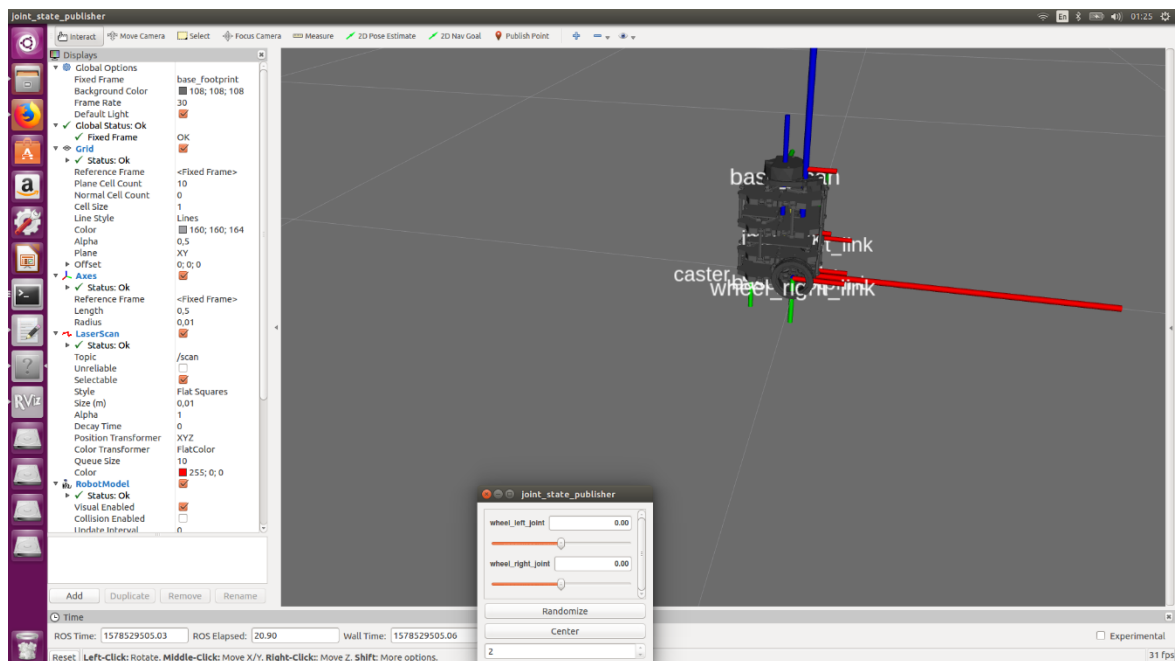
Tekan Enter. Pada terminal yang sama ketikkan perintah

```
$ source ~/catkin_ws/devel/setup.bash
```

Tekan Enter. Pada terminal yang sama ketikkan perintah

```
$ roslaunch tb3_burger_rviz.launch
```

Tekan Enter. Jika semua telah dilakukan dengan benar maka akan diperoleh tampilan sebagai berikut



Dengan menggunakan trackbar pada joint_state_publisher kita dapat memutar roda yang terdapat pada robot. Gunakan tombol printscreen menyimpan tampilan Rviz anda simpan sebagai NIM-unit6-tugas01.

2. Melakukan Simulasi Robot pada RViz

Tutup semua terminal yang sebelumnya digunakan. Pastikan semua telah tertutup.

Buka Terminal baru dan ketikkan perintah berikut :

```
$ cd Downloads/
```

Tekan tombol Enter. Masih pada terminal yang sama lanjutkan ketik perintah berikut :

```
$ cp -R ~/Downloads/turtlebot3/turtlebot3_teleop ~/catkin_ws/src/turtlebot3_teleop
```

Tekan tombol Enter. Masih pada terminal yang sama lanjutkan ketik perintah berikut :

```
$ cp -R ~/Downloads/turtlebot3/turtlebot3_bringup ~/catkin_ws/src/turtlebot3_bringup
```

Tekan tombol Enter. Masih pada terminal yang sama lanjutkan ketik perintah berikut (pastikan laptop anda terkoneksi internet) :

```
$ git clone https://github.com/ROBOTIS-GIT/turtlebot3_msgs.git
```

Tekan tombol Enter. Tunggu sampai proses selesai (100%). Masih pada terminal yang sama lanjutkan ketik perintah berikut:

```
$ cp -R ~/Downloads/turtlebot3_msgs ~/catkin_ws/src/turtlebot3_msgs
```

Tekan tombol Enter. Masih pada terminal yang sama lanjutkan ketik perintah berikut (pastikan laptop anda terkoneksi internet) :

```
$ git clone https://github.com/ROBOTIS-GIT/turtlebot3_simulations.git
```

Tekan tombol Enter. Tunggu sampai proses selesai (100%). Masih pada terminal yang sama lanjutkan ketik perintah berikut:

```
$ cd turtlebot3_simulations
```

Tekan tombol Enter. Masih pada terminal yang sama lanjutkan ketik perintah berikut :

```
$ ls
```

Tekan Enter. Pastikan anda peroleh hasil berikut

```
LICENSE      README.md      turtlebot3_fake      turtlebot3_gazebo
              turtlebot3_simulations
```

Masih pada terminal yang sama lanjutkan ketik perintah berikut :

```
$ rm -R turtlebot3_gazebo/
```

Tekan tombol Enter. Masih pada terminal yang sama lanjutkan ketik perintah berikut :

```
$ ls
```

Tekan Enter. Pastikan anda peroleh hasil berikut

```
LICENSE  README.md  turtlebot3_fake  turtlebot3_simulations
```

Masih pada terminal yang sama lanjutkan ketik perintah berikut :

```
$ cp -R ~/Downloads/turtlebot3_simulations ~/catkin_ws/src/turtlebot3_simulations
```

Tekan tombol Enter. Masih pada terminal yang sama lanjutkan ketik perintah berikut :

```
$ cd ~/catkin_ws
```

Tekan tombol Enter. Masih pada terminal yang sama lanjutkan ketik perintah berikut :

```
$ catkin_make
```

Tekan tombol Enter. Tunggu proses kompilasi selesai tanpa error. Masih pada terminal yang sama lanjutkan ketik perintah berikut :

```
$ source ~/catkin_ws/devel/setup.bash
```

Tekan tombol Enter. Selanjutnya buka Terminal baru dan ketikkan perintah berikut :

```
$ roscore
```

Tekan tombol Enter. Selanjutnya buka Terminal baru dan ketikkan perintah berikut :

```
$ export TURTLEBOT3_MODEL=burger
```

Tekan tombol Enter. Masih pada terminal yang sama lanjutkan ketik perintah berikut :

```
$ roslaunch turtlebot3_fake turtlebot3_fake.launch
```

Tekan tombol Enter. Selanjutnya buka Terminal baru dan ketikkan perintah berikut :

```
$ rosrun turtlebot3_teleop turtlebot3_teleop_key
```

Tekan tombol Enter. Selanjutnya coba jalankan robot dengan tombol sesuai petunjuk yang muncul, amati pergerakan robot pada RViz. Pastikan terminal turtlebot3_teleop_key aktif saat anda menekan tombol. Setelah beberapa pergerakan dengan tombol teleop, gunakan tombol printscreen untuk menyimpan tampilan Rviz anda simpan sebagai NIM-unit6-tugas02.

3. Menjalankan robot dengan node pengendali (controller)

Tutup semua terminal yang sebleumnya digunakan. Pastikan semua telah tertutup.

Buka Terminal baru dan ketikkan perintah berikut :

```
$ cd Downloads/
```

Tekan tombol Enter. Masih pada terminal yang sama lanjutkan ketik perintah berikut (pastikan laptop anda terkoneksi internet) :

```
$ git clone https://github.com/nuryonosw/tb3_controller.git
```

Tekan tombol Enter. Tunggu sampai proses selesai (100%). Masih pada terminal yang sama lanjutkan ketik perintah berikut:

```
$ cp -R ~/Downloads/tb3_controller ~/catkin_ws/src/tb3_controller
```

Tekan tombol Enter. Masih pada terminal yang sama lanjutkan ketik perintah berikut :

```
$ cd ~/catkin_ws
```

Tekan tombol Enter. Masih pada terminal yang sama lanjutkan ketik perintah berikut :

```
$ catkin_make
```

Tekan tombol Enter. Tunggu proses kompilasi selesai tanpa error. Masih pada terminal yang sama lanjutkan ketik perintah berikut :

```
$ source ~/catkin_ws/devel/setup.bash
```

Tekan tombol Enter. Selanjutnya buka Terminal baru dan ketikkan perintah berikut :

```
$ roscore
```

Tekan tombol Enter. Selanjutnya buka Terminal baru dan ketikkan perintah berikut :

```
$ export TURTLEBOT3_MODEL=burger
```

Tekan tombol Enter. Masih pada terminal yang sama lanjutkan ketik perintah berikut :

```
$ roslaunch turtlebot3_fake turtlebot3_fake.launch
```

Tekan tombol Enter. Selanjutnya buka Terminal baru dan ketikkan perintah berikut :

```
$ rosrun tb3_controller tb3_controller_node
```

Tekan tombol Enter. Selanjutnya coba masukkan nilai-nilai berikut diikuti penekanan tombol Enter.

Speed=1

Distance=1

isForward=1

angular_speed=5

angle=90

clockwise=1

Tunggu sampai pergerakan selesai. Setelah pergerakan selesai, gunakan tombol printscreen untuk menyimpan tampilan Rviz anda simpan sebagai NIM-unit6-tugas03.

Aturan Penilaian (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-08	CPMK-05	Selesaikan langkah praktikum	Hasil praktikum langkah	100

6.7. POST TEST

1. Dengan menggunakan `roslaunch` buatlah dua file launch pada folder `/catkin_ws/src` dan jalankan melalui dari folder tersebut agar diperoleh tampilan `turtlebot3_waffle` dan `turtlebot3_waffle_for_open_manipulator` pada dua file launch terpisah (`tb3_waffle_rviz.launch` dan `tb3_wafflemanipulator_rviz.launch`). Gunakan tombol printscreen untuk menyimpan tampilan Rviz anda untuk kedua robot tersebut. (petunjuk: perhatikan isi `tb3_burger_rviz.launch` ikuti pola penulisannya untuk menyusun `tb3_waffle_rviz.launch` dan `tb3_wafflemanipulator_rviz.launch`)
2. Lakukan perubahan pada `tb3_controller_node.cpp` agar anda bisa memperoleh pergerakan robot berupa bujur sangkar dengan sisi 1 m, dengan satu kali memasukkan parameter. Lakukan kompilasi ulang dan coba jalankan serta simpan tampina Rviz yang diperoleh.

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-08	CPMK-05	Silahkan kerjakan soal diatas.	100

6.8. HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-03	CPMK-01	20%		
2.	Praktik	CPL-03	CPMK-01	30%		
3.	Post-Test	CPL-03	CPMK-01	50%		
Total Nilai						

PRAKTIKUM 7: PENGENALAN SIMULASI MIKROKONTROLLER

Pertemuan ke : 7

Total Alokasi Waktu : 90 menit

- Materi : 15 menit
- Pre-Test : 15 menit
- Praktikum : 45 menit
- Post-Test : 15 menit

Total Bobot Penilaian : 100%

- Pre-Test : 20 %
- Praktik : 30 %
- Post-Test : 50 %

Pemenuhan CPL dan CPMK:

CPL-05	Mampu mengkaji / menganalisis implikasi pengembangan atau implementasi ilmu pengetahuan teknologi, menyusun deskripsi saintifik hasil kajian untuk pemecahan masalah dengan mempertimbangkan multidisiplin ilmu
CPMK-02	Jenis Penggerak yg dipergunakan robot

7.1. DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

1. Menjelaskan apa itu pemrograman mikrokontroller.
2. Menerapkan pemrograman mikrokontroller.

7.2. INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

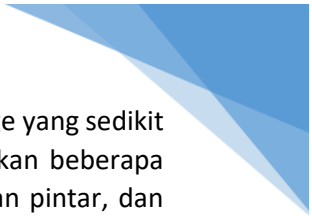
CPL-05	CPMK-02	Mampu membangun dan mempresentasikan sistem mikrokontroller
--------	---------	---

7.3. TEORI PENDUKUNG

Apa itu Arduino?

Arduino berguna untuk merancang, memproduksi, dan mendukung perangkat elektronik dan perangkat lunak, memungkinkan orang di seluruh dunia dengan mudah mengakses teknologi canggih yang berinteraksi dengan dunia fisik. Produk ini mudah, sederhana, dan kuat, siap untuk memenuhi kebutuhan pengguna mulai dari pelajar hingga pembuat dan hingga pengembang profesional.

