
TUTORIAL INSTALASI *MINDWAVE* *NEUROSKY, PYTHON* DAN PENJELASAN PADA *CODING PYTHON*

TUTORIAL INSTALASI *MINDWAVE NEUROSKY*, *PYTHON* DAN PENJELASAN PADA *CODING PYTHON*

Asep Setiawan

1164005

-
1. *Install Mindwave Neurosky*
 2. *Install Python 2.7.11*
 3. *Penjelasan dan Codingan Python*
 4. *Laporan Internship I*

KATA PENGATAR

Terimakasih politeknik pos indonesia sebagai institusi yang membuat saya berkembang, kemudian para dosen yang mengajarkan matakuliah dan teman-teman yang selalu support sampai sekarang.

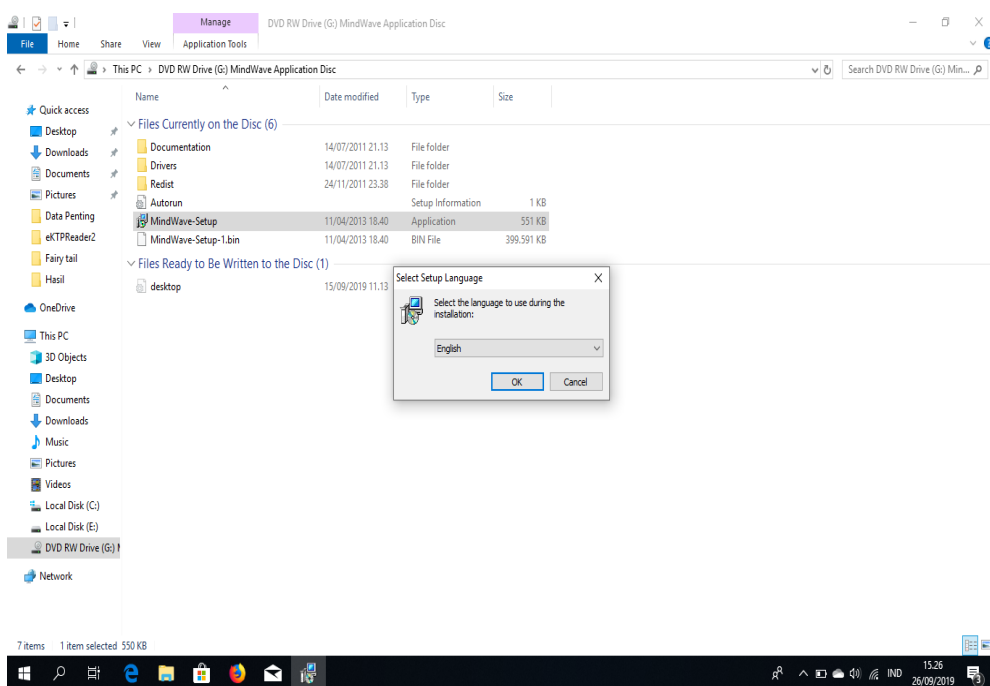
DAFTAR ISI

DAFTAR GAMBAR

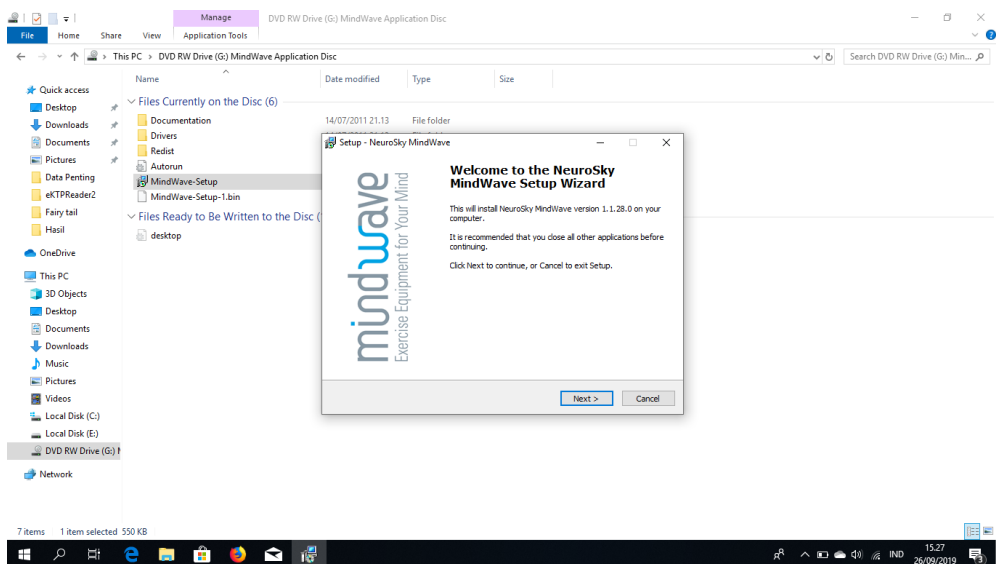
CHAPTER I

INSTALL MINDWAVE NEUROSKY

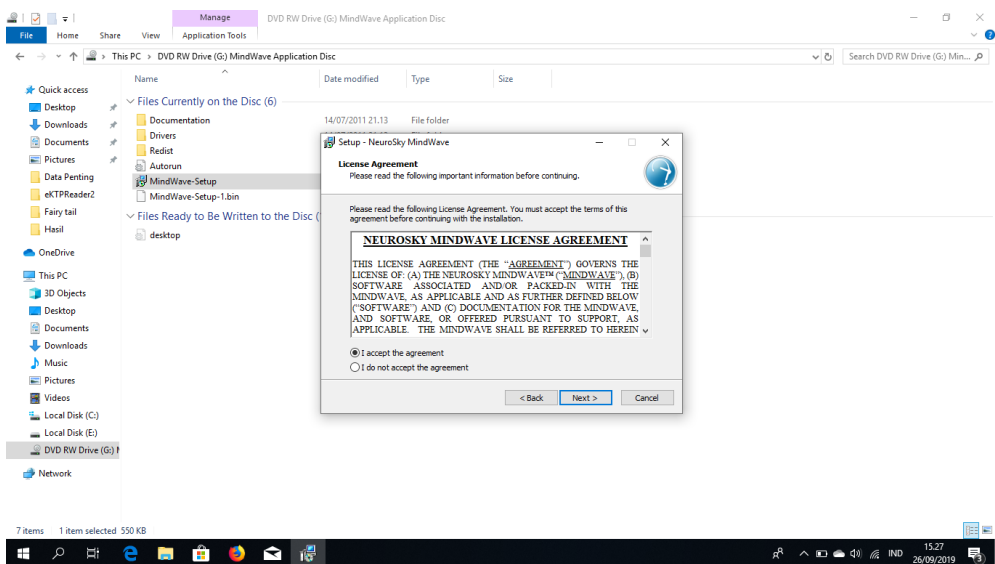
Pertama Install aplikasi mindwave neurosky, pilih ok dalam bahasa inggris.



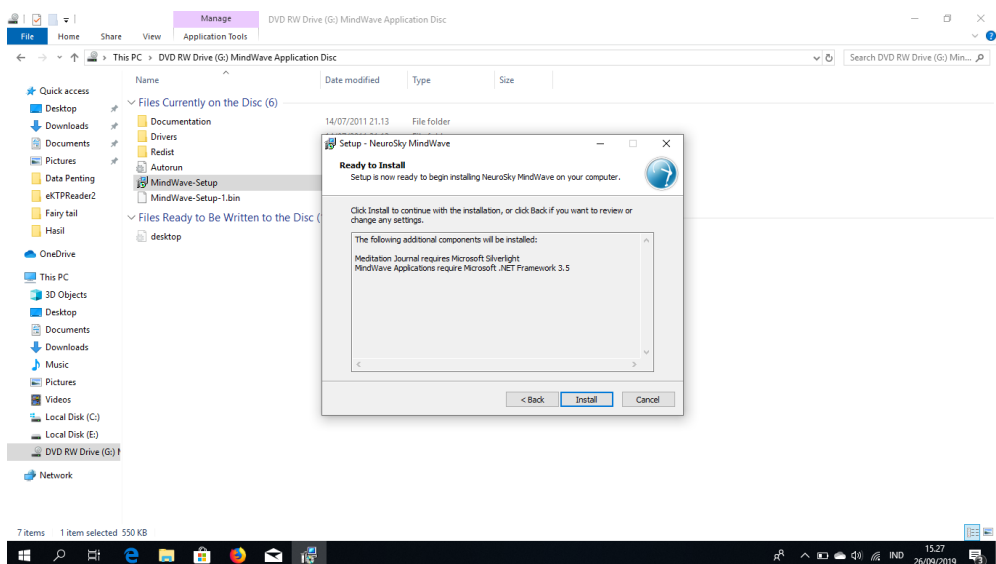
Kedua, pilih next untuk melanjutkan tahap kedua ini.



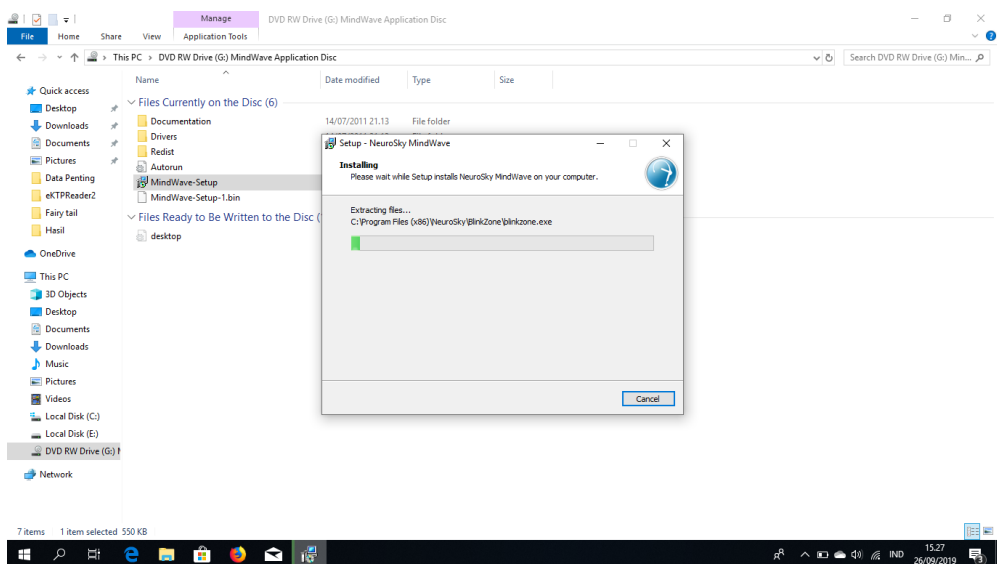
Ketiga, pilih “I accept the agreement” lalu klik Next.