Arduino sendiri adalah sebuah perangkat purwarupa (prototyping) yang bersifat open source. Arduino ini memiliki beberapa komponen penting di dalamnya, seperti pin, mikrokontroler, dan



konektor. Arduino juga sudah menggunakan bahasa pemrograman Arduino Language yang sedikit mirip dengan bahasa pemrograman C++. Arduino digunakan untuk mengembangkan beberapa sistem seperti pengatur suhu, sensor untuk bidang agrikultur, pengendali peralatan pintar, dan masih banyak lagi.

Komponen Arduino

Arduino ini memiliki beberapa komponen di dalamnya.

- Mikrokontroler

Mikrokontroler merupakan chip yang memungkinkan Arduino deprogram dan memproses output berdasarkan input yang diberikan. Mikrokontroler ini adalah otak dari Arduino. Terdapat banyak jenis chip yang digunakan bergantung dari jenis Arduino-nya.

- Pin

Pin ini digunakan untuk menghubungkan Arduino dengan berbagai komponen eksternal yang akan digunakan. Pada Arduino secara umum terdapat dua jenis pin, yakni pin analog dan pin digital.

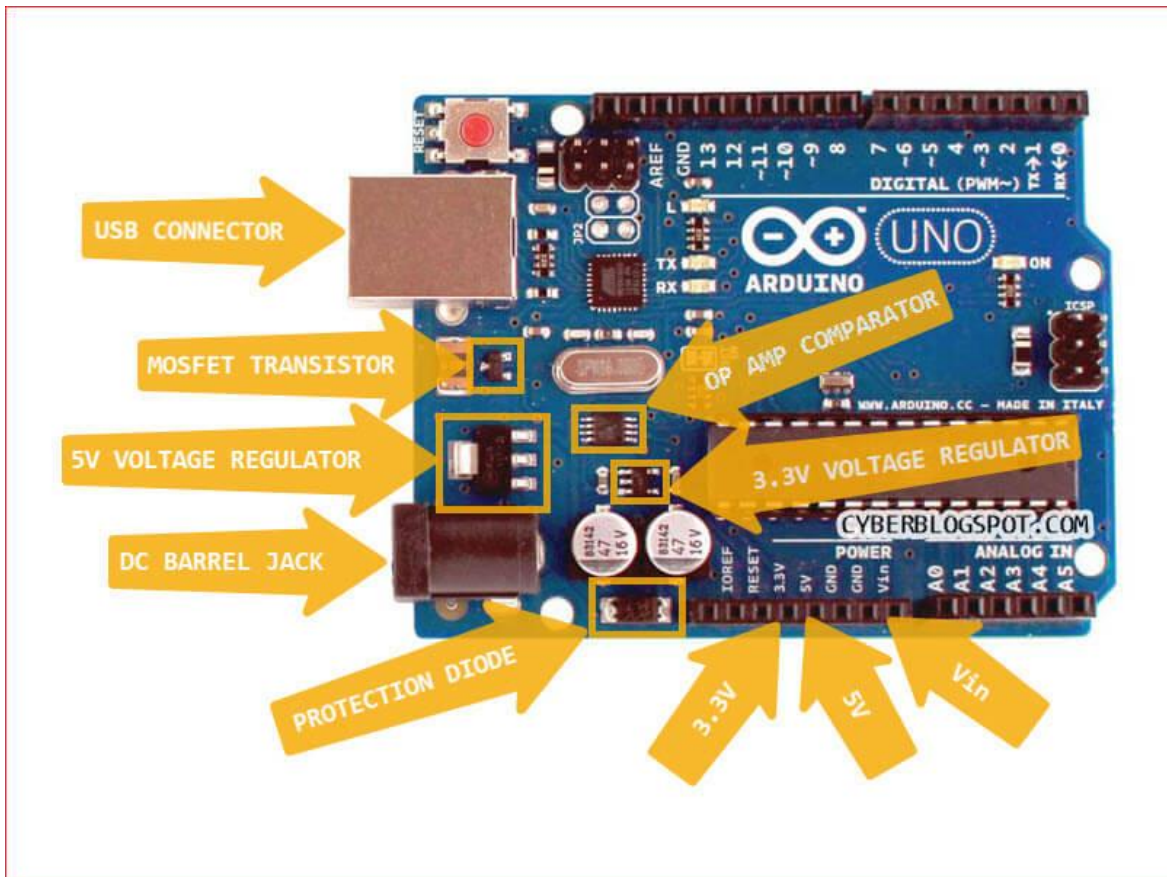
- Pin digital

- Pin digital ini dapat menerima maupun mengirim sinyal digital. Digital berarti sinyal yang bernilai 1 atau 0 alias HIGH atau LOW.

- Pin analog

- Pin analog pada arduino adalah pin yang digunakan untuk menerima input analog. Ia dapat menerima tegangan analog dari 0V sampai dengan 5V.

Setiap pin pada Arduino biasanya dapat dikonfigurasi ke dalam dua mode, yaitu input dan output. Pada mode input, pin akan diatur untuk dapat menerima sinyal input. Sama halnya pada mode output, pin akan diatur untuk mengirimkan sinyal.



Sumber <https://cyberblogspot.com/arduino-power-supply-schematic/>

- Konektor
 - Arduino sendiri memiliki dua jenis konektor, yaitu power konektor dan serial konektor.
 - Power konektor
 - Power konektor adalah konektor yang digunakan untuk menyalurkan daya untuk Arduino. Daya ini digunakan untuk menghidupkan Arduino dan juga perangkat lain yang terhubung dengannya, seperti sensor dan layar monitoring.
 - Serial konektor
 - Serial konektor ini biasanya digunakan untuk menghubungkan Arduino dengan perangkatmu seperti komputer atau laptop. Konektor ini menggunakan port USB standar pada Arduino. Selain itu, konektor ini juga dapat digunakan sebagai power konektor. Namun, serial konektor hanya diimplementasikan pada perangkat Arduino yang lebih baru.

7.4. HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

1. Laptop
2. Browser Online
3. <https://wokwi.com/>

7.5. PRE-TEST

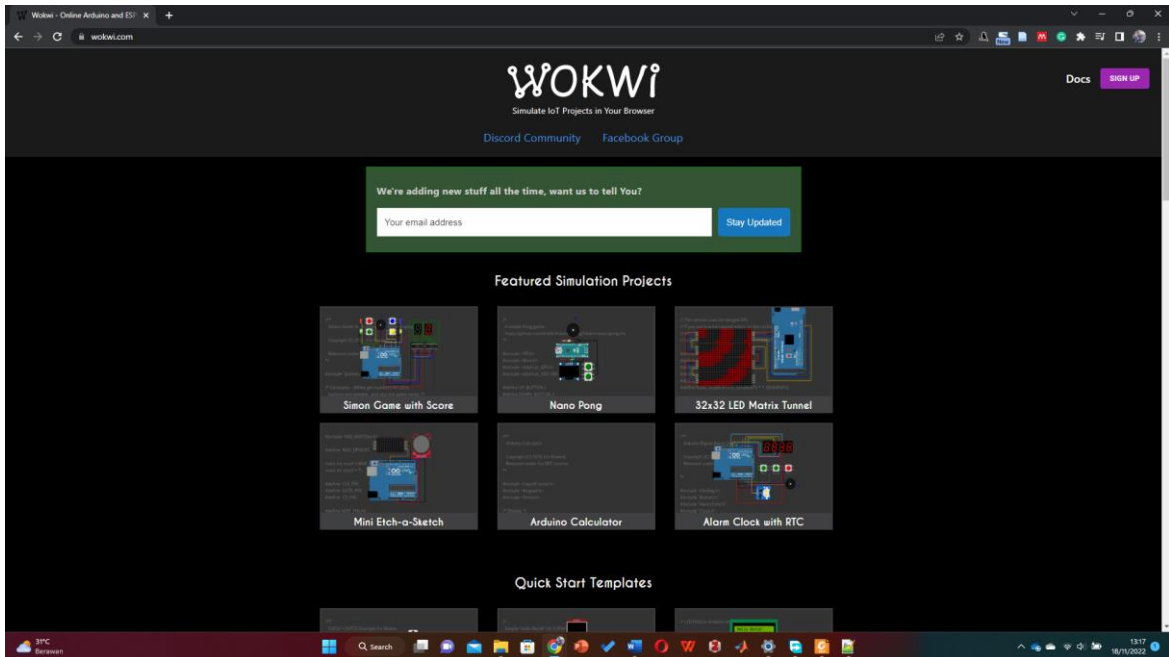
Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-05	CPMK-02	Apa itu mikrokontroler?	50
2.	CPL-05	CPMK-02	Sebutkan jenis mikrokontroler yang anda ketahui!	50

7.6. LANGKAH PRAKTIKUM

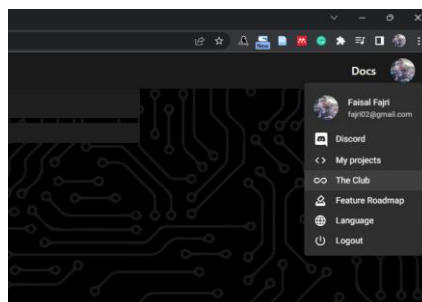
1. Buka Browser

Silahkan buka alamat <https://wokwi.com/>, alamat ini digunakan untuk simulasi menggunakan mikrokontroler.



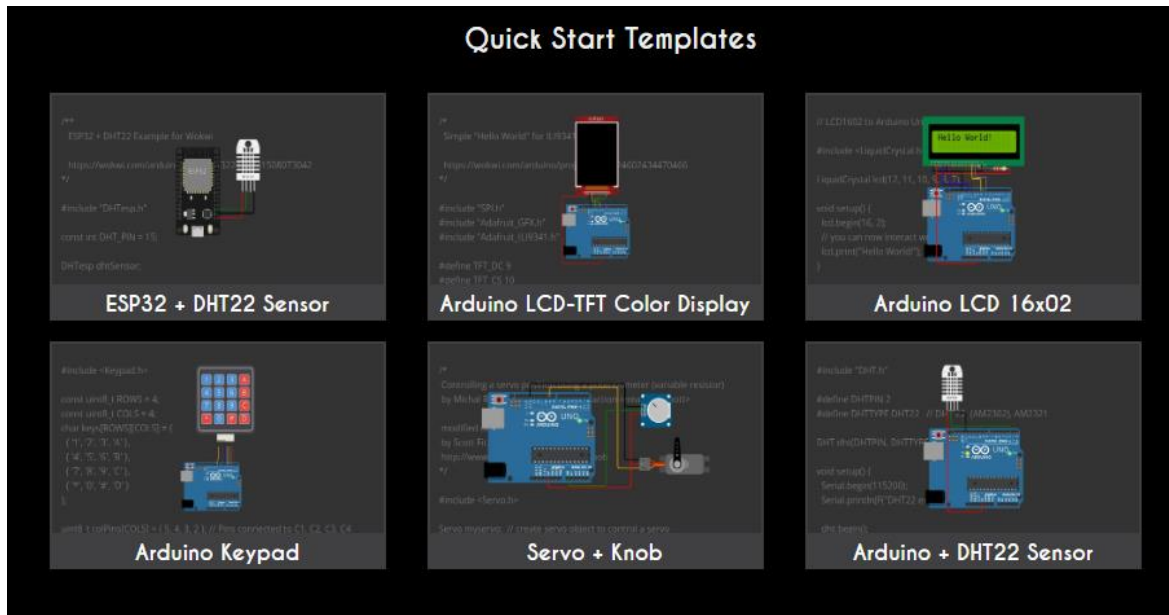
2. Silahkan sign Up menggunakan akun yang ada.

Silahkan anda melakukan pendaftaran akun. Pendaftaran akun ini berguna untuk menyimpan project anda secara online.