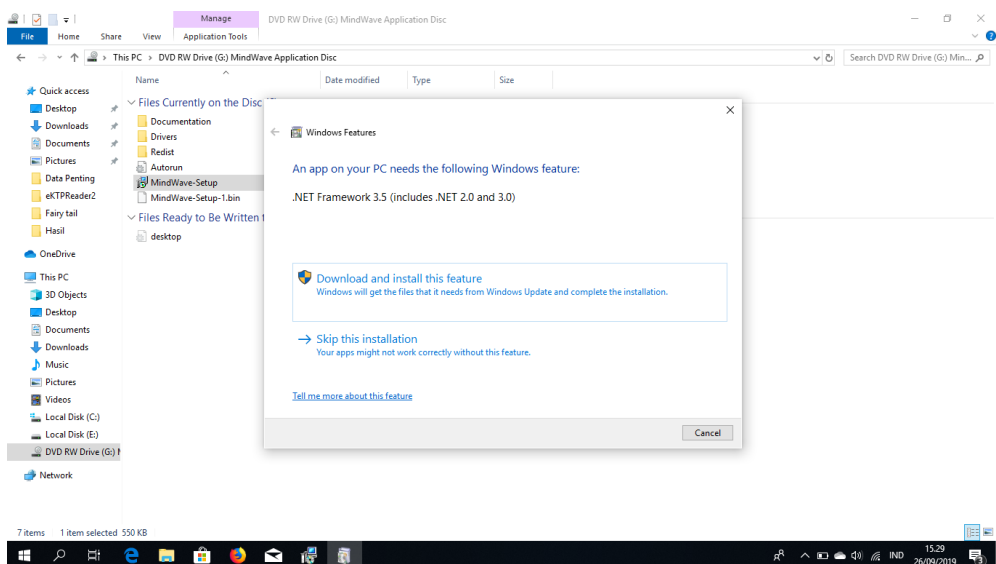
Keempat, ada tulisan yang mengarah untuk klik “Next”.



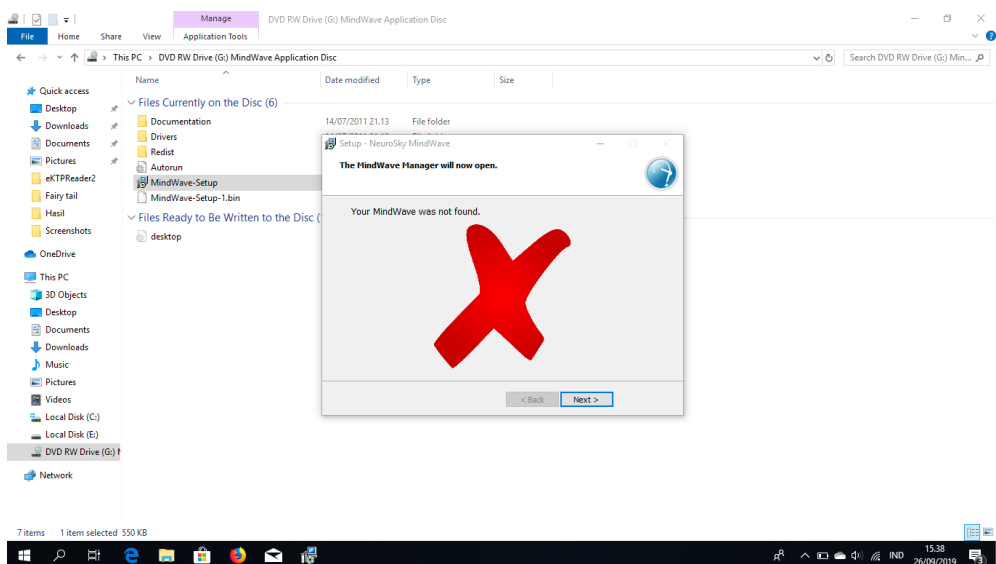
Kelima, tunggu sampai loding selesai.



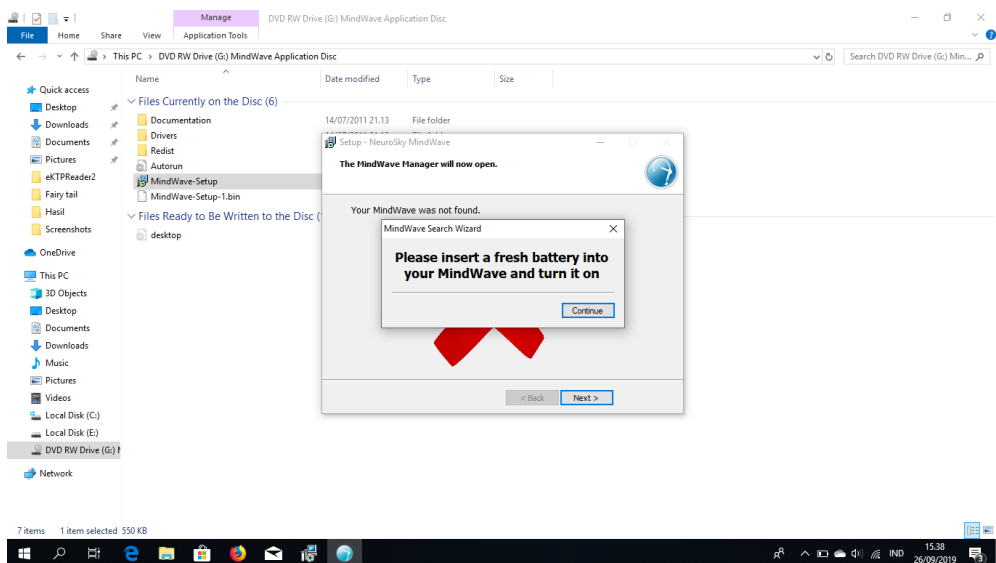
Keenam, saat pemilihan update NET. Framework 3.5 , kemudian pilih “download and install this feature”.



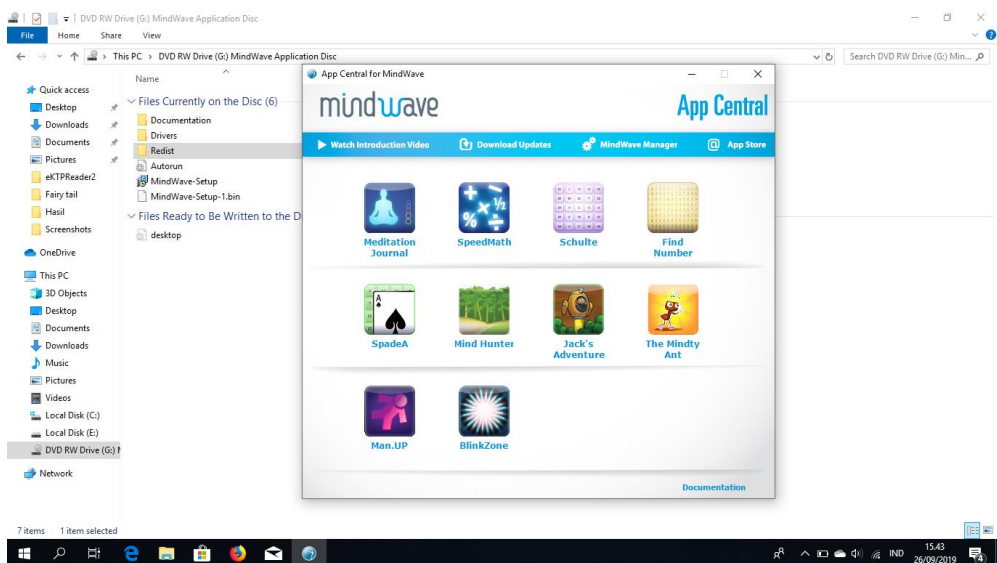
Ketujuh, jika selesai maka akan tampil yang dibawah ini, tanda (x) itu karena belum di pasang usb dari mindwavenya. Jika sudah terhubung maka akan ada tanda (✓) berarti sudah terhubung, lalu klik next.



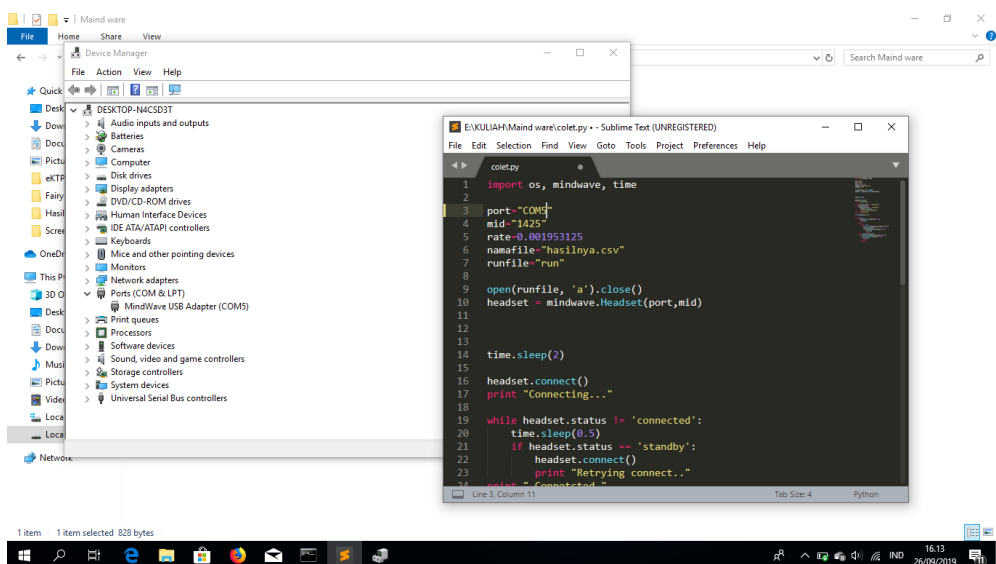
Kedelapan, muncul popup seperti dibawah kemudian klik “continue”.



Kesembilan, jika sudah proses penginstallan maka akan tampil seperti dibawah ini



Kesepuluh, sesuaikan portnya contoh dibawah ini port (COM5).



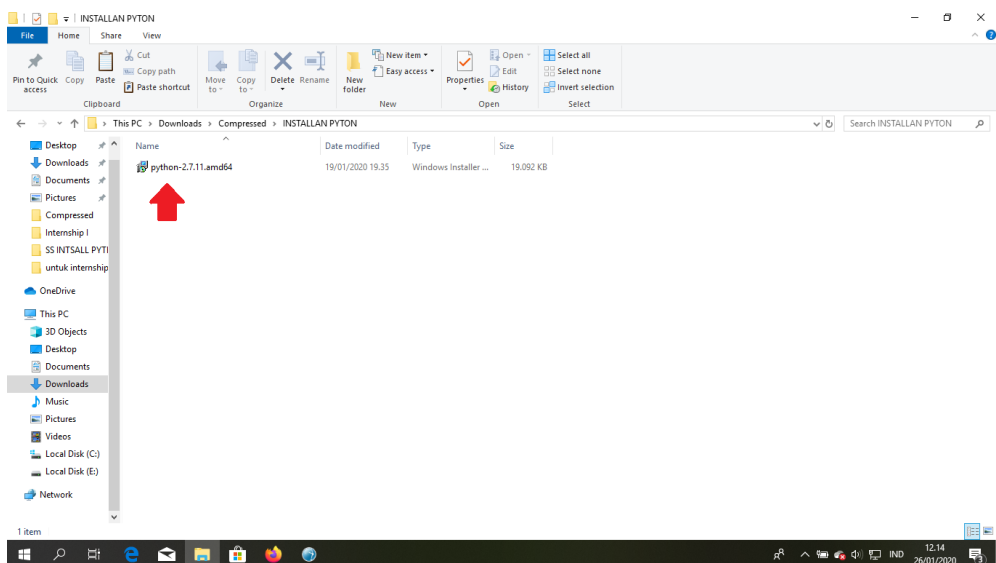
Selesai, penginstalan *mindwave neurosky*.