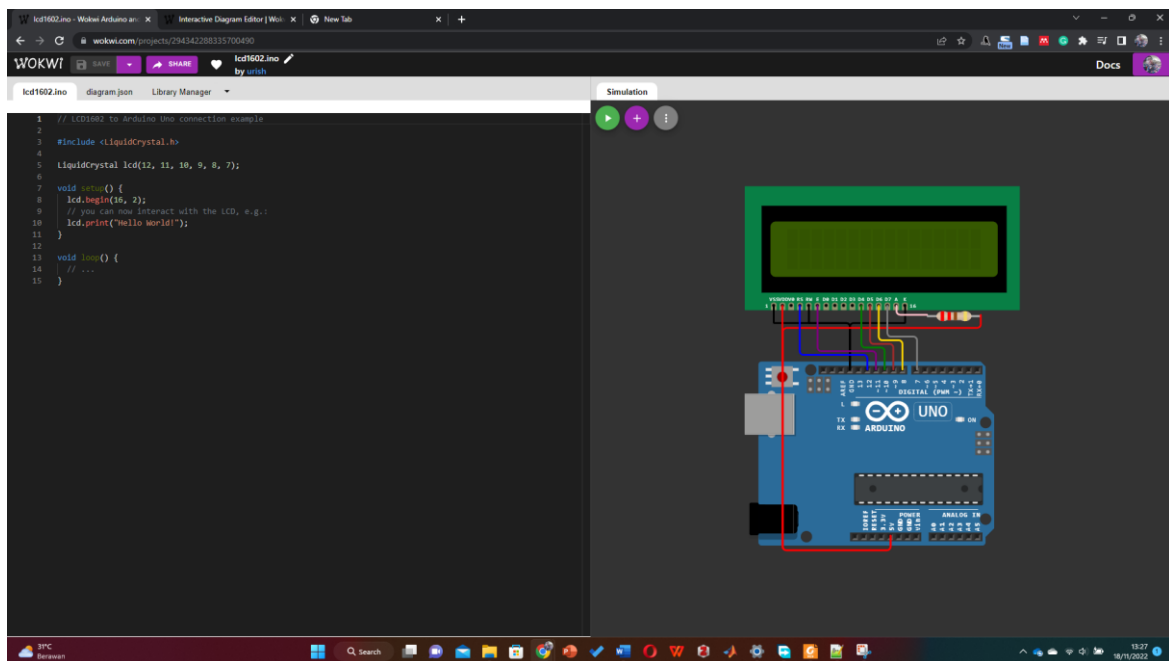


Perhatikan tampilan yang anda peroleh, tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit7-tugas01.

3. Mencoba membuat dari template



Silahkan tuju pada bagian Quick Start Template, pilih arduino LCD 16x02.



Silahkan pelajari masing- masing bagian. Bagian kanan merupakan wiring yang digunakan. Pada bagian tersebut menggunakan pin power dan pin data. Pada bagian kiri merupakan program yang digunakan.

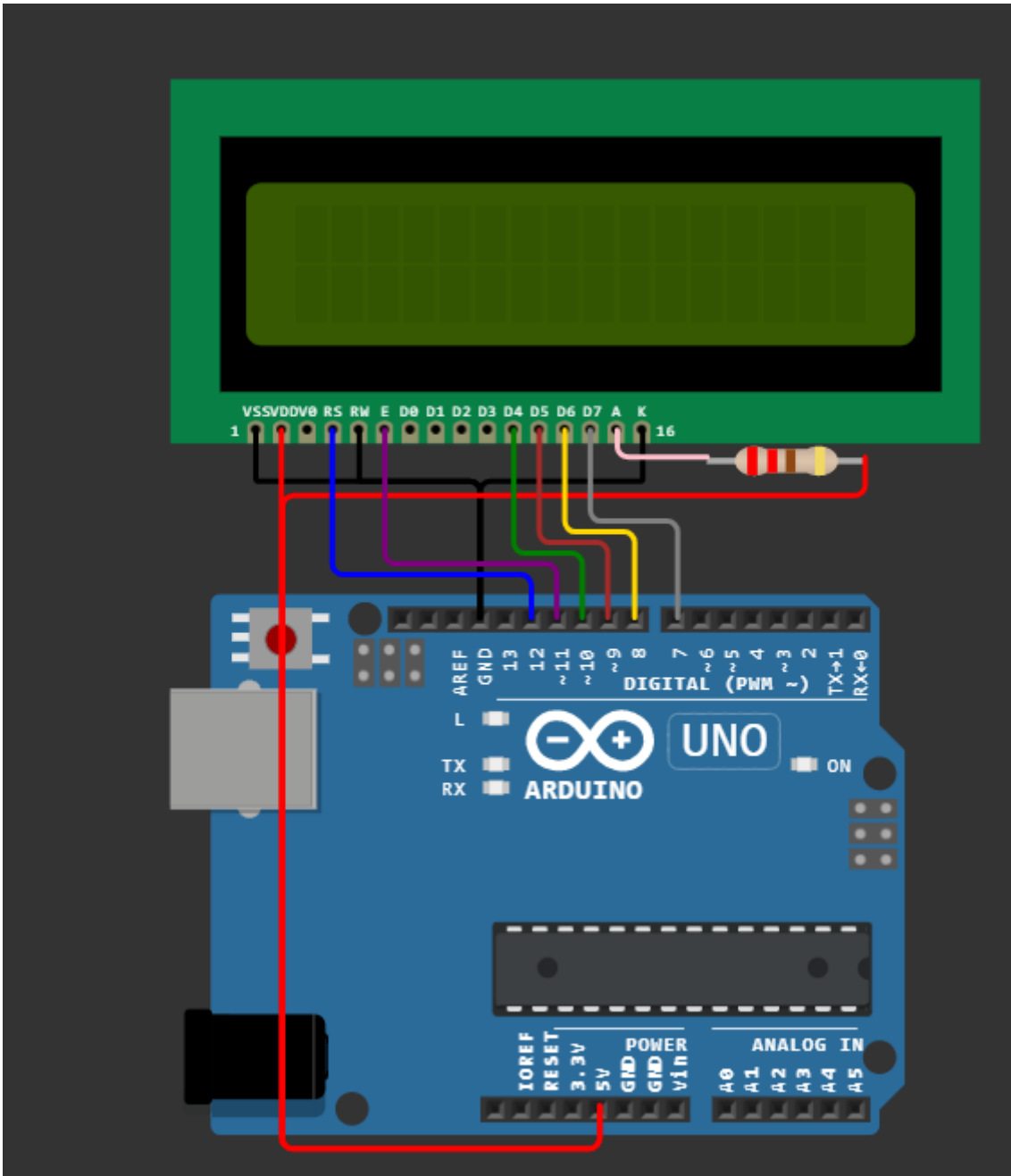


Diagram yang digunakan untuk perangkat keras LCD 16x02.

```

1 // LCD1602 to Arduino Uno connection example
2
3 #include <LiquidCrystal.h>
4
5 LiquidCrystal lcd(12, 11, 10, 9, 8, 7);
6
7 void setup() {
8     lcd.begin(16, 2);
9     // you can now interact with the LCD, e.g.:
10    lcd.print("Hello World!");
11 }
12
13 void loop() {
14     // ...
15 }

```

Listing program untuk memunculkan Hello world di LCD.

4. Mencoba mengubah project yang ada.

Silahkan anda variasikan program yang sudah ada diatas untuk skenario berikut:

- a. Munculkan nama anda.
- b. Delay 5 detik.
- c. Munculkan NIM anda.
- d. Delay 5 Detik.
- e. Munculkan email anda.
- f. Delay 5 detik.
- g. Silahkan dilakukan secara berulang.

Perhatikan tampilan yang anda peroleh, tekan tombol PrintScreen pada Laptop anda pada bagian listing program dan simpan dengan nama NIM-unit7-tugas02.

5. Mencoba project yang ada.

Silahkan coba project lain yang terdapat pada web tersebut minimal 3 project. Perhatikan tampilan yang anda peroleh, tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit7-tugas03,04, dan 05.

Aturan Penilaian (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-05	CPMK-02	Selesaikan langkah 1-3	Hasil praktikum langkah	100

7.7. POST TEST

Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-05	CPMK-02	Selesaikan langkah 4	60
2.	CPL-05	CPMK-02	Selesaikan langkah 5	40

7.8. HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-05	CPMK-02	10%		
2.	Praktik	CPL-05	CPMK-02	50%		
3.	Post-Test	CPL-05	CPMK-02	40%		
Total Nilai						

PRAKTIKUM 8: PENGENALAN SIMULASI STEPPER MOTOR

Pertemuan ke : 8

Total Alokasi Waktu : 90 menit

- Materi : 15 menit
- Pre-Test : 15 menit
- Praktikum : 45 menit
- Post-Test : 15 menit

Total Bobot Penilaian : 100%

- Pre-Test : 10 %
- Praktik : 50 %
- Post-Test : 40 %

Pemenuhan CPL dan CPMK:

CPL-05	Mampu mengkaji / menganalisis implikasi pengembangan atau implementasi ilmu pengetahuan teknologi, menyusun deskripsi saintifik hasil kajian untuk pemecahan masalah dengan mempertimbangkan multidisiplin ilmu
CPMK-03	Motor DC, metode pengendalian dan pemrogramannya

8.1. DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

1. Menjelaskan apa itu pemrograman mikrokontroler.
2. Menerapkan pemrograman mikrokontroler.

8.2. INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-05	CPMK-03	Mampu membangun dan mempresentasikan macam- macam sistem motor stepper
--------	---------	--

8.3. TEORI PENDUKUNG

Motor stepper, juga dikenal sebagai motor langkah atau motor langkah, adalah motor listrik DC tanpa sikat yang membagi putaran penuh menjadi beberapa langkah yang sama. Posisi motor dapat diperintahkan untuk bergerak dan bertahan di salah satu langkah ini tanpa sensor posisi untuk umpan balik (pengontrol loop terbuka), asalkan ukuran motor benar untuk aplikasi sehubungan dengan torsi dan kecepatan.

Motor keengganan yang diaktifkan adalah motor stepping yang sangat besar dengan jumlah kutub yang berkurang, dan umumnya dikomutasikan loop tertutup.

Motor stepper, karena desainnya yang unik, dapat dikontrol dengan tingkat akurasi yang tinggi tanpa mekanisme umpan balik. Poros stepper, dipasang dengan serangkaian magnet, dikendalikan oleh serangkaian kumparan elektromagnetik yang bermuatan positif dan negatif dalam urutan tertentu, dengan tepat menggerakannya maju atau mundur dalam "langkah" kecil.

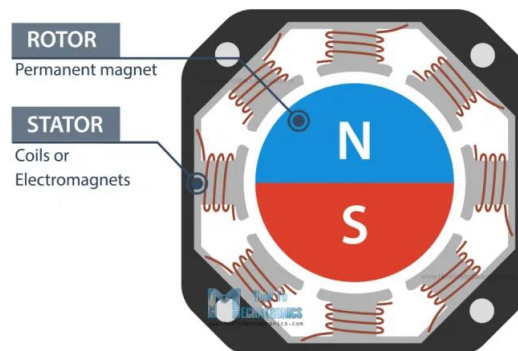
Ada dua jenis stepper, Unipolar dan Bipolar, dan sangat penting untuk mengetahui jenis yang Anda gunakan. Untuk masing-masing motor, ada sirkuit yang berbeda. Kode contoh akan mengontrol kedua jenis motor. Lihat skema motor unipolar dan bipolar untuk informasi tentang cara menyambungkan motor Anda.

Apa itu Motor Stepper dan Bagaimana Cara Kerjanya?

Bagian dalam motor Stepper NEMA 17 - Cara Kerjanya

Motor stepper adalah jenis motor DC brushless unik yang posisinya dapat dikontrol dengan tepat bahkan tanpa umpan balik.

Prinsip kerja motor stepper didasarkan pada medan magnet. Ini memiliki dua komponen utama, stator dan rotor. Rotor biasanya merupakan magnet permanen dan dikelilingi oleh beberapa kumparan pada stator.

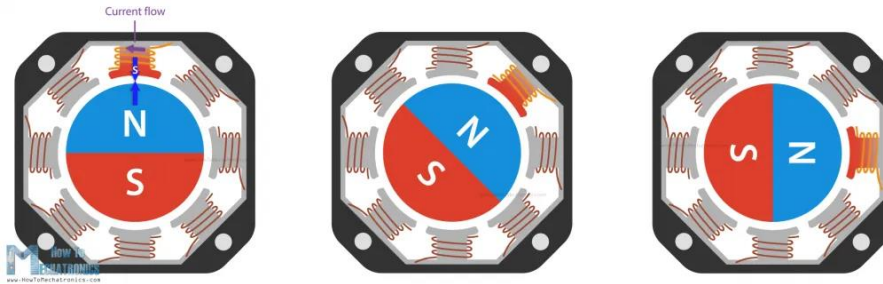


Sumber <https://howtomechatronics.com/tutorials/arduino/stepper-motors-and-arduino-the-ultimate-guide/>

Komponen utama Motor Stepper - stator dan rotor

Saat kita memberi energi atau membiarkan arus mengalir melalui kumparan, medan magnet tertentu dihasilkan di stator yang menarik atau menolak rotor. Dengan mengaktifkan kumparan, selangkah demi selangkah, satu demi satu dalam urutan tertentu, kita dapat mencapai gerakan rotor yang berkelanjutan, tetapi juga, kita dapat menghentikannya di posisi mana pun.

Prinsip Kerja Motor Stepper

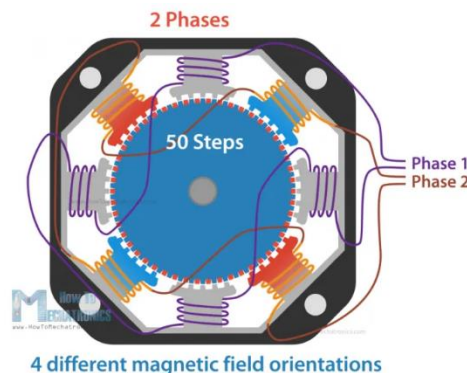


Sumber <https://howtomechatronics.com/tutorials/arduino/stepper-motors-and-arduino-the-ultimate-guide/>

Jadi, itulah mengapa motor ini disebut motor stepper, mereka bergerak dalam langkah-langkah terpisah.

Dengan menambah jumlah kutub magnet pada rotor, kita dapat menambah jumlah kemungkinan posisi berhenti, sehingga meningkatkan resolusi atau presisi motor. Harap dicatat bahwa ini hanyalah penjelasan dasar dan Anda dapat menemukan detail lebih lanjut di tutorial Cara Kerja Motor Stepper saya.

Motor stepper tipikal, NEMA17 misalnya, memiliki 50 titik henti atau langkah pada rotor. Di sisi lain, stator dapat memiliki beberapa kumparan yang diatur dalam dua fase yang memberikan empat orientasi atau posisi medan magnet yang berbeda.

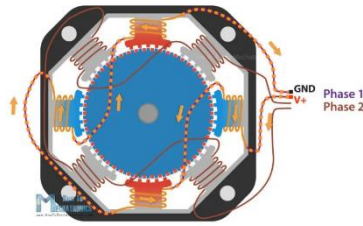


Sumber <https://howtomechatronics.com/tutorials/arduino/stepper-motors-and-arduino-the-ultimate-guide/>

Motor stepper - 50 Langkah, 2 fase dan 4 orientasi medan magnet berbeda

Jadi, 50 langkah rotor dikalikan dengan 4 orientasi medan magnet yang berbeda, menghasilkan total 200 langkah untuk menyelesaikan satu putaran penuh. Atau jika kita membagi 360 derajat dengan 200 langkah, itu adalah resolusi 1,8 derajat per langkah.

Kumparan stator diatur dalam dua fase, dan kita juga dapat memperhatikannya jika kita melihat jumlah kabel motor stepper. Ini memiliki 4 empat kabel, dua untuk setiap fase. Empat orientasi medan magnet yang berbeda dimungkinkan karena kita dapat membiarkan arus mengalir melalui fase di kedua arah.



Sumber <https://howtomechatronics.com/tutorials/arduino/stepper-motors-and-arduino-the-ultimate-guide/>

Arus mengalir melalui fase - prinsip kerja motor stepper

Ada juga motor stepper dengan 5, 6 atau bahkan 8 kabel, tetapi masih bekerja dalam dua fase atau kami mengontrolnya hanya dengan empat terminal.

Koneksi motor stepper 6-kawat dan 8-kawat dalam konfigurasi bi-polar

Masalahnya dengan mereka adalah mereka dapat memberikan karakteristik kinerja yang berbeda, seperti lebih banyak torsi atau lebih banyak kecepatan, tergantung pada bagaimana kita menghubungkan kabel ini pada empat terminal kontrol.

Namun demikian, dengan penjelasan singkat ini, sekarang kami memahami bahwa untuk menggerakkan motor stepper, kami tidak dapat begitu saja menyambungkan daya ke motor tersebut karena tidak akan terjadi apa-apa. Sebagai gantinya, kita harus memberi energi pada dua fase motor di kedua arah, dan mengaktifkan atau mengirim pulsa ke keduanya dalam urutan tertentu, dalam urutan yang tepat waktu. Nah oleh karena itu diperlukan driver untuk pengendalian motor stepper.

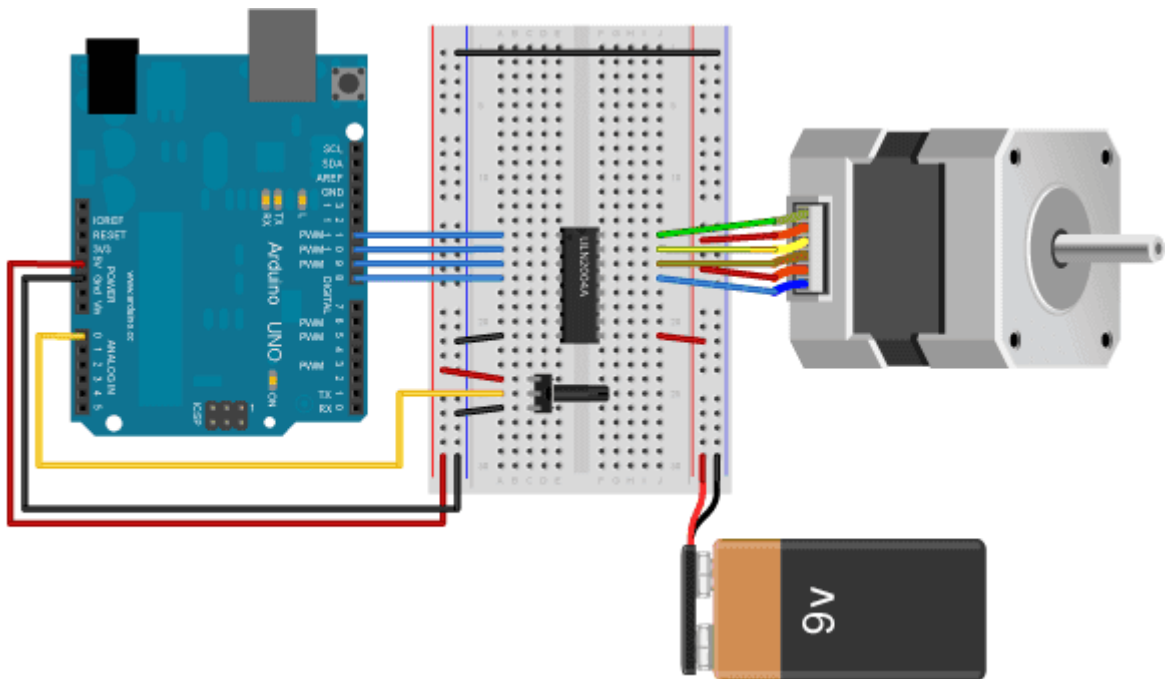
Ada banyak tipe dan ukuran driver, sesuai dengan banyaknya tipe dan ukuran motor stepper. Namun, prinsip kerja dasar dari semuanya adalah bahwa mereka memiliki dua H-Bridges yang memungkinkan memberi energi fase motor di kedua arah.

H-bridge di dalam driver motor stepper - cara kerjanya

Tentu saja, mereka memiliki banyak fungsi lain seperti langkah mikro, pembatas arus, dan sebagainya yang memungkinkan kita untuk dengan mudah mengontrol motor stepper, yang merupakan tujuan keseluruhannya.

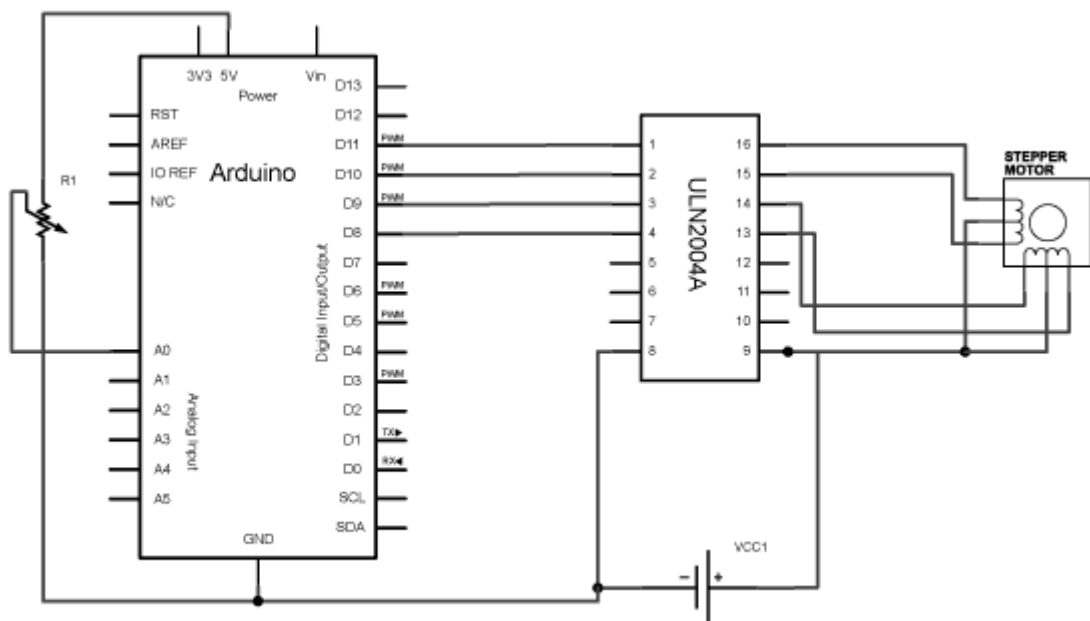
Stepper dikendalikan dengan pin digital 8, 9, 10, dan 11 untuk motor unipolar atau bipolar. Papan Arduino akan terhubung ke U2004 Darlington Array jika Anda menggunakan stepper unipolar atau SN754410NE H-Bridge jika Anda memiliki motor bipolar.

Di bawah ini sirkuit untuk stepper unipolar dan bipolar. Dalam kedua kasus tersebut, yang terbaik adalah memberi daya pada motor stepper Anda dari suplai eksternal, karena terlalu banyak daya untuk ditenagai langsung dari papan Arduino Anda.



Made with Fritzing.org

Sumber <https://docs.arduino.cc/learn/electronics/stepper-motors>



Made with Fritzing.org

Sumber <https://docs.arduino.cc/learn/electronics/stepper-motors>

8.4. HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

1. Laptop
2. Browser Online
3. <https://wokwi.com/>

8.5. PRE-TEST

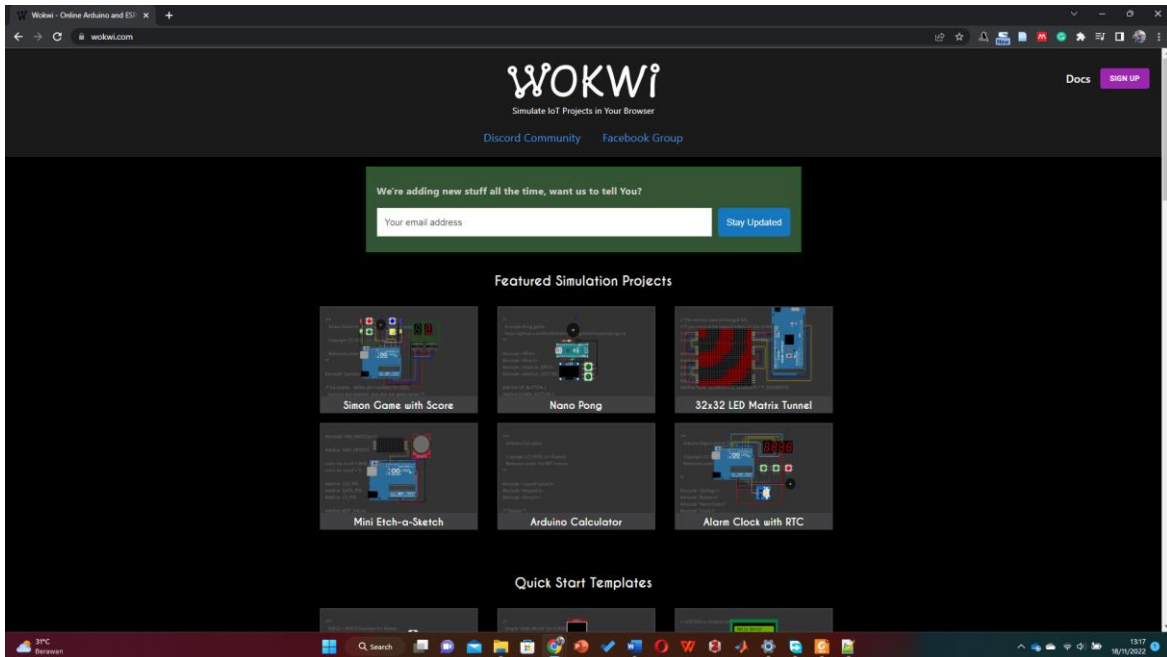
Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-05	CPMK-03	Sebutkan macam- macam jenis motor penggerak!	30
2.	CPL-05	CPMK-03	Apa itu motor stepper? Jelaskan cara kerjanya!	70