CHAPTER II

INSTALL PYTHON 2.7.11

Pertama download terlebih dahulu python 2.7.11 di link ini
<https://www.python.org/downloads/release/python-2711/>

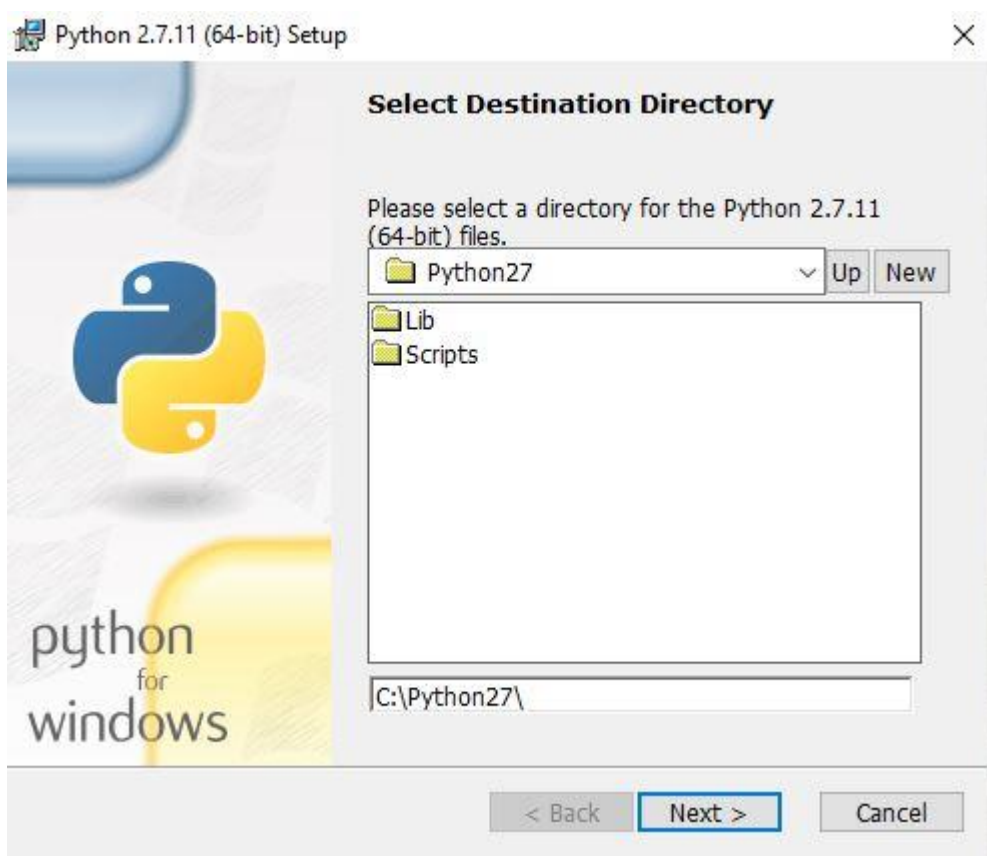
Kedua, Jika sudah didownload kemudian double klik pada python.



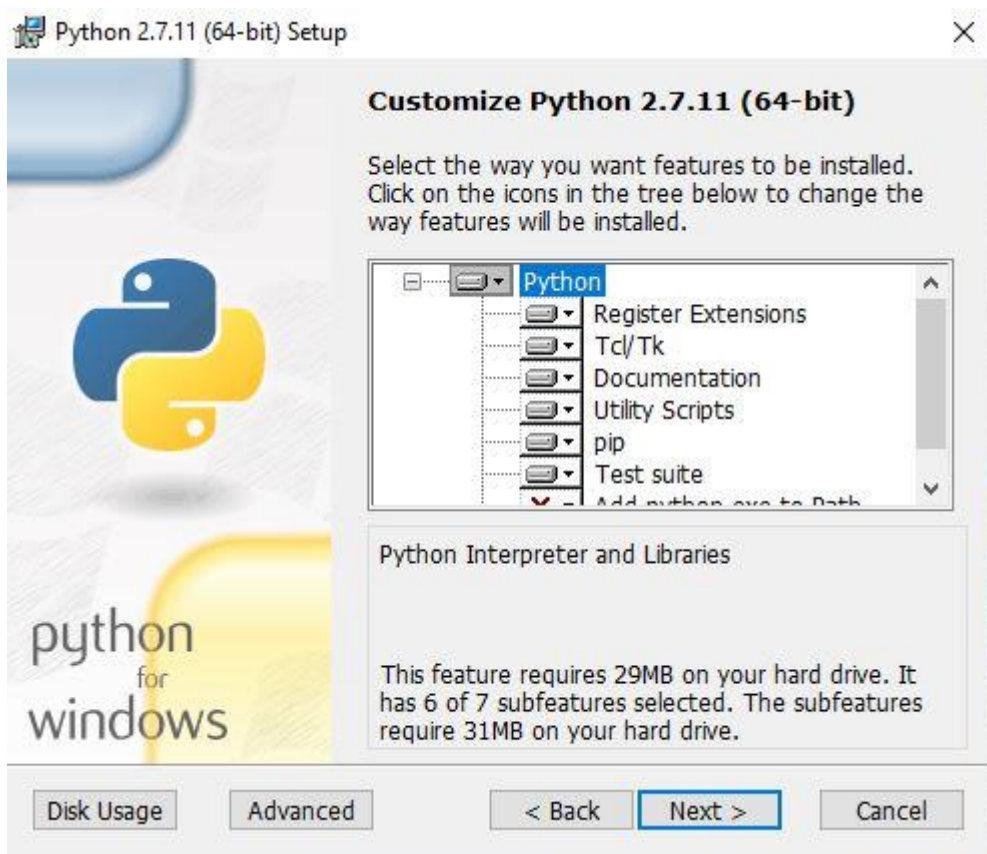
Ketiga, muncul seperti tampilan di bawah kemudian klik “Next”.



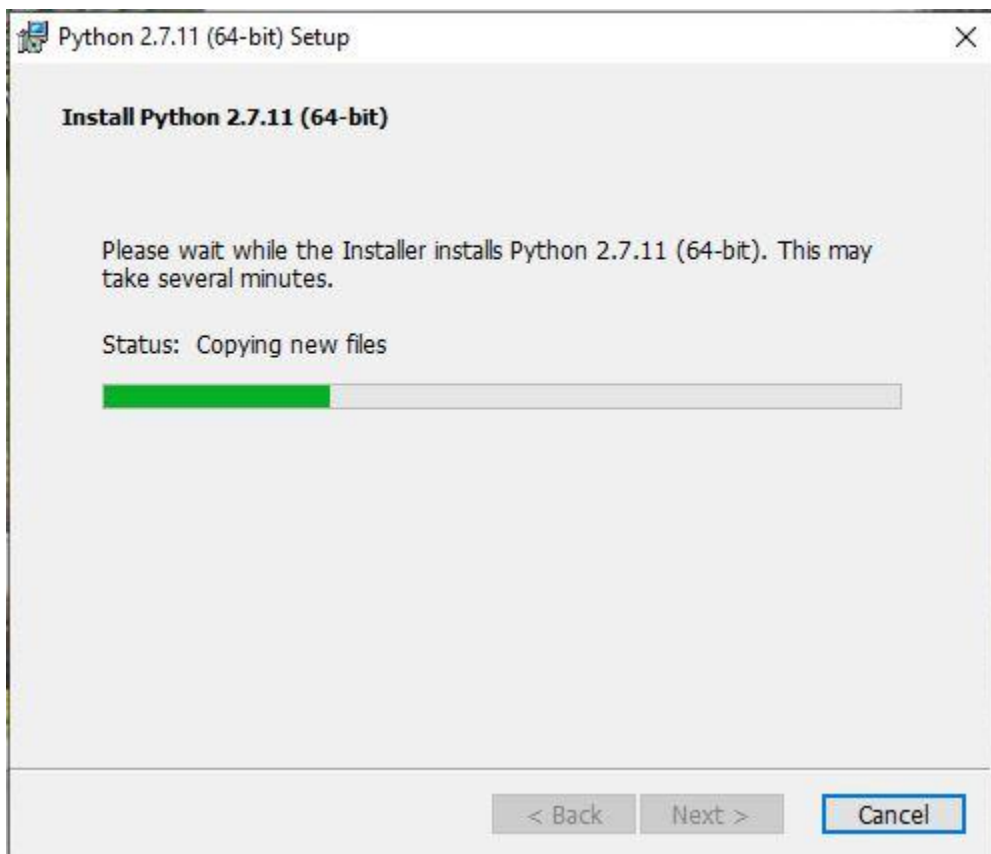
Keempat, pilih direktori sesuai yang diinginkan seperti dibawah ini, lalu klik “Next”.



Kelima, cek pilihan yang diinstall jika sudah sesuai maka klik “Next”.



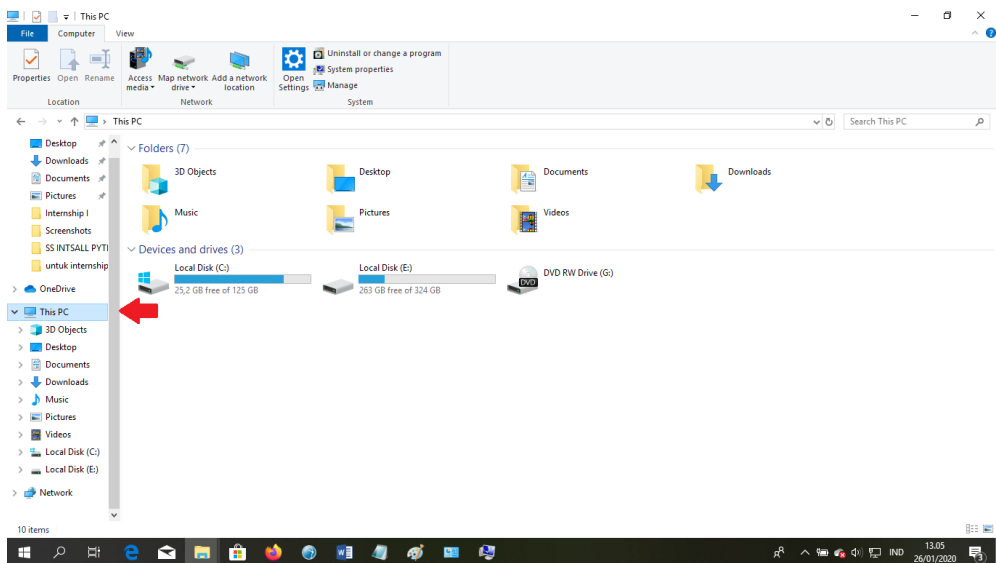
Keenam, tunggu sampai loding ini selesai.



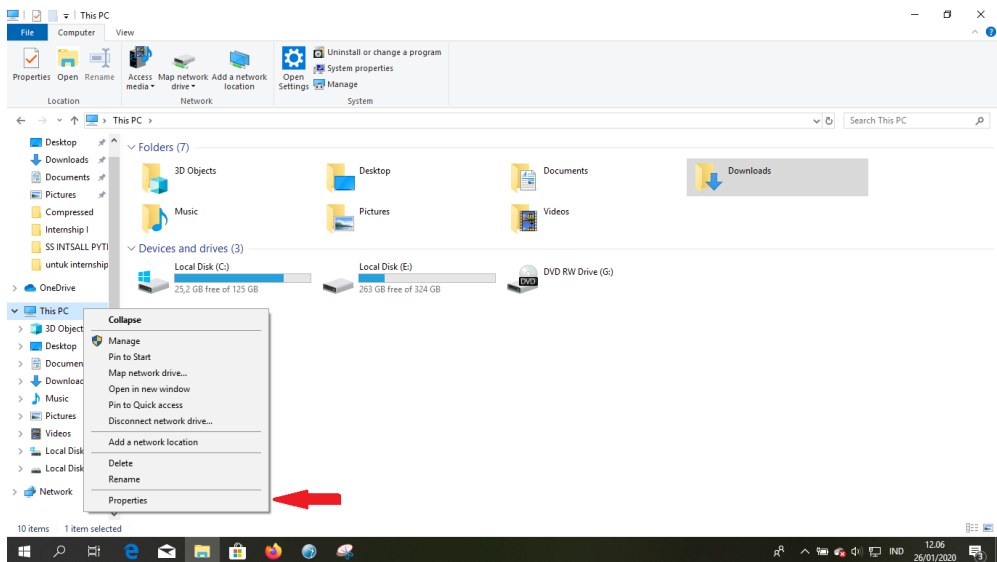
Ketujuh, selesai penginstallan *python*, kemudian klik “Finish”.