8.6. LANGKAH PRAKTIKUM

1. Buka Browser

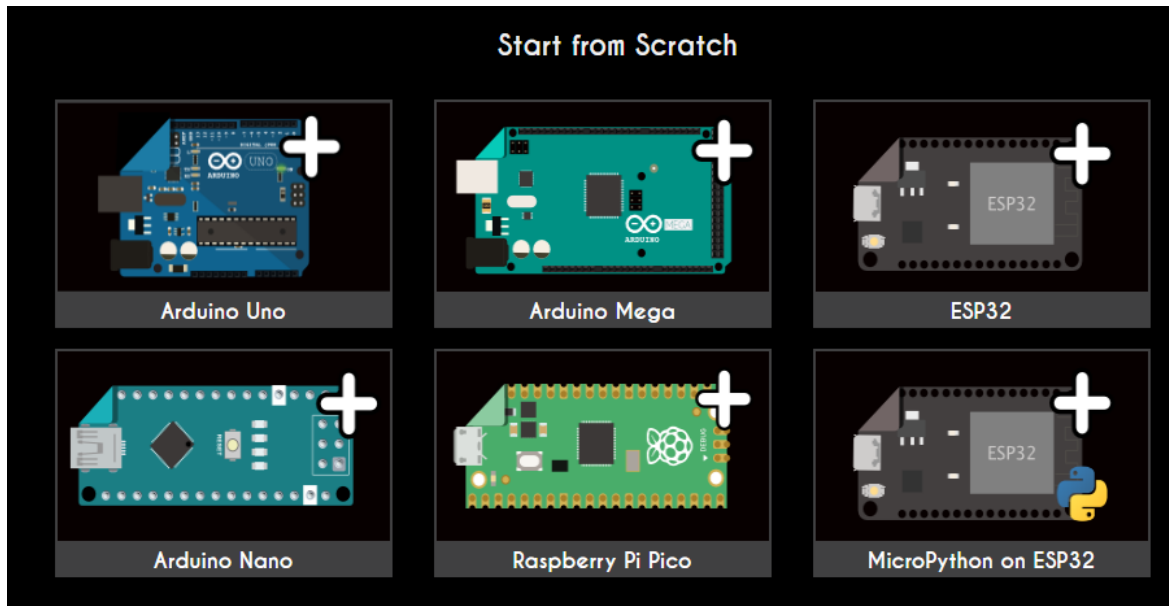
Silahkan buka alamat <https://wokwi.com/>, alamat ini digunakan untuk simulasi menggunakan mikrokontroler.



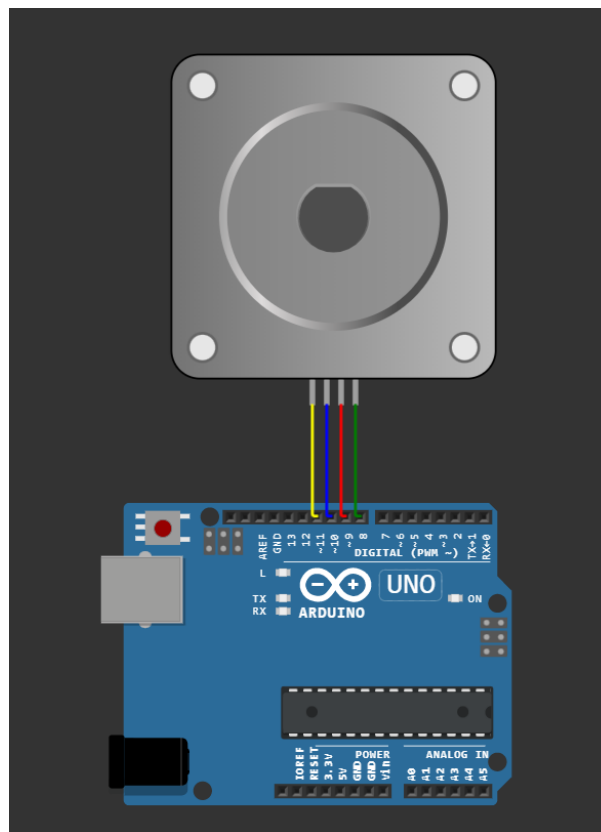
2. Silahkan sign In menggunakan akun yang ada.

Silahkan anda melakukan pendaftaran akun. Pendaftaran akun ini berguna untuk menyimpan project anda secara online.

3. Silahkan pilih start from scratch dengan mikrokontroler arduino



4. Silahkan buat diagram seperti dibawah ini dengan komponen stepper motor



Perhatikan pemasangan kabel yang ada dalam skema.

5. Silahkan tuliskan program sesuai dengan listing dibawah

```
// pins connected
#define Ap 10 // A+ line
#define Am 11 // A- line
#define Bp 9 // .
#define Bm 8 // .
```

```

void setup() {
  // set output
  pinMode(Ap,OUTPUT); pinMode(Am,OUTPUT);
  pinMode(Bp,OUTPUT); pinMode(Bm,OUTPUT);
}

int l = 0 ;
void loop() {
  // This is Wave Fullstep motion - only one coil energized at a time
  if ( l++ < 25 ) { // stop after 25*4 steps (=100) - quarter turn with gearRatio "2:1"
    delay(10);
    digitalWrite(Bm,LOW) ; digitalWrite(Ap,HIGH);
    delay(10);
    digitalWrite(Ap,LOW) ; digitalWrite(Bp,HIGH);
    delay(10);
    digitalWrite(Bp,LOW) ; digitalWrite(Am,HIGH);
    delay(10);
    digitalWrite(Am,LOW) ; digitalWrite(Bm,HIGH);
  } else while(1);
}

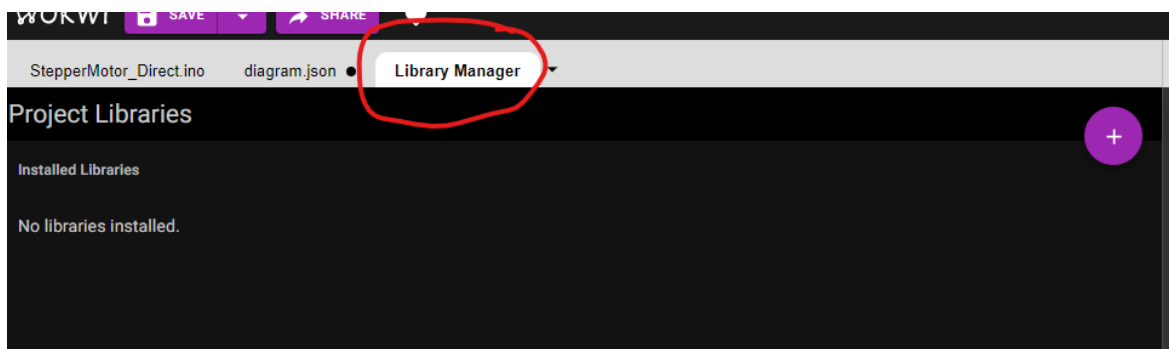
```

6. Jalankan project anda.

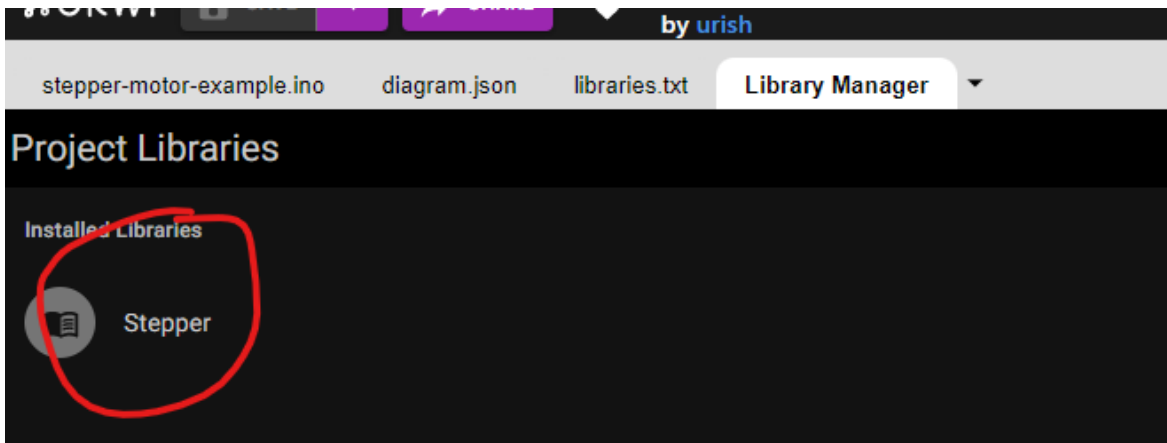
Perhatikan tampilan yang anda peroleh, tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit8-tugas01.

Mencoba project dengan library

1. Silahkan buat project baru dengan mikrokontroller dan skema seperti yang di atas.
2. Masuk pada bagian library manager, lalu add library dan cari library "stepper"



Maka akan ada library seperti pada gambar dibawah ini



3. Lalu tuliskan listing seperti pada bawah ini

```

1 // Stepper motor on Wokwi!
2
3 #include <Stepper.h>
4
5 const int stepsPerRevolution = 200; // change this to fit the number of steps per revolution
6 // for your motor
7
8 // initialize the stepper library on pins 8 through 11:
9 Stepper myStepper(stepsPerRevolution, 8, 9, 10, 11);
10
11 void setup() {
12     // set the speed at 60 rpm:
13     myStepper.setSpeed(60);
14     // initialize the serial port:
15     Serial.begin(9600);
16 }
17
18 void loop() {
19     // step one revolution in one direction:
20     Serial.println("clockwise");
21     myStepper.step(stepsPerRevolution);
22     delay(500);
23
24     // step one revolution in the other direction:
25     Serial.println("counterclockwise");
26     myStepper.step(-stepsPerRevolution);
27     delay(500);
28 }

```

4. Silahkan uji dengan menjalankan simulasi.

Jelaskan apa pengaruh perubahan variabel get speed,, step positif dan negatif, dan variabel stepsPerRevolution.

Aturan Penilaian (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-08	CPMK-05	Selesaikan praktikum tanpa library	Hasil praktikum langkah	100

8.7. POST TEST

Jawablah pertanyaan berikut (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-05	CPMK-03	Selesaikan praktikum dengan menggunakan library	60
2.	CPL-05	CPMK-03	Jelaskan apa pengaruh perubahan variabel get speed, step positif dan negatif, dan variabel stepsPerRevolution.	40

8.8. HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-03	CPMK-01	10%		
2.	Praktik	CPL-03	CPMK-01	50%		
3.	Post-Test	CPL-03	CPMK-01	40%		
Total Nilai						

PRAKTIKUM 9: PENGENALAN SIMULASI MPU6050

Pertemuan ke : 9

Total Alokasi Waktu : 90 menit

- Materi : 15 menit
- Pre-Test : 15 menit
- Praktikum : 45 menit
- Post-Test : 15 menit

Total Bobot Penilaian : 100%

- Pre-Test : 10 %
- Praktik : 50 %
- Post-Test : 40 %

Pemenuhan CPL dan CPMK:

CPL-05	Mampu mengkaji / menganalisis implikasi pengembangan atau implementasi ilmu pengetahuan teknologi, menyusun deskripsi saintifik hasil kajian untuk pemecahan masalah dengan mempertimbangkan multidisiplin ilmu
CPMK-04	Sensor Gyroscope dan pemrogramannya Sensor Accelerometer dan pemrogramannya Heading sensor (compass sensor) dan pemrogramannya

9.1. DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

1. Menjelaskan apa itu pemrograman mikrokontroler.
2. Menerapkan pemrograman mikrokontroler.

9.2. INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-05	CPMK-04	Mampu memprogram Sensor Gyroscope, Sensor Accelerometer, dan Heading sensor (compass sensor).
--------	---------	---

9.3. TEORI PENDUKUNG

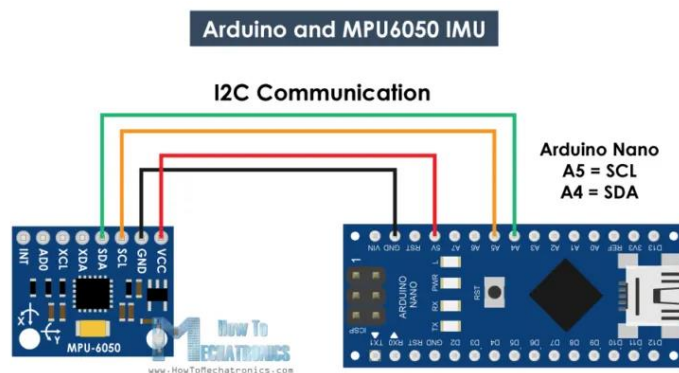
IMU MPU6050 memiliki akselerometer 3-Axis dan giroskop 3-Axis yang terintegrasi dalam satu chip.