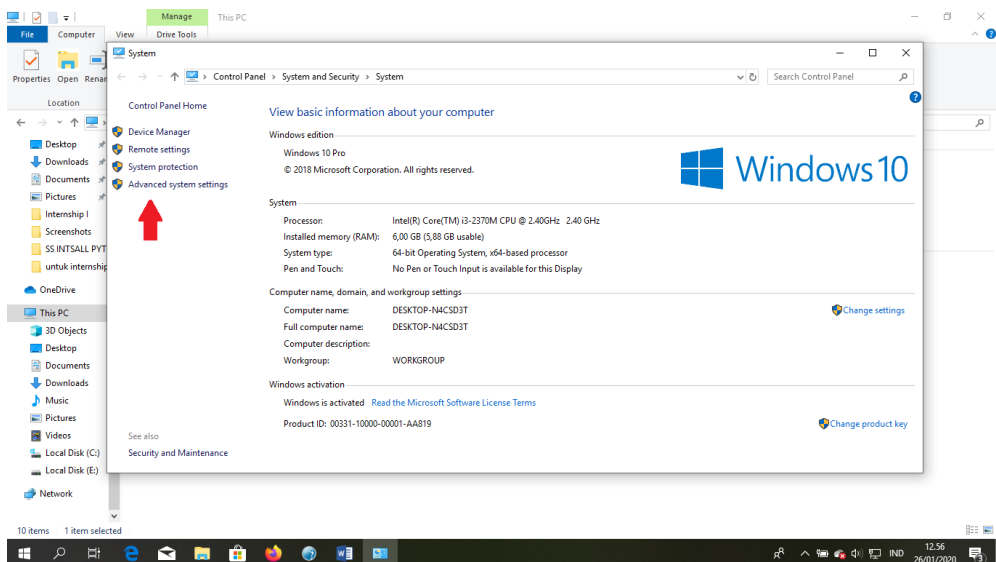
Kedelapan, atur environment variable klik kanan pada “This PC” seperti gambar dibawah ini.



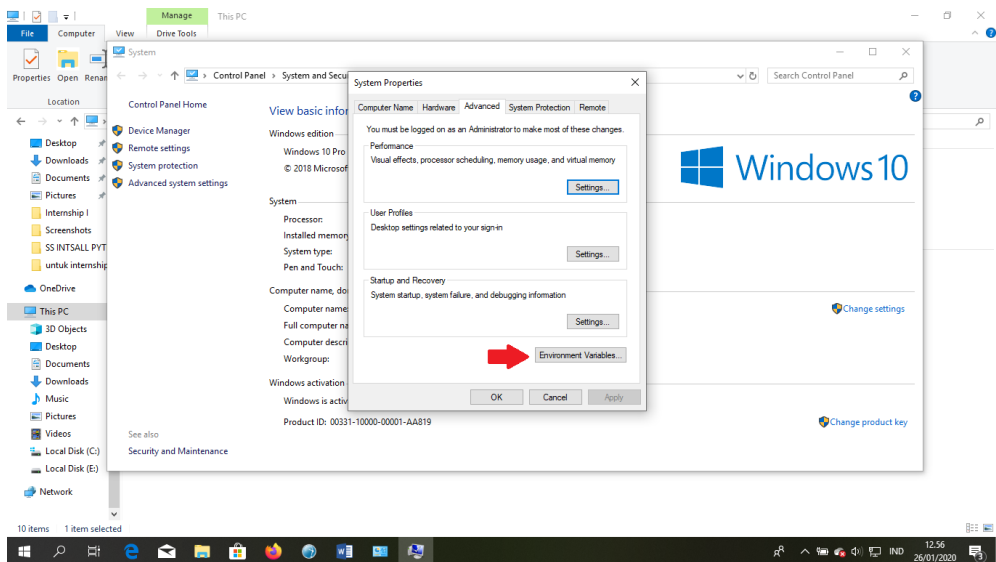
Kesembilan, kemudian klik pada “properties” seperti gambar dibawah ini.



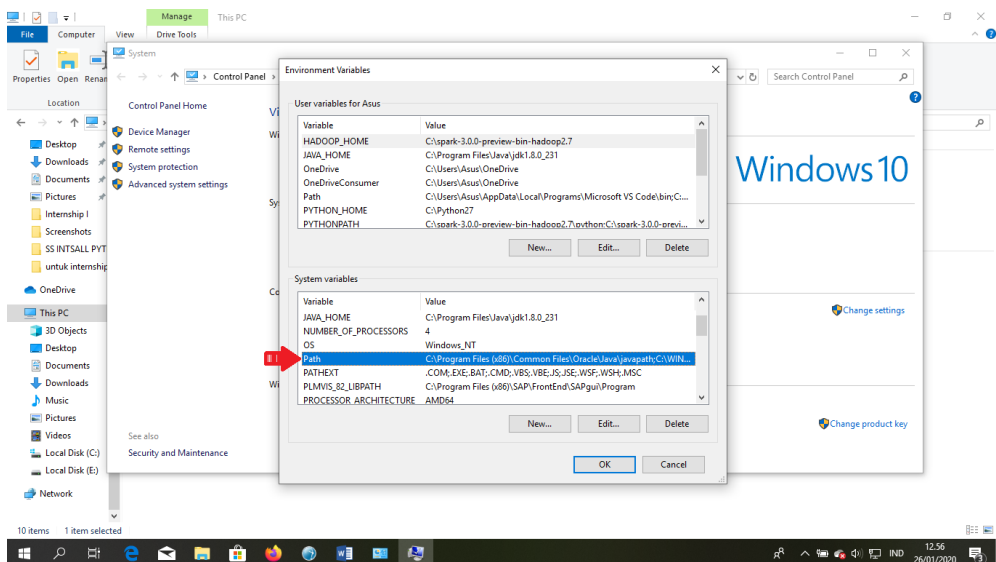
Kesepuluh, pilih “Advanced system setting”.



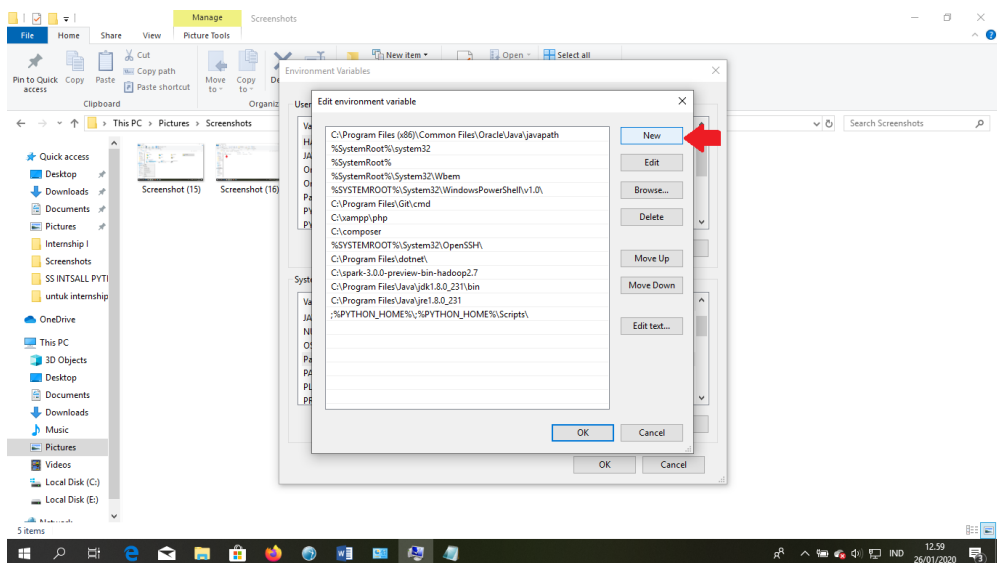
Kesebelas, klik “Enviroment Variables...”.



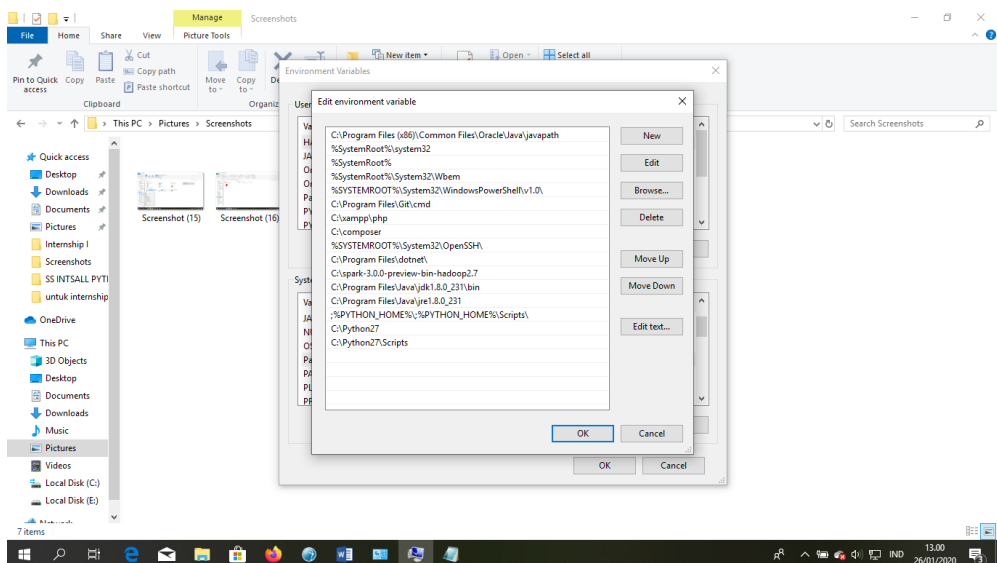
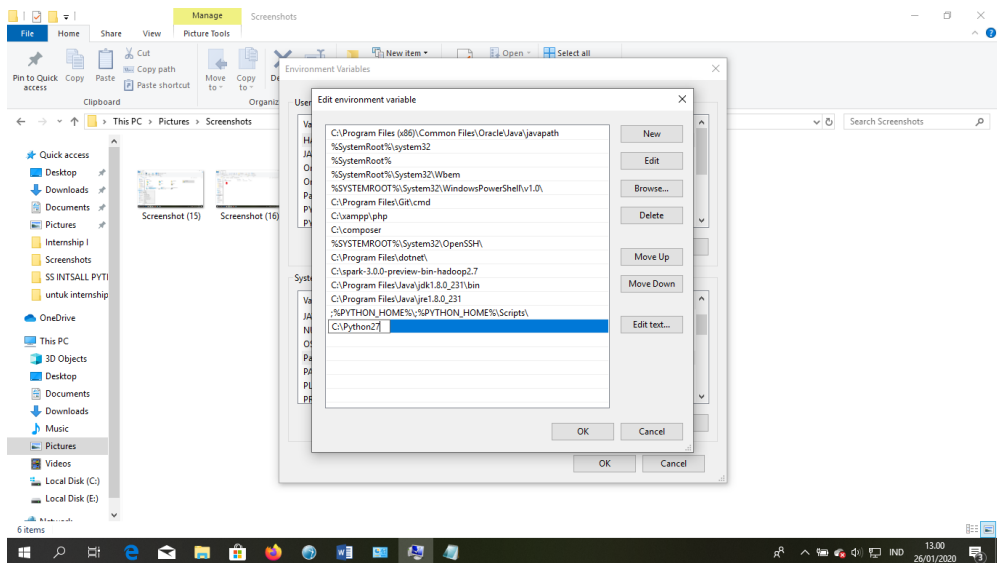
Keduabelas, lalu klik “Path”.



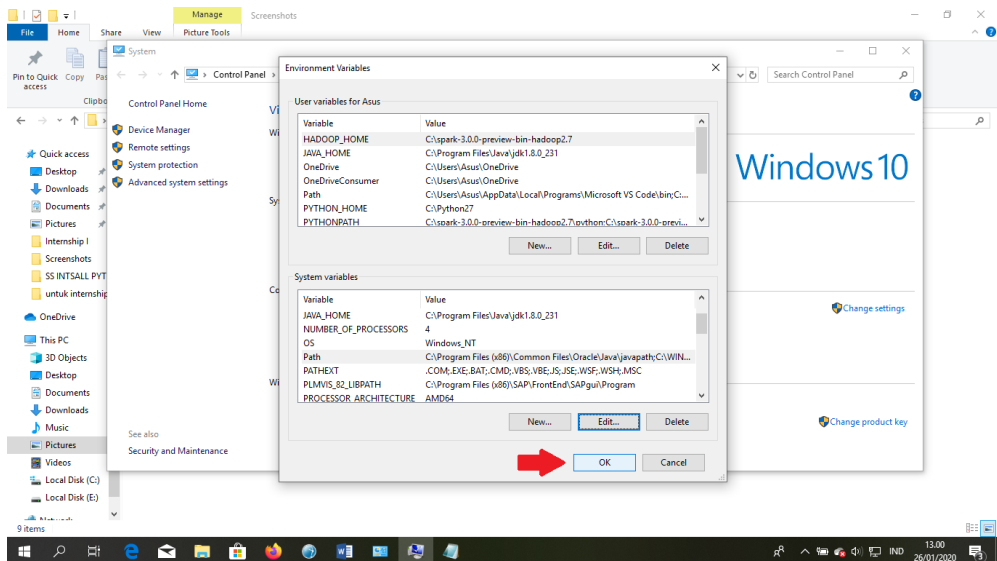
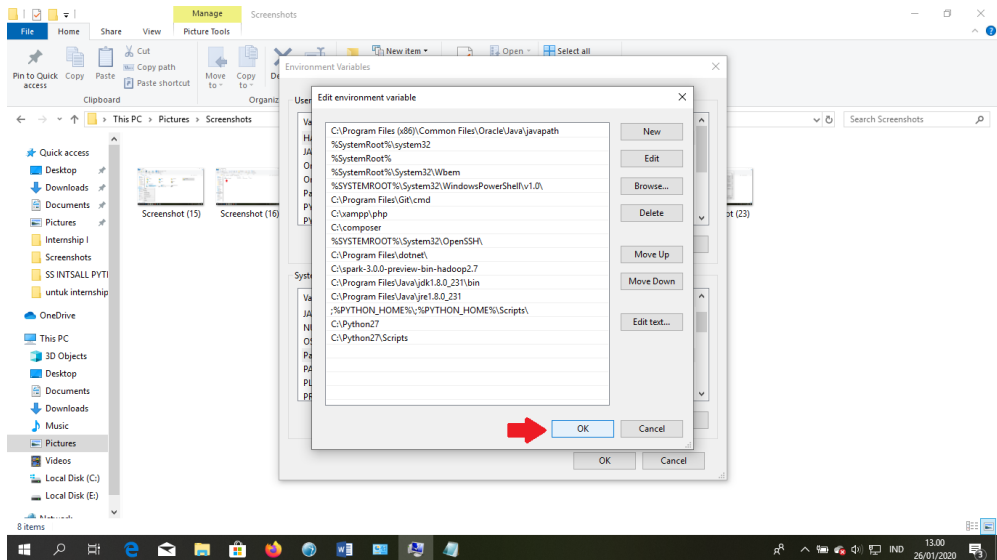
Ketigabelas, muncul seperti dibawah ini, kemudian klik “New” untuk membuat slot baru pada enviroment variable.



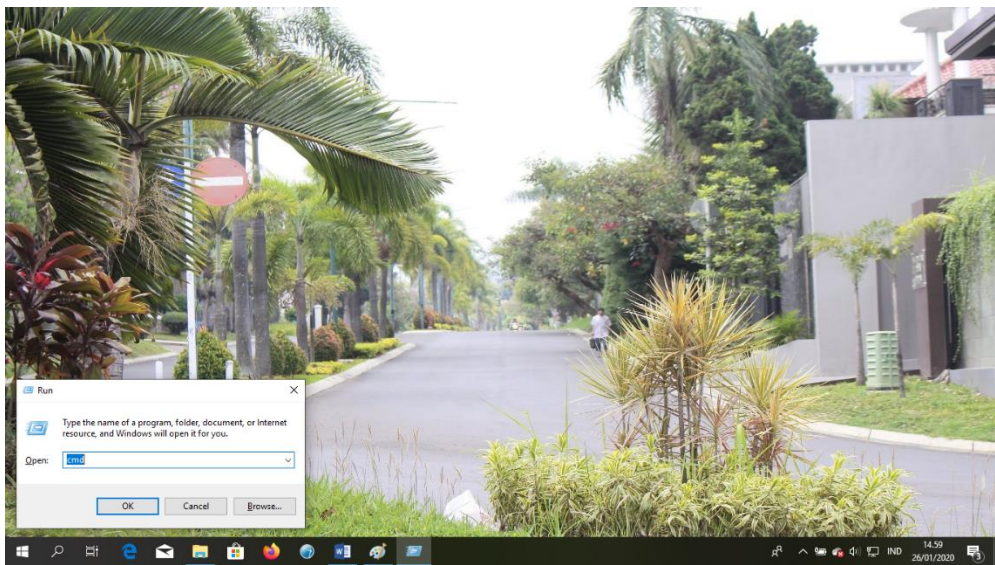
Keempatbelas, tuliskan atau ketik “C:\Python27” dan “C:\Python27\Scripts” seperti dibawah ini.



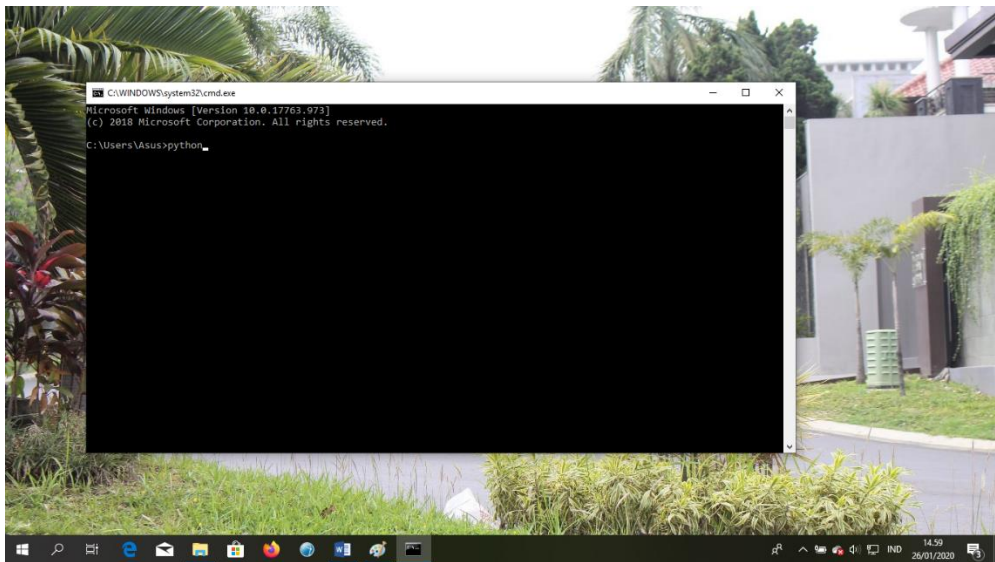
Kelimabelas, klik “Ok dan Ok”. Selesai membuat enviromentnya.



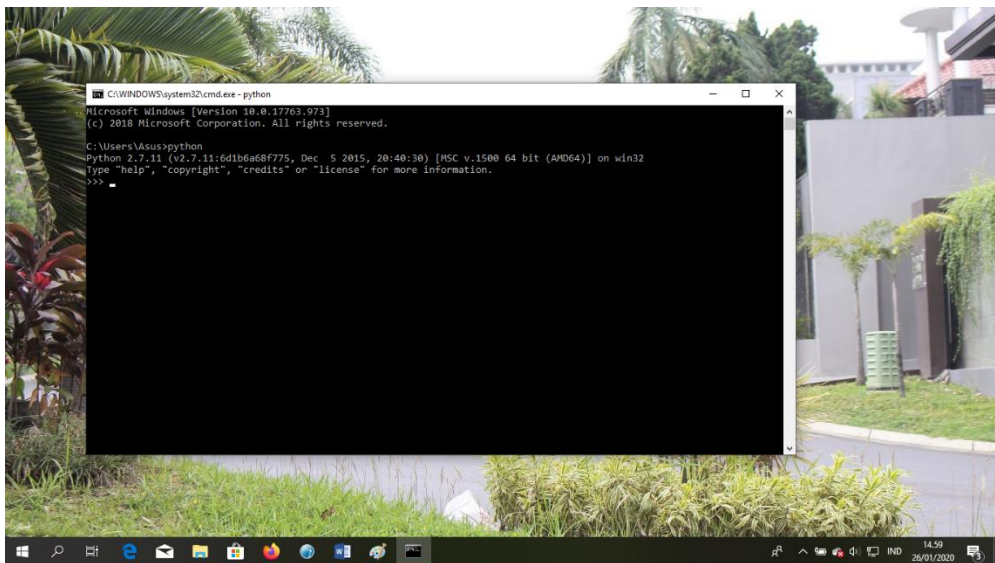
Terakhir kita jalankan python tersebut apakah sesuai atau tidak,
Pertama, Run Administrator “CMD”.



Kedua, ketik perintah “python” pada CMD.



Ketiga, jika tampil seperti digambar dibawah ini. Maka dapat disimpulkan pegininstallan dan pembuatan environment berhasil.



CHAPTER III

PENJELASAN DAN CODINGAN

PENJELASAN *PYTHON*

Python merupakan bahasa pemrograman yang dibuat dan dikembangkan oleh Guido Van Rossum pada tahun 1990 di Stichting Mathematisch Centrum (CWI). Bahasa pemrograman python itu bahasa pemrograman yang kekinian(terbaru), python juga bisa dikombinasikan dengan java dan arduino.

Python juga ada 2 versi yang berbeda yaitu versi *python 2.x* dan *python 3.x*, pada codingannya juga berbeda.

PERBEDAAN *PYTHON 2.X* DAN *PYTHON 3.X*

Perbedaan ypada codingan seperti dibawah ini.

Python 2.x :

Print 'Hallo!'

Python 3.x :

Print ('Hallo!')

Kesimpulannya pada *python 2.x* tidak menggunakan “()” akan tetapi *python 3.x* menggunakannya.

PERBEDAAN PYTHON DENGAN BAHASA PEMROGRAMAN LAIN.

Bahasa pemrograman python yaitu bahasa yang singkat, padat dan jelas. Berdasarkan codingan yang telah dibuat seperti dibawah ini.