Giroskop mengukur kecepatan rotasi atau laju perubahan posisi sudut dari waktu ke waktu, sepanjang sumbu X, Y, dan Z. Ini menggunakan teknologi MEMS dan Efek Coriolis untuk mengukur, tetapi untuk detail lebih lanjut tentang itu Anda dapat memeriksa tutorial khusus Cara Kerja Sensor MEMS saya. Keluaran giroskop adalah dalam derajat per detik, jadi untuk mendapatkan posisi sudut kita hanya perlu mengintegrasikan kecepatan sudut.

Sebaliknya, akselerometer MPU6050 mengukur akselerasi dengan cara yang sama seperti yang dijelaskan pada video sebelumnya untuk sensor akselerometer ADXL345. Secara singkat, itu dapat mengukur percepatan gravitasi sepanjang 3 sumbu dan menggunakan beberapa matematika trigonometri kita dapat menghitung sudut di mana sensor diposisikan. Jadi, jika kita memadukan, atau menggabungkan data akselerometer dan giroskop, kita bisa mendapatkan informasi yang sangat akurat tentang orientasi sensor.

IMU MPU6050 juga disebut perangkat pelacak gerak enam sumbu atau perangkat 6 DoF (enam Derajat Kebebasan), karena 6 keluarannya, atau 3 keluaran akselerometer dan 3 keluaran giroskop.

Kami menggunakan protokol I2C untuk komunikasi dengan Arduino sehingga kami hanya memerlukan dua kabel untuk menghubungkannya, ditambah dua kabel untuk menyalaknya.



Sumber <https://howtomechatronics.com/tutorials/arduino/arduino-and-mpu6050-accelerometer-and-gyroscope-tutorial/>

9.4. HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

1. Laptop
2. Browser Online
3. <https://wokwi.com/>

9.5. PRE-TEST

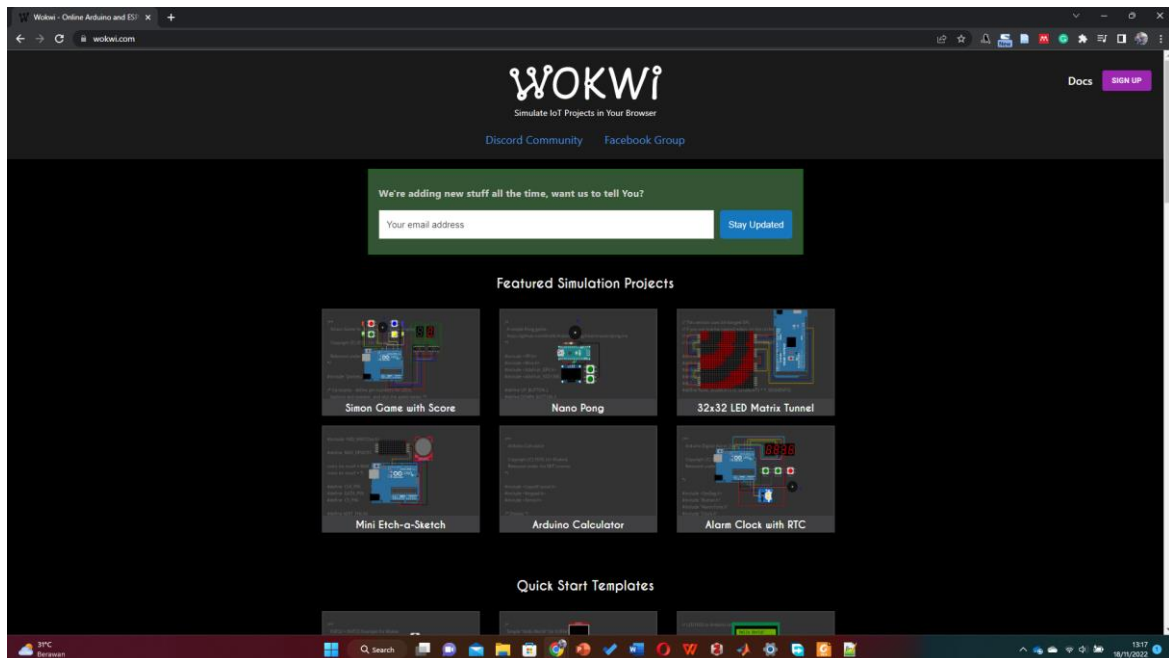
Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-05	CPMK-04	Bagaimana cara kerja Sensor Gyroscope dan Sensor Accelerometer?	100

9.6. LANGKAH PRAKTIKUM

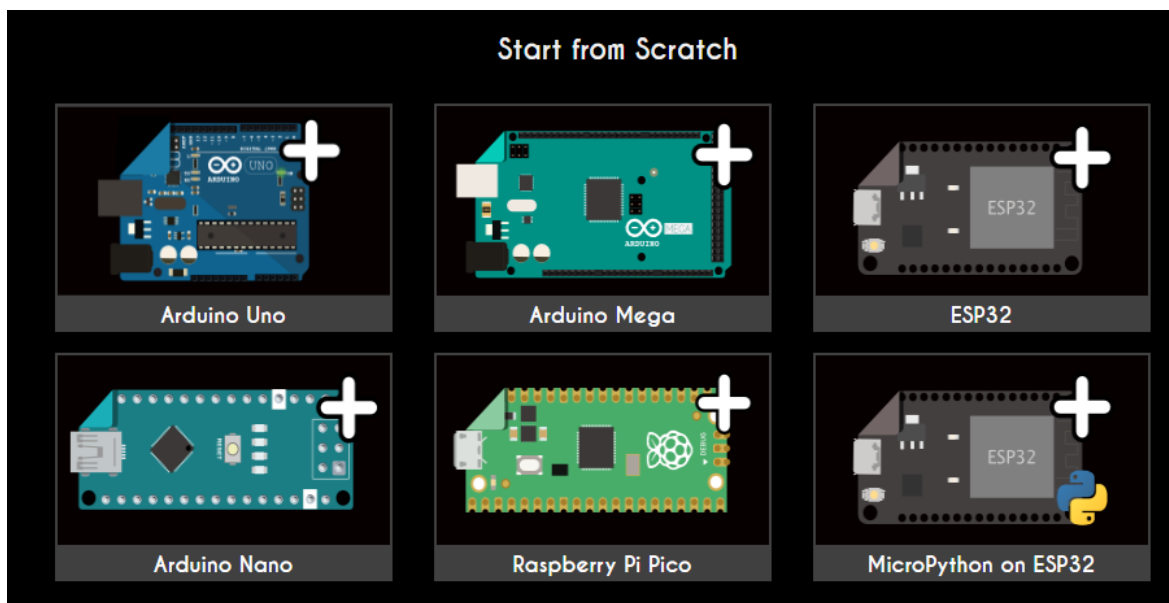
1. Buka Browser

Silahkan buka alamat <https://wokwi.com/>, alamat ini digunakan untuk simulasi menggunakan mikrokontroler.

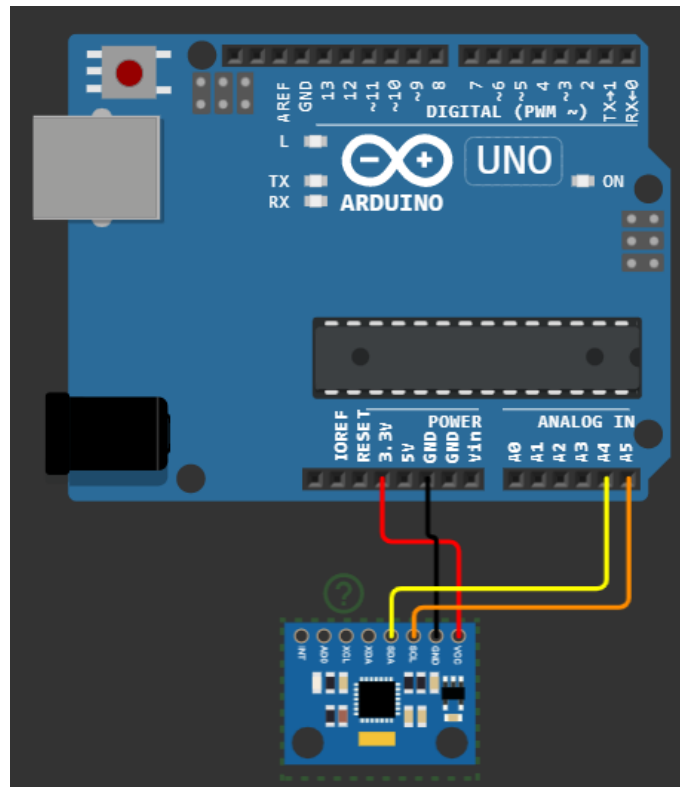


2. Silahkan sign in menggunakan akun yang ada.

3. Silahkan pilih start from scratch dengan mikrokontroler arduino



4. Silahkan buat diagram seperti dibawah ini dengan komponen MPU6050



Perhatikan pemasangan kabel yang ada dalam skema.

5. Silahkan masukkan library "Adafruit MPU6050"

6. Silahkan tuliskan program sesuai dengan listing dibawah

```
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
#include <Wire.h>

Adafruit_MPU6050 mpu;

void setup(void) {
  Serial.begin(115200);

  // Try to initialize!
  if (!mpu.begin()) {
    Serial.println("Failed to find MPU6050 chip");
    while (1) {
      delay(10);
    }
  }

  mpu.setAccelerometerRange(MPU6050_RANGE_16_G);
  mpu.setGyroRange(MPU6050_RANGE_250_DEG);
  mpu.setFilterBandwidth(MPU6050_BAND_21_HZ);
  Serial.println("");
  delay(100);
}

void loop() {

  /* Get new sensor events with the readings */
  sensors_event_t a, g, temp;
  mpu.getEvent(&a, &g, &temp);

  /* Print out the values */
```

```

Serial.print(a.acceleration.x);
Serial.print(",");
Serial.print(a.acceleration.y);
Serial.print(",");
Serial.print(a.acceleration.z);
Serial.print(", ");
Serial.print(g.gyro.x);
Serial.print(",");
Serial.print(g.gyro.y);
Serial.print(",");
Serial.print(g.gyro.z);
Serial.println("");

delay(10);
}

```

7. Jalankan project anda.

Perhatikan tampilan yang anda peroleh, tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit9-tugas01.