Coding dari bahasa pemrograman C++ :

```
Int x = 2;
```

Coding dari bahasa pemrograman Python:

```
x = 2
```

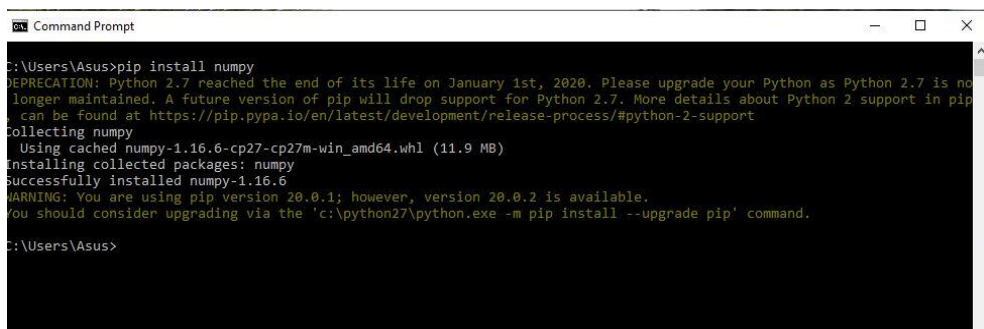
perbedaan sudah jelas dimana C++ harus menggunakan class dan type data. Akan tetapi tidak menggunakan class dan tipe data.

MACAM – MACAM *LIBRARY* YANG DIGUNAKAN

1. Numpy

Numpy adalah *library* pada *python* yang berfungsi untuk melakukan operasi vektor dan matriks dengan pengolahan array dan array multidimensi.

Cara instalnya yaitu ketik “pip install numpy” pada command prompt.

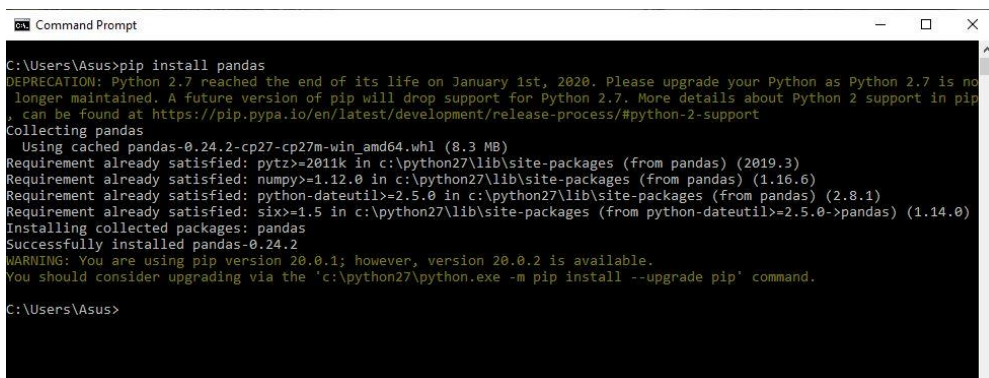


```
Command Prompt
C:\Users\Asus>pip install numpy
DEPRECATION: Python 2.7 reached the end of its life on January 1st, 2020. Please upgrade your Python as Python 2.7 is no longer maintained. A future version of pip will drop support for Python 2.7. More details about Python 2 support in pip can be found at https://pip.pypa.io/en/latest/development/release-process/#python-2-support
Collecting numpy
  Using cached numpy-1.16.6-cp27-cp27m-win_amd64.whl (11.9 MB)
Installing collected packages: numpy
Successfully installed numpy-1.16.6
WARNING: You are using pip version 20.0.1; however, version 20.0.2 is available.
You should consider upgrading via the 'c:\python27\python.exe -m pip install --upgrade pip' command.
C:\Users\Asus>
```

2. Pandas

Pandas merupakan librari berlisensi BSD dan open source yang menyediakan struktur data, analisis data yang mudah untuk digunakan dan berkinerja tinggi pada bahasa pemrograman python.

Cara instalnya ketik pada command prompt “pip install pandas” seperti dibawah ini.

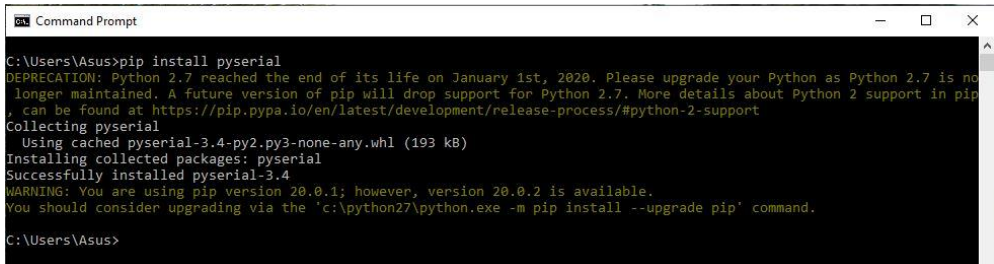


```
C:\Users\Asus>pip install pandas
DEPRECATION: Python 2.7 reached the end of its life on January 1st, 2020. Please upgrade your Python as Python 2.7 is no longer maintained. A future version of pip will drop support for Python 2.7. More details about Python 2 support in pip can be found at https://pip.pypa.io/en/latest/development/release-process/#python-2-support
Collecting pandas
  Using cached pandas-0.24.2-cp27-cp27m-win_amd64.whl (8.3 MB)
Requirement already satisfied: pytz>=2011k in c:\python27\lib\site-packages (from pandas) (2019.3)
Requirement already satisfied: numpy>=1.12.0 in c:\python27\lib\site-packages (from pandas) (1.16.6)
Requirement already satisfied: python-dateutil>=2.5.0 in c:\python27\lib\site-packages (from pandas) (2.8.1)
Requirement already satisfied: six>=1.5 in c:\python27\lib\site-packages (from python-dateutil>=2.5.0->pandas) (1.14.0)
Installing collected packages: pandas
Successfully installed pandas-0.24.2
WARNING: You are using pip version 20.0.1; however, version 20.0.2 is available.
You should consider upgrading via the 'c:\python27\python.exe -m pip install --upgrade pip' command.
C:\Users\Asus>
```

3. Serial/PySerial

PySerial adalah librari python yang dibuat untuk memudahkan kita dalam membuat program komunikasi data serial RS232 dalam bahasa python. Semua yang berkaitan dengan USB atau alat wajib di instal pyserial.

Cara instalnya ketik pada command prompt “pip install pyserial” seperti dibawah ini.

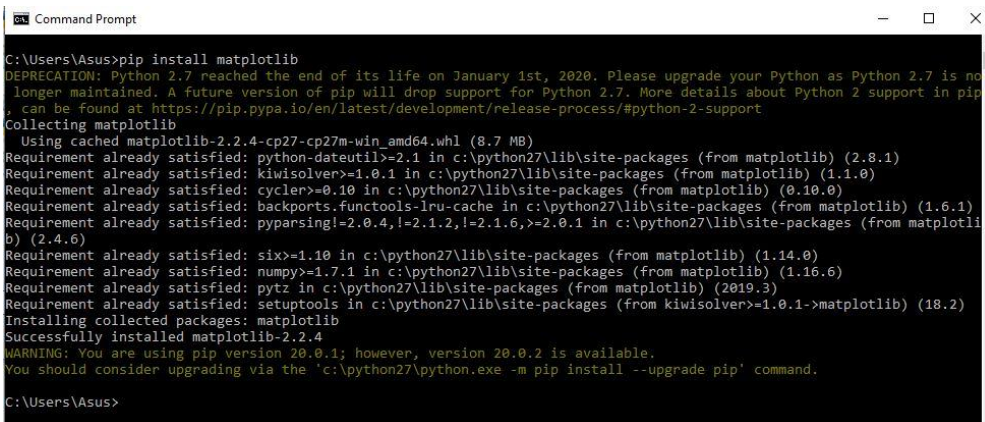


```
Command Prompt
C:\Users\Asus>pip install pyserial
DEPRECATION: Python 2.7 reached the end of its life on January 1st, 2020. Please upgrade your Python as Python 2.7 is no longer maintained. A future version of pip will drop support for Python 2.7. More details about Python 2 support in pip, can be found at https://pip.pypa.io/en/latest/development/release-process/#python-2-support
Collecting pyserial
  Using cached pyserial-3.4-py2.py3-none-any.whl (193 kB)
Installing collected packages: pyserial
Successfully installed pyserial-3.4
WARNING: You are using pip version 20.0.1; however, version 20.0.2 is available.
You should consider upgrading via the 'c:\python27\python.exe -m pip install --upgrade pip' command.
C:\Users\Asus>
```

4. Matplotlib

Matplotlib adalah librari plotting 2D Python yang menghasilkan gambar publikasi bermutu di berbagai format hardcopy dan lingkungan interaktif sepanjang platform.

Cara instalnya ketik pada command prompt “pip install matplotlib” seperti dibawah ini.



```
Command Prompt
C:\Users\Asus>pip install matplotlib
DEPRECATION: Python 2.7 reached the end of its life on January 1st, 2020. Please upgrade your Python as Python 2.7 is no longer maintained. A future version of pip will drop support for Python 2.7. More details about Python 2 support in pip, can be found at https://pip.pypa.io/en/latest/development/release-process/#python-2-support
Collecting matplotlib
  Using cached matplotlib-2.2.4-cp27-cp27m-win_amd64.whl (8.7 MB)
Requirement already satisfied: python-dateutil>=2.1 in c:\python27\lib\site-packages (from matplotlib) (2.8.1)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\python27\lib\site-packages (from matplotlib) (1.1.0)
Requirement already satisfied: cycler>=0.10 in c:\python27\lib\site-packages (from matplotlib) (0.10.0)
Requirement already satisfied: backports.functools-lru-cache in c:\python27\lib\site-packages (from matplotlib) (1.6.1)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in c:\python27\lib\site-packages (from matplotlib) (2.4.6)
Requirement already satisfied: six>=1.10 in c:\python27\lib\site-packages (from matplotlib) (1.14.0)
Requirement already satisfied: numpy>=1.7.1 in c:\python27\lib\site-packages (from matplotlib) (1.16.6)
Requirement already satisfied: pytz in c:\python27\lib\site-packages (from matplotlib) (2019.3)
Requirement already satisfied: setuptools in c:\python27\lib\site-packages (from kiwisolver>=1.0.1->matplotlib) (18.2)
Installing collected packages: matplotlib
Successfully installed matplotlib-2.2.4
WARNING: You are using pip version 20.0.1; however, version 20.0.2 is available.
You should consider upgrading via the 'c:\python27\python.exe -m pip install --upgrade pip' command.
C:\Users\Asus>
```

PERTAMA ADALAH CODINGAN MINDWAVE

```
import select, serial, threading
```

```
# Byte codes
```

```
CONNECT          = '\xc0'
```

```
DISCONNECT       = '\xc1'
```

```
AUTOCONNECT      = '\xc2'
```

```
SYNC             = '\xaa'
```

```
EXCODE           = '\x55'
```

```
POOR_SIGNAL      = '\x02'
```

```
ATTENTION        = '\x04'
```

```
MEDITATION      = '\x05'
```

```
BLINK            = '\x16'
```

```
HEADSET_CONNECTED = '\xd0'
```

```
HEADSET_NOT_FOUND = '\xd1'
```

```
HEADSET_DISCONNECTED = '\xd2'
```

```
REQUEST_DENIED   = '\xd3'
```

```
STANDBY_SCAN     = '\xd4'
```

```
RAW_VALUE        = '\x80'
```

```
# Status codes
```

```
STATUS_CONNECTED = 'connected'
```

```
STATUS_SCANNING    = 'scanning'
```

```
STATUS_STANDBY     = 'standby'
```

```
class Headset(object):
```

```
    """
```

```
    A MindWave Headset
```

```
    """
```

```
class DongleListener(threading.Thread):
```

```
    """
```

```
    Serial listener for dongle device.
```

```
    """
```

```
    def __init__(self, headset, *args, **kwargs):
```

```
        """Set up the listener device."""
```

```
        self.headset = headset
```

```
        super(Headset.DongleListener, self).__init__(*args,  
**kwargs)
```

```
    def run(self):
```

```
        """Run the listener thread."""
```

```
        s = self.headset.dongle
```

```
        # Re-apply settings to ensure packet stream
```

```
s.write(DISCONNECT)
d = s.getSettingsDict()
for i in xrange(2):
    d['rtscts'] = not d['rtscts']
    s.applySettingsDict(d)

while True:
    # Begin listening for packets
    try:
        if s.read() == SYNC and s.read() == SYNC:
            # Packet found, determine plength
            while True:
                plength = ord(s.read())
                if plength != 170:
                    break
            if plength > 170:
                continue

            # Read in the payload
            payload = s.read(plength)

            # Verify its checksum
            val = sum(ord(b) for b in payload[:-1])
```

```
        val &= 0xff
        val = ~val & 0xff
        chksum = ord(s.read())

        #if val == chksum:
            if True: # ignore bad checksums
                self.parse_payload(payload)
except (select.error, OSError):
    break
except serial.SerialException:
    s.close()
    break

def parse_payload(self, payload):
    """Parse the payload to determine an action."""
    while payload:
        # Parse data row
        excode = 0
        try:
            code, payload = payload[0], payload[1:]
        except IndexError:
            pass
        while code == EXCODE:
```

```
# Count excode bytes
excode += 1

try:
    code, payload = payload[0], payload[1:]
except IndexError:
    pass

if ord(code) < 0x80:
    # This is a single-byte code
    try:
        value, payload = payload[0], payload[1:]
    except IndexError:
        pass

    if code == POOR_SIGNAL:
        # Poor signal
        old_poor_signal = self.headset.poor_signal
        self.headset.poor_signal = ord(value)

        if self.headset.poor_signal > 0:
            if old_poor_signal == 0:
                for handler in \
                    self.headset.poor_signal_handlers:
                    handler(self.headset,
                           self.headset.poor_signal)
            else:
```

```
        if old_poor_signal > 0:
            for handler in \
                self.headset.good_signal_handlers:
                handler(self.headset,
                        self.headset.poor_signal)
    elif code == ATTENTION:
        # Attention level
        self.headset.attention = ord(value)
        for handler in self.headset.attention_handlers:
            handler(self.headset, self.headset.attention)
    elif code == MEDITATION:
        # Meditation level
        self.headset.meditation = ord(value)
        for handler in self.headset.meditation_handlers:
            handler(self.headset, self.headset.meditation)
    elif code == BLINK:
        # Blink strength
        self.headset.blink = ord(value)
        for handler in self.headset.blink_handlers:
            handler(self.headset, self.headset.blink)
    else:
        # This is a multi-byte code
        try:
```

```

        vlength, payload = ord(payload[0]), payload[1:]
    except IndexError:
        continue

    value, payload = payload[:vlength],
payload[vlength:]

# Multi-byte EEG and Raw Wave codes not included
# Raw Value added due to Mindset Communications
Protocol

if code == RAW_VALUE:
    try:
        anu = value[0]
        itu = value[1]
    except IndexError:
        anu = "x"
        itu = "x"
    raw=ord(anu)*256+ord(itu)
    if (raw>=32768):
        raw=raw-65536
    self.headset.raw_value = raw
    for handler in self.headset.raw_value_handlers:
        handler(self.headset, self.headset.raw_value)

if code == HEADSET_CONNECTED:
    # Headset connect success

```

```
        run_handlers = self.headset.status !=
STATUS_CONNECTED

        self.headset.status = STATUS_CONNECTED
        self.headset.headset_id = value.encode('hex')
        if run_handlers:
            for handler in \
                self.headset.headset_connected_handlers:
                handler(self.headset)
    elif code == HEADSET_NOT_FOUND:
        # Headset not found
        if vlength > 0:
            not_found_id = value.encode('hex')
            for handler in \
                self.headset.headset_notfound_handlers:
                handler(self.headset, not_found_id)
        else:
            for handler in \
                self.headset.headset_notfound_handlers:
                handler(self.headset, None)
    elif code == HEADSET_DISCONNECTED:
        # Headset disconnected
        headset_id = value.encode('hex')
        for handler in \
```

```
        self.headset.headset_disconnected_handlers:
            handler(self.headset, headset_id)
elif code == REQUEST_DENIED:
    # Request denied
    for handler in
self.headset.request_denied_handlers:
        handler(self.headset)
elif code == STANDBY_SCAN:
    # Standby/Scan mode
    try:
        byte = ord(value[0])
    except IndexError:
        byte = None
    if byte:
        run_handlers = (self.headset.status !=
                        STATUS_SCANNING)
        self.headset.status = STATUS_SCANNING
        if run_handlers:
            for handler in self.headset.scanning_handlers:
                handler(self.headset)
    else:
        run_handlers = (self.headset.status !=
                        STATUS_STANDBY)
```

```
        self.headset.status = STATUS_STANDBY
        if run_handlers:
            for handler in self.headset.standby_handlers:
                handler(self.headset)

def __init__(self, device, headset_id=None, open_serial=True):
    """Initialize the headset."""
    # Initialize headset values
    self.dongle = None
    self.listener = None
    self.device = device
    self.headset_id = headset_id
    self.poor_signal = 255
    self.attention = 0
    self.meditation = 0
    self.blink = 0
    self.raw_value = 0
    self.status = None

    # Create event handler lists
    self.poor_signal_handlers = []
    self.good_signal_handlers = []
```

```
self.attention_handlers = []
self.meditation_handlers = []
self.blink_handlers = []
self.raw_value_handlers = []
self.headset_connected_handlers = []
self.headset_notfound_handlers = []
self.headset_disconnected_handlers = []
self.request_denied_handlers = []
self.scanning_handlers = []
self.standby_handlers = []

# Open the socket
if open_serial:
    self.serial_open()

def connect(self, headset_id=None):
    """Connect to the specified headset id."""
    if headset_id:
        self.headset_id = headset_id
    else:
        headset_id = self.headset_id
    if not headset_id:
        self.autoconnect()
```

```
        return

        self.dongle.write(".join([CONNECT,
headset_id.decode('hex')]))

def autoconnect(self):
    """Automatically connect device to headset."""
    self.dongle.write(AUTOCONNECT)

def disconnect(self):
    """Disconnect the device from the headset."""
    self.dongle.write(DISCONNECT)

def serial_open(self):
    """Open the serial connection and begin listening for
data."""
    # Establish serial connection to the dongle
    if not self.dongle or not self.dongle.isOpen():
        self.dongle = serial.Serial(self.device, 115200)

    # Begin listening to the serial device
    if not self.listener or not self.listener.isAlive():
        self.listener = self.DongleListener(self)
        self.listener.daemon = True
        self.listener.start()
```

```
def serial_close(self):  
    """Close the serial connection."""  
    self.dongle.close()
```

KEDUA CODINGAN CONNECTING

```
import serial

ser = serial.Serial("COM8", 9600)
temp = ""
while 1:
    data=ser.readline()
    print data
    i = int(data, 16)
    data=data.script()
    print(data)
    if data[:1]=="[":
        print "\a"
```

KETIGA CODINGAN TESTING

```
import mindwave, time
```

```
headset = mindwave.Headset('COM8','1425')
```

```
time.sleep(2)
```

```
headset.connect()
```

```
print "Connecting..."
```

```
while headset.status != 'connected':
```

```
    time.sleep(0.5)
```

```
    if headset.status == 'standby':
```

```
        headset.connect()
```

```
        print "Retrying connect.."
```

```
print " Connetcted."
```

```
while True:
```

```
    #print "Auttention: %s, meditation: %s" % (headset.attention,  
headset.meditation)
```

```
    #print headset.serial_open()
```

```
    print headset.raw_value#"raw_value: %s" %  
(headset.raw_value)
```

```
    time.sleep(0.1) #pause 0.5 seconds
```

KEEMPAT CODINGAN RUNNING

```
import os, mindwave, time

port="COM8"
mid="1425"
rate=0.001953125
namafile="hasilnya.csv"
runfile="run"

open(runfile, 'a').close()
headset = mindwave.Headset(port,mid)

time.sleep(2)

headset.connect()
print "Connecting..."

while headset.status != 'connected':
    time.sleep(0.5)
    if headset.status == 'standby':
```

```
    headset.connect()
    print "Retrying connect.."
print " Connetcted."

f=open(namafile,'a+')

while True:
    if os.path.exists(runfile)!=True:
        f.close()
        break
    try:
        while True:
            f.write(str(headset.raw_value)+'0\n')
            time.sleep(rate)
            if os.path.exists(runfile)!=True:
                f.close()
                break
        except KeyboardInterrupt:
            f.write(str(headset.raw_value)+'1\n')
            if os.path.exists(runfile)!=True:
                f.close()
                break
    continue
```

KEENAM CODINGAN GRAFIK

Codingan ini untuk menampilkan matplotlib yang nanti akan muncul grafik gelombang otak dan amplitudo.

CODING:

```
from matplotlib import pyplot as plt
```

```
from matplotlib import style
```

```
import numpy as np
```

```
style.use('ggplot')
```

```
x,y = np.loadtxt('asep.csv', unpack = True, delimiter = ',')
```

```
plt.plot(x,y)
```

```
plt.title('Brainwave data result test')
```

```
plt.ylabel('Y axis')
```

```
plt.xlabel('X axis')
```

```
plt.show()
```

CHAPTER IV

LAPORAN INTERNSHIP I

PENDAHULUAN

1. Latar Belakang

EEG adalah sebuah alat elektromedik yang dapat digunakan untuk mencatat dan menganalisa aktifitas gelombang otak dalam kurun waktu tertentu dengan penempatan elektrode di kepala. Pada suatu penelitian ditunjukkan bahwa hanya dalam 10-20 menit bermain video game dapat meningkatkan kinerja otak pada rangsangan. Apakah kondisi gelombang otak tersebut akan berbeda dengan kondisi otak ketika bermain games dengan berbeda aliran, mengenai dampak dari bermain games akan menunjukkan hasil positif atau negatif. Pengukuran bentuk konsentrasi tersebut diukur dari informasi sinyal Electroencephalogram (EEG). Saat manusia beraktivitas, otak akan bekerja dan menghasilkan sinyal otak. Sinyal otak ini muncul dan mengakibatkan adanya aktivitas elektrik. Berdasarkan rentang frekuensi, sinyal EEG dapat dibagi ke dalam lima jenis gelombang yaitu delta, theta, alfa, beta, gamma. Dalam bermain game crash team racing (CTR) manusia akan mengalami perbedaan gelombang pada permainan tersebut.

Pada permasalahan tersebut, Internship 1 ini bertujuan untuk menganalisis gelombang otak saat bermain game dengan berbeda aliran. Agar mengetahui berdampak positif dan juga negatif berindikasi menyebabkan kelelahan pada otak dan mencegah kelebihan dalam bermain game agar dapat meminimalisir hal yang buruk dalam bermain game. Penelitian Internship 1 ini menggunakan EEG dengan menganalisis gelombang delta, theta, alfa, beta, gamma. Karakteristik gelombang sinyal EEG terbagi berdasarkan daerah frekuensi yang menunjukkan dominasi aktivitas yang sedang dialami.

Melalui penelitian ini dapat memperlihatkan frekuensi gelombang otak pada saat bermain game crash team racing (CTR), gelombang

otak pada saat bermain setiap orang memiliki sinyal berbeda, kemudian setelah mengetahui masing-masing perbedaanya selanjutnya akan menghitung algoritma c4.5.