8. Mencoba listing kedua

9. Silahkan tuliskan program sesuai dengan listing dibawah

```

// Basic demo for accelerometer readings from Adafruit MPU6050

#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
#include <Wire.h>

Adafruit_MPU6050 mpu;

void setup(void) {
  Serial.begin(115200);
  while (!Serial)
    delay(10); // will pause Zero, Leonardo, etc until serial console opens

  Serial.println("Adafruit MPU6050 test!");

  // Try to initialize!
  if (!mpu.begin()) {
    Serial.println("Failed to find MPU6050 chip");
    while (1) {
      delay(10);
    }
  }
  Serial.println("MPU6050 Found!");

  mpu.setAccelerometerRange(MPU6050_RANGE_8_G);
  Serial.print("Accelerometer range set to: ");
  switch (mpu.getAccelerometerRange()) {
    case MPU6050_RANGE_2_G:
      Serial.println("+2G");
      break;
    case MPU6050_RANGE_4_G:
      Serial.println("+4G");
      break;
    case MPU6050_RANGE_8_G:
      Serial.println("+8G");
      break;
    case MPU6050_RANGE_16_G:
      Serial.println("+16G");
      break;
  }

  mpu.setGyroRange(MPU6050_RANGE_2000_DEG);
  Serial.print("Gyro range set to: ");

```



```

switch (mpu.getGyroRange()) {
case MPU6050_RANGE_250_DEG:
    Serial.println("+ 250 deg/s");
    break;
case MPU6050_RANGE_500_DEG:
    Serial.println("+ 500 deg/s");
    break;
case MPU6050_RANGE_1000_DEG:
    Serial.println("+ 1000 deg/s");
    break;
case MPU6050_RANGE_2000_DEG:
    Serial.println("+ 2000 deg/s");
    break;
}

mpu.setFilterBandwidth(MPU6050_BAND_21_HZ);
Serial.print("Filter bandwidth set to: ");
switch (mpu.getFilterBandwidth()) {
case MPU6050_BAND_260_HZ:
    Serial.println("260 Hz");
    break;
case MPU6050_BAND_184_HZ:
    Serial.println("184 Hz");
    break;
case MPU6050_BAND_94_HZ:
    Serial.println("94 Hz");
    break;
case MPU6050_BAND_44_HZ:
    Serial.println("44 Hz");
    break;
case MPU6050_BAND_21_HZ:
    Serial.println("21 Hz");
    break;
case MPU6050_BAND_10_HZ:
    Serial.println("10 Hz");
    break;
case MPU6050_BAND_5_HZ:
    Serial.println("5 Hz");
    break;
}

Serial.println("");
delay(100);
}

void loop() {

    /* Get new sensor events with the readings */
    sensors_event_t a, g, temp;
    mpu.getEvent(&a, &g, &temp);

    /* Print out the values */
    Serial.print("Acceleration X: ");
    Serial.print(a.acceleration.x);
    Serial.print(", Y: ");
    Serial.print(a.acceleration.y);
    Serial.print(", Z: ");
    Serial.print(a.acceleration.z);
    Serial.println(" m/s^2");

    Serial.print("Rotation X: ");
    Serial.print(g.gyro.x);
    Serial.print(", Y: ");
    Serial.print(g.gyro.y);
    Serial.print(", Z: ");
    Serial.print(g.gyro.z);
    Serial.println(" rad/s");

    Serial.print("Temperature: ");
    Serial.print(temp.temperature);

```

```

Serial.println(" degC");

Serial.println("");
delay(500);
}

```

10. Jalankan project anda.

Perhatikan tampilan yang anda peroleh, tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit9-tugas02.

Aturan Penilaian (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-05	CPMK-04	Selesaikan langkah 1-7	Hasil praktikum langkah	100

9.7. POST TEST

Jawablah pertanyaan berikut (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-05	CPMK-04	Selesaikan langkah 7-10	100

9.8. HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-05	CPMK-04	10%		
2.	Praktik	CPL-05	CPMK-04	50%		
3.	Post-Test	CPL-05	CPMK-04	40%		
Total Nilai						



LEMBAR JAWABAN PRE-TEST DAN POST-TEST PRAKTIKUM

Nama : NIM :	Asisten: Paraf Asisten:	Tanggal: Nilai:
-------------------------------	--	----------------------------------

--

PRAKTIKUM 10: PENGENALAN PENGGERAK SERVO

Pertemuan ke : 10

Total Alokasi Waktu : 90 menit

- Materi : 15 menit
- Pre-Test : 15 menit
- Praktikum : 45 menit
- Post-Test : 15 menit

Total Bobot Penilaian : 100%

- Pre-Test : 10 %
- Praktik : 50 %
- Post-Test : 40 %

Pemenuhan CPL dan CPMK:

CPL-05	Mampu mengkaji / menganalisis implikasi pengembangan atau implementasi ilmu pengetahuan teknologi, menyusun deskripsi saintifik hasil kajian untuk pemecahan masalah dengan mempertimbangkan multidisiplin ilmu
CPMK-03	Motor DC, metode pengendalian dan pemrogramannya

10.1. DESKRIPSI CAPAIAN PEMBELAJARAN

Setelah mengikuti praktikum ini mahasiswa diharapkan mampu:

1. Menjelaskan apa itu pemrograman mikrokontroler.
2. Menerapkan pemrograman mikrokontroler.

10.2. INDIKATOR KETERCAPAIAN PEMBELAJARAN

Indikator ketercapaian diukur dengan:

CPL-05	CPMK-03	Mampu membangun dan mempresentasikan sistem Motor DC, metode pengendalian dan pemrogramannya
--------	---------	--

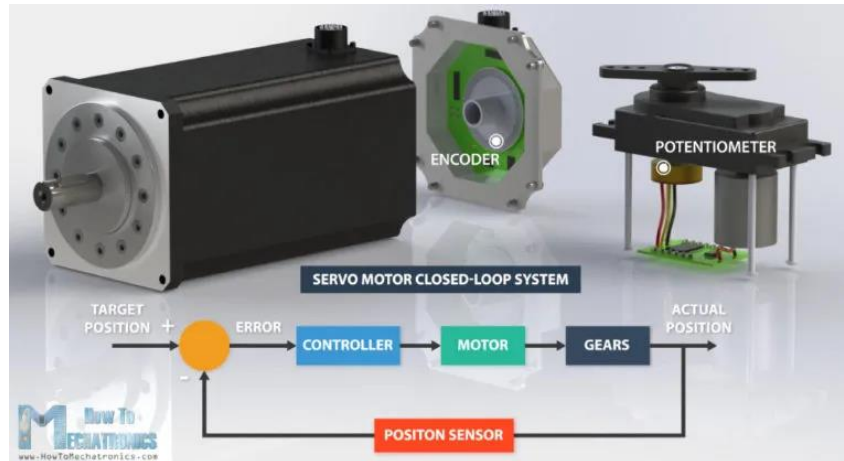
10.3. TEORI PENDUKUNG

Servo adalah pilihan tepat untuk proyek robotika, otomasi, model RC, dan sebagainya. Saya telah menggunakannya di banyak proyek Arduino.

Apa itu Motor Servo?

Motor servo adalah sistem loop tertutup yang menggunakan umpan balik posisi untuk mengontrol gerakan dan posisi akhirnya. Ada banyak jenis motor servo dan fitur utamanya adalah kemampuan untuk mengontrol posisi porosnya secara tepat.

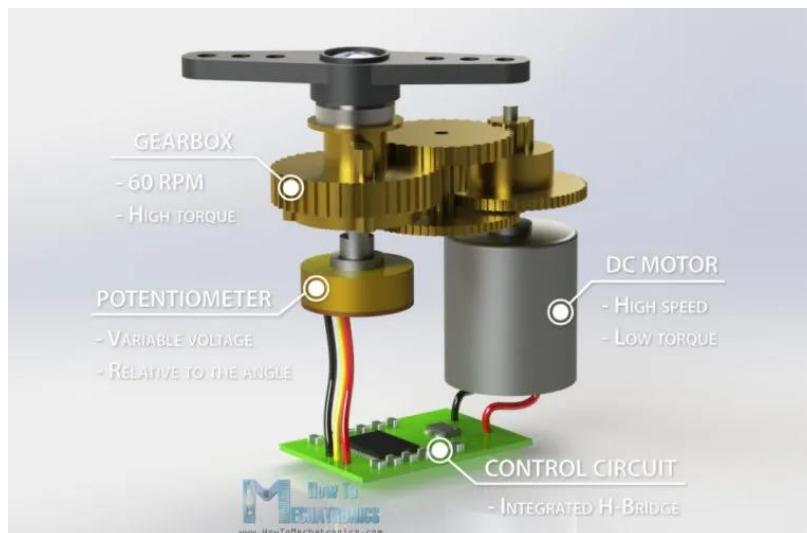
Pada motor servo tipe industri, sensor umpan balik posisi biasanya berupa enkoder presisi tinggi, sedangkan pada servo RC atau hobi yang lebih kecil, sensor posisi biasanya berupa potensiometer sederhana. Posisi sebenarnya yang ditangkap oleh perangkat ini diumpankan kembali ke detektor kesalahan yang dibandingkan dengan posisi target. Kemudian sesuai dengan kesalahan pengontrol mengoreksi posisi motor yang sebenarnya agar sesuai dengan posisi target.



Sumber <https://howtomechanics.com/how-it-works/how-servo-motors-work-how-to-control-servos-using-arduino/>

Bagaimana Motor Servo Bekerja?

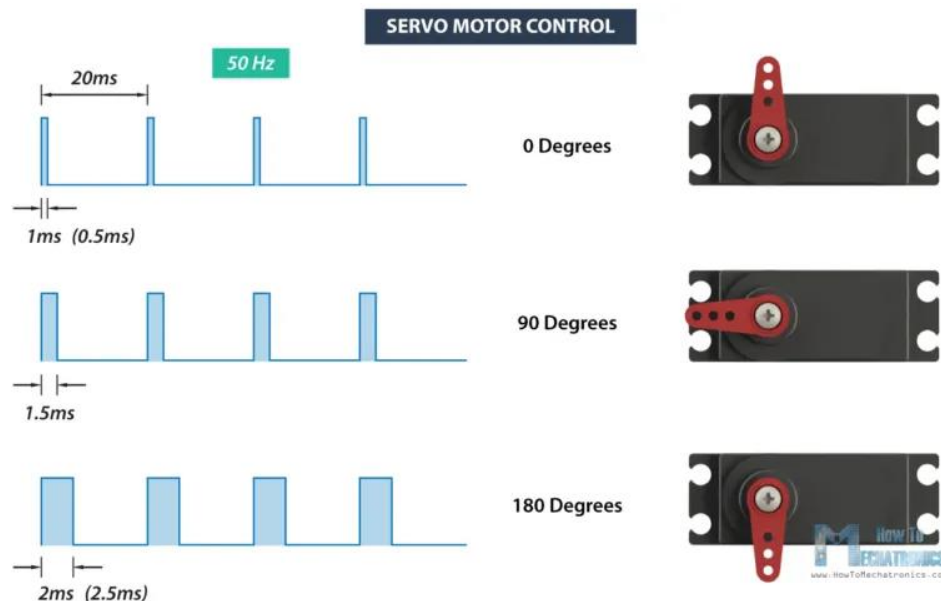
Ada empat komponen utama di dalam hobby servo, motor DC, gearbox, potensiometer dan rangkaian kontrol. Motor DC berkecepatan tinggi dan torsi rendah tetapi gearbox mengurangi kecepatan menjadi sekitar 60 RPM dan pada saat yang sama meningkatkan torsi.



Sumber <https://howtomechanics.com/how-it-works/how-servo-motors-work-how-to-control-servos-using-arduino/>

Potensiometer terpasang pada gigi akhir atau poros keluaran, sehingga saat motor berputar, potensiometer juga berputar, sehingga menghasilkan tegangan yang terkait dengan sudut absolut poros keluaran. Pada rangkaian kontrol, tegangan potensiometer ini dibandingkan dengan tegangan yang berasal dari jalur sinyal. Jika diperlukan, pengontrol mengaktifkan H-Bridge terintegrasi yang memungkinkan motor berputar ke salah satu arah hingga kedua sinyal mencapai selisih nol.

Motor servo dikendalikan dengan mengirimkan serangkaian pulsa melalui jalur sinyal. Frekuensi sinyal kontrol harus 50Hz atau pulsa harus muncul setiap 20ms. Lebar pulsa menentukan posisi sudut servo dan jenis servo ini biasanya dapat berputar 180 derajat (mereka memiliki batas perjalanan fisik).



Sumber <https://howtomechatronics.com/how-it-works/how-servo-motors-work-how-to-control-servos-using-arduino/>

Umumnya pulsa dengan durasi 1ms sesuai dengan posisi 0 derajat, durasi 1,5ms hingga 90 derajat dan 2ms hingga 180 derajat. Padahal durasi minimum dan maksimum pulsa terkadang bisa berbeda-beda dengan merek yang berbeda dan bisa 0,5 ms untuk posisi 0 derajat dan 2,5 ms untuk posisi 180 derajat.

10.4. HARDWARE DAN SOFTWARE

Hardware dan software yang digunakan dalam praktikum ini yaitu:

1. Laptop
2. Browser Online
3. <https://wokwi.com/>

10.5. PRE-TEST

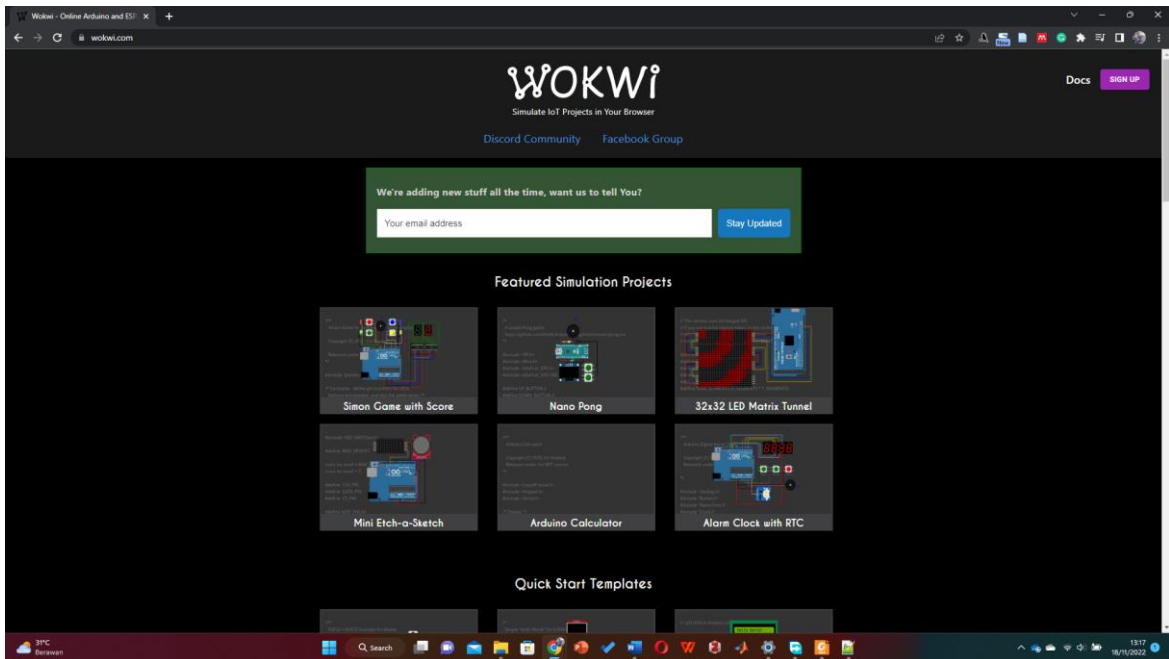
Jawablah pertanyaan berikut (**Total Skor: 100**):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-05	CPMK-03	Apa itu motor servo?	30
2.	CPL-05	CPMK-03	Bagaimana cara kerja motor servo?	70

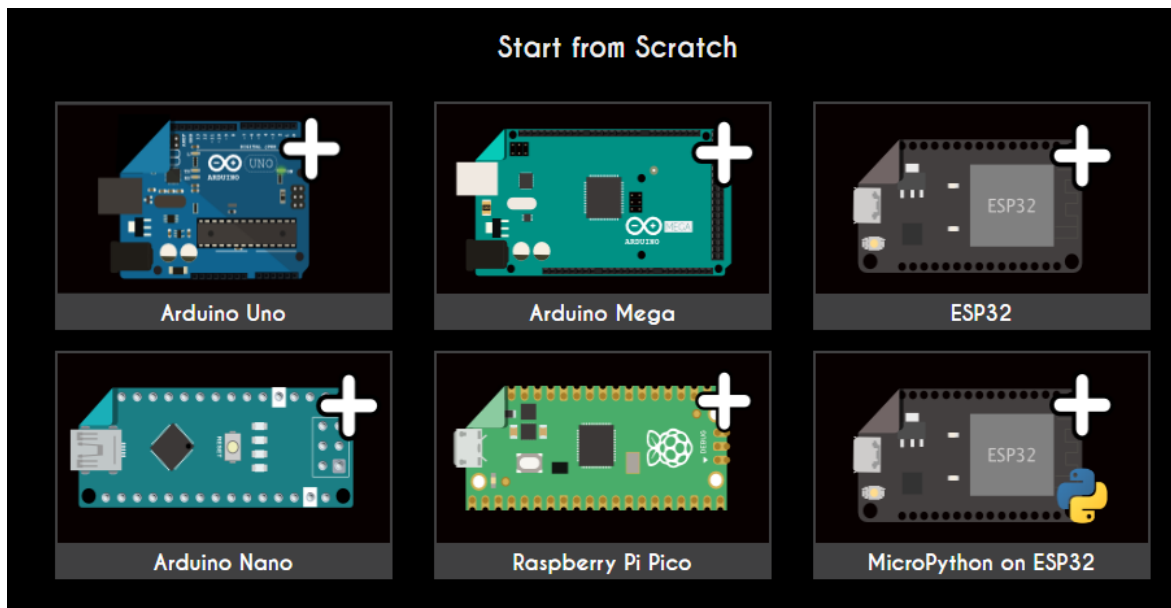
10.6. LANGKAH PRAKTIKUM

1. Buka Browser

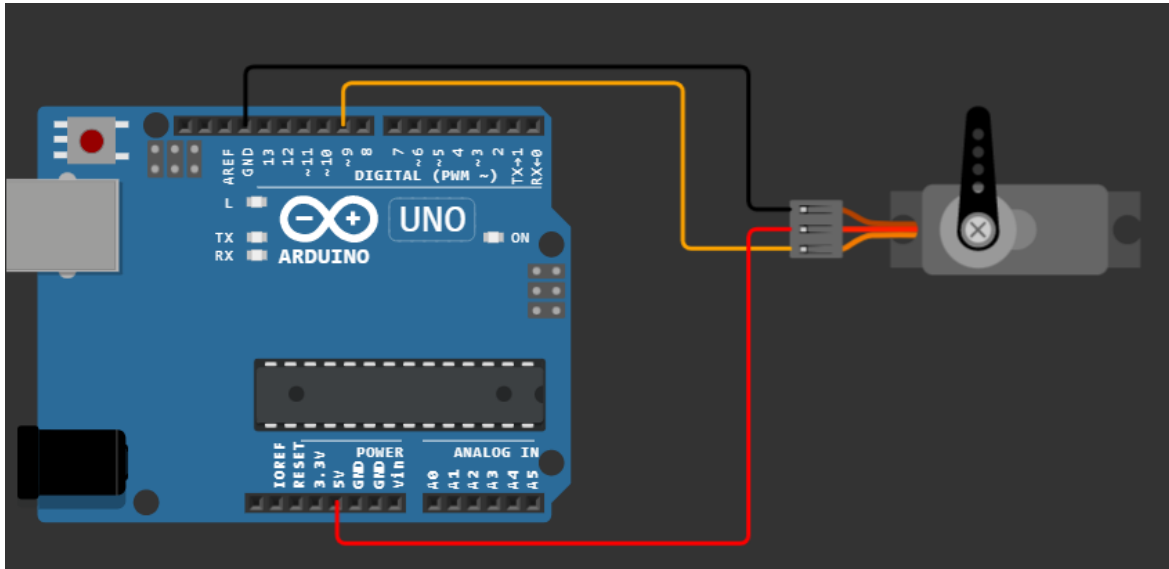
Silahkan buka alamat <https://wokwi.com/>, alamat ini digunakan untuk simulasi menggunakan mikrokontroler.



2. Silahkan sign in menggunakan akun yang ada.
3. Silahkan pilih start from scratch dengan mikrokontroler arduino



4. Silahkan buat diagram seperti dibawah ini dengan komponen SERVO



Perhatikan pemasangan kabel yang ada dalam skema.

5. Silahkan tuliskan program sesuai dengan listing dibawah

```
#include <Servo.h>

Servo myservo; // create servo object to control a servo
// twelve servo objects can be created on most boards

int pos = 0; // variable to store the servo position

void setup() {
  myservo.attach(9); // attaches the servo on pin 9 to the servo object
}

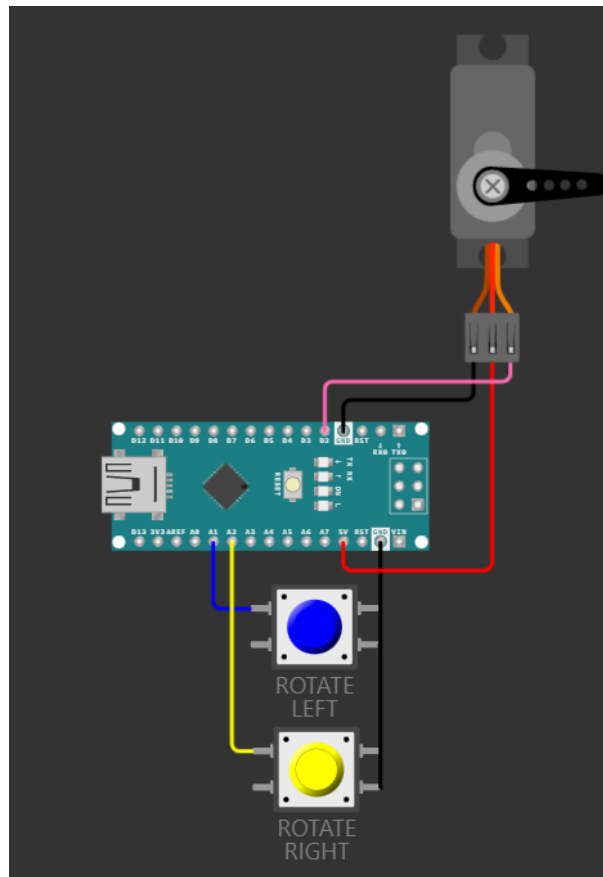
void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180
degrees
    // in steps of 1 degree
    myservo.write(pos); // tell servo to go to position in
variable 'pos'
    delay(15); // waits 15ms for the servo to reach
the position
  }
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0
degrees
    myservo.write(pos); // tell servo to go to position in
variable 'pos'
    delay(15); // waits 15ms for the servo to reach
the position
  }
}
```

6. Jalankan project anda.

Perhatikan tampilan yang anda peroleh, tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit10-tugas01.

Skema Kedua

1. Silahkan buat diagram seperti dibawah ini dengan komponen SERVO



Perhatikan pemasangan kabel yang ada dalam skema.

2. Silahkan gunakan library "servo"
3. Silahkan tuliskan program sesuai dengan listing dibawah

```
#include <Servo.h>

Servo arm; // Create a "Servo" object called "arm"
float pos = 0.0; // Variable where the arm's position will be stored (in degrees)
float step = 1.0; // Variable used for the arm's position step

void setup()
{
  pinMode(A1, INPUT_PULLUP); // Set the A1 pin to a pushbutton in pullup mode
  pinMode(A2, INPUT_PULLUP); // Set the A1 pin to a pushbutton in pullup mode

  arm.attach(2); // Attache the arm to the pin 2
  arm.write(pos); // Initialize the arm's position to 0 (leftmost)
}

void loop()
{
  if (!digitalRead(A1)) // Check for the yellow button input
  {
    if (pos>0) // Check that the position won't go lower than 0°
    {
      arm.write(pos); // Set the arm's position to "pos" value
      pos-=step; // Decrement "pos" of "step" value
    }
  }
}
```

```

        delay(5); // Wait 5ms for the arm to reach the position
    }
}

if (!digitalRead(A2)) // Check for the blue button input
{
    if (pos<180) // Check that the position won't go higher than 180°
    {
        arm.write(pos); // Set the arm's position to "pos" value
        pos+=step; // Increment "pos" of "step" value
        delay(5); // Wait 5ms for the arm to reach the position
    }
}
}
}

```

4. Jalankan project anda.

Perhatikan tampilan yang anda peroleh, tekan tombol PrintScreen pada Laptop anda dan simpan dengan nama NIM-unit10-tugas02.

Aturan Penilaian (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Dokumen Pendukung	Skor
1.	CPL-05	CPMK-03	Selesaikan langkah 1-3	Hasil praktikum langkah	100

10.7. POST TEST

Jawablah pertanyaan berikut (Total Skor: 100):

No	CPL	CPMK	Pertanyaan	Skor
1.	CPL-05	CPMK-03	Selesaikan langkah 4	60
2.	CPL-05	CPMK-03	Selesaikan langkah 5	40

10.8. HASIL CAPAIAN PRAKTIKUM

Diisi oleh asisten setelah semua assessment dinilai.

No	Bentuk Assessment	CPL	CPMK	Bobot	Skor (0-100)	Nilai Akhir (Bobot x Skor)
1.	Pre-Test	CPL-03	CPMK-01	10%		
2.	Praktik	CPL-03	CPMK-01	50%		
3.	Post-Test	CPL-03	CPMK-01	40%		
Total Nilai						



DAFTAR PUSTAKA

1.