Melihat dari permasalahan yang dijabarkan sebelumnya, diperlukan suatu alternatif yang dapat mencegah terjadinya dampak buruk akibat bermain game. Salah satu alternatif tersebut adalah dengan mendeteksi suatu indikasi yang menyebabkan kelelahan pada otak ketika bermain game. Oleh karena itu pada Internship 1 ini dirancang suatu sistem klasifikasi dari kelelahan otak menggunakan perhitungan algoritma c4.5 kemudian akan mengambil keputusan.

2. Identifikasi Masalah

Identifikasi masalah berdasarkan latar belakang adalah:

1. Bagaimana cara memonitor gelombang otak pemain game ctr pada yang sedang bermain?
2. Bagaimana mengetahui karakteristik otak pada orang (subjek) yang berbeda?

3. Tujuan dan Manfaat

Berdasarkan latar belakang dan identifikasi masalah yang ada, maka akan memberikan tujuan dan manfaat sebagai berikut :

1. Untuk memonitor aktivitas gelombang otak dalam kondisi bermain game ctr.
2. Dengan mengetahui sinyal gelombang otak pada orang yang bermain game ctr , sehingga dapat terlihat perbedaan sinyal gelombang otak pada pemain game CTR.

4. Ruang Lingkup

Ruang lingkup penelitian yang dilakukan oleh peneliti sebagai berikut :

1. Data yang diambil hanya saat bermain game Crash Team Racing (CTR).
2. Menggunakan alat perekaman sinyal Neurosky Mindwave
3. Menggunakan bahasa pemrograman python untuk program perekaman datanya
4. Pengambilan data EEG hanya dilakukan pada subjek dengan jenis kelamin laki-laki

LANDASAN TEORI

1. Electroencephalography

Electroencephalography (EEG) adalah tes yang banyak digunakan untuk mendeteksi gelombang otak. Sinyal ini direkam oleh perangkat \textit{Electroencephalogram}, Adalah perangkat keras yang berfungsi merekam aktivitas listrik dari gelombang otak. Prinsip kerja EEG ini mendeteksi aktivitas listrik dari otak orang-orang dengan rekaman elektroda perak yang dipasang oleh teknisi terlatih di kulit kepala.

Kalangan medis menggunakan EEG, antara lain, untuk mendiagnosis penyakit yang berkaitan dengan otak dan gangguan psikologis. EEG juga diterapkan untuk mendeteksi pola pikiran atau kondisi mental seseorang. Pengamatan visual dari sinyal EEG secara langsung adalah tantangan mengingat amplitudo sinyal EEG sangat rendah, dan desainnya sangat rumit. Selain itu, sinyal EEG sangat dipengaruhi oleh berbagai variabel, termasuk kondisi mental, kesehatan, aktivitas pasien, lingkungan perekaman, gangguan listrik dari organ lain, rangsangan eksternal, dan usia pasien \cite {turnip2017detection}. Sifat sinyal EEG, secara umum, adalah non-stasioner dan acak, sehingga menambah kompleksitas pemrosesan sinyal EEG. Namun, klasifikasi sinyal EEG terhadap perubahan variabel tertentu dapat menjelaskan fungsi kerja otak dan menangkap perubahan aktivitas otak dengan variabel yang bersangkutan

2. Data Mining

Data mining adalah proses yang menggunakan teknik statistik, matematika, kecerdasan buatan dan machine learning untuk mengekstrak dan mendefinisikan informasi yang bermanfaat dan pengetahuan dari seluruh yang berkaitan dengan besar.

3. Algoritma C4.5

Algoritma C4.5 merupakan algoritma klasifikasi pohon keputusan yang banyak digunakan karena memiliki perbedaan dan kelebihan dari algoritma yang lain. Kelebihannya yaitu memiliki tingkat akurasi yang bisa diterima, efisiensi dalam mengatasi atribut bertipe diskret dan dapat mengatasi atribut yang bertipe diskret dan numerik, perbedaannya dari segi penghitungan yaitu menggunakan rumus entropy dan gain. \cite{kamagi2014implementasi}. Algoritma C4.5 sendiri merupakan pengembangan dari algoritma ID3.

Algoritma ID3 merupakan algoritma yang dipergunakan untuk membuat sebuah decision tree atau pohon keputusan. Algoritma ini ditemukan oleh J. Ross Quinlan (1979), dengan memanfaatkan Teori Informasi atau Information Teori milik Shanon. ID3 sendiri merupakan singkatan dari Iterative Dichotomiser 3. Decision tree menggunakan struktur hierarki untuk pembelajaran supervised. Dalam pemrosesan decision tree dimulai dari root node hingga leaf node yang dilakukan secara rekursif. Di mana setiap percabangan menyatakan suatu kondisi yang harus dipenuhi dan pada setiap ujung pohon menyatakan kelas dari suatu data.

Algoritma C4.5 juga mempunyai konsep pohon pengambil keputusan seperti dibawah ini:

1. Pilih atribut sebagai akar
2. Buat cabang untuk masing-masing nilai
3. Bagi kasus dalam cabang
4. Ulang proses untuk masing-masing cabang, sampai semua kasus pada cabang memiliki kelas yang sama.

Sebelum menjadi algoritma c4.5, konsep dari algoritma id3 yaitu seperti dibawah ini:

1. Tahan (robust) terhadap data noise, sehingga mencegah adanya.
2. Mampu menangani variabel dengan tipe diskrit maupun kontinu.
3. Mampu menangani variable yang memiliki missing value.
4. Dapat memangkas cabang dari pohon keputusan.

3.1 Entropy

Entropy merupakan metode yang dapat memberikan nilai yang alternatif bagi pengambil keputusan. Metode Entropy dapat memberikan bobot terhadap data yang kualitatif maupun kuantitatif. Sebelum diolah semua data akan dinormalisasi sehingga memberikan range antara 0-1, dan tidak memerlukan range dari setiap kriteria harus sama.

penjelasan:

rumus

S : Total Kasus

sigma : Jumlah bilangan

P_j = adalah probabilitas yang di dapat dari $\text{Sum}(Y_a / \text{di isi } 1)$
dibagi Total keseluruhan

3.2 Information Gain

Information gain atau gain merupakan salah satu attribute selection measure yang digunakan untuk memilih test attribute tiap node pada tree. Atribut dengan informasi gain tertinggi dipilih sebagai test atribut dari suatu node, gain juga merupakan hasil dari perhitungan entropy sehingga hasil itu biasanya untuk menentukan keputusan.\cite{ginting2014analisis}

rumus

Penjelasan:

S = Total Kasus

n = Jumlah partisi S

A = Atribut

$[S_i]$ = proporsi S_i terhadap S

$[S]$ = Jumlah kasus dalam S

4. Gelombang Otak

Gelombang adalah getaran yang berpindah. gelombang selama menjalar(merambat), hanya memindahkan energi, akan tetapi materi selama berpindah selalu memindahkan massa dan energi tersebut. \cite{susanti2014profil}. Otak adalah organ tubuh manusia yang posisinya ditempatkan secara terhormat di bagian atas tubuh manusia dan terlindungi dengan kokoh di bagian dalam tengkorak (batok) kepala. \cite{purwati2016optimalisasi}. Gelombang otak merupakan getaran yang berpindah(merambat), yang memlalui sel-sel otak.

4.1 Delta

Delta yaitu gelombang otak dengan rentang frekuensi antara 0.5 - 4 Hz. Delta muncul pada saat seseorang dalam keadaan tidur nyenyak. Sinyal delta dapat pula mengindikasikan adanya cacat fisik di otak.

4.2 Theta

Theta yaitu gelombang otak dengan rentang fiekuensi antara 4.5 - 8 Hz. Theta muncul pada seseoarang yang mengalami stress secara emosional terutama sedih atau kekecewaan.

4.3 Alpha

Alpha yaitu gelombang otak dengan rentang frekuensi antara 8.5 - 13 Hz. Alpha mengindikasikan seseorang dalam keadaan kondisi pikiran relaks dan tidak berpikir keras. Gelombang ini paling besar muncul pada bagian otak daerah occipital cortex dan juga pada bagian frontal.

4.4 Beta

Beta yaitu gelombang otak dengan rentang frekuensi antara 13.5 - 29.5 Hz. Beta adalah gelombang otak yang biasanya terjadi pada saat seseorang sedang aktif berpikir, aktif konsentrasi atau sedang fokus dalam memecahkan suatu permasalahan. Beta terbagi menjadi 2 kelompok berdasarkan rentang frekuensi yaitu Beta Low (13-20 Hz) dan Beta High (20 - 30 Hz).

4.5 Gamma

Gamma yaitu gelombang otak dengan rentang frekuensi antara 30 - 45 Hz. Gelombang otak ini terjadi pada saat seseorang mengalami aktivitas yang sangat berat, keadaan sangat panik ataupun ketakutan dalam kondisi kesadaran penuh.

5. Neurosky Mindwave

Neurosky adalah perusahaan yang berdiri pada tahun 2004 di Silicon Valley, California, Amerika Serikat. Perusahaan ini berfokus pada tujuan utamanya yaitu memanfaatkan teknologi Brain Computing interface (BCI) untuk dipasarkan pada konsumen secara luas. Neurosky mengadaptasi teknologi EEG dan mengembangkannya untuk dapat digunakan dalam beberapa bidang yang memenuhi permintaan komersial, seperti untuk entertainment, kesehatan, serta otomotif.

Terobosan terbesar Neurosky adalah ketika mereka berhasil mengembangkan perangkat EEG dengan harga yang cukup murah. Perangkat tersebut meliputi : sensor kening yang berfungsi menangkap input sinyal dan gelombang yang dihasilkan oleh otak, perangkat lunak dan perangkat keras yang mampu mengurangi dan memfilter setiap elektrik noise, serta think Gear Chip yang berfungsi sebagai sirkuit pemrosesan sinyal dan penghasil output. Seluruh perangkat tersebut dapat ditanamkan pada perangkat sederhana serupa headset (Karvinan, Tero dan Kimmo Karnivan, 2011).

Neurosky Mindwave Headset ini terdiri beberapa bagian penting yang dapat mempengaruhi hasil pembacaan yaitu klip pada telinga (ear clip), sensor yang berada pada ujung lengan (Sensor arm) dan tiga bagian yang berada di ujung (sensor arm), dan lengan sensor. Referensi dan ground elektroda headset berada pada klip telinga (ear clip) dan sensor elektroda yang digunakan untuk membaca gelombang otak berada pada pengait depan (sensor arm) yang bertumpu pada dahi di atas mata.

Poor Signal Quality menjelaskan sinyal kurang baik yang diukur oleh Headset (ThinkGear) yang berkisar pada nilai 1-200. Pada saat nilai poor signal quality menunjukkan nilai bukan nol menandakan adanya noise yang terdeteksi, semakin tinggi nilai yang ditunjukkan maka semakin besar pula noise. Sementara, nilai 200 memiliki arti khusus, bahwa sensor tidak menyentuh kulit kepala. Nilai poor signal quality dapat digunakan untuk beberapa aplikasi yang sensitif seperti mengedipkan mata.

ANALISIS ORGANISASI PERUSAHAAN

1. SEJARAH PERUSAHAAN

Pada Tanggal 9 Mei 2018. Dosen-dosen dan Formatur IRC Politeknik Pos Indonesia melakukan kegiatan pembentukan formatur dan konsep IRC di Laboratorium IRC untuk melaksanakan beberapa rangkaian penelitian. IRC merupakan program penelitian yang diinisiasi oleh para dosen Teknik Informatika Politeknik Pos Indonesia. Program ini banyak berkaitan dengan teknologi serta penelitian.

Dengan menggunakan teknologi yang sudah stabil, proses penelitian dapat berjalan serta menyelesaikan permasalahan yang ada disekitar. Beberapa hal terkait dengan penelitian sudah dikonsepskan dari dosen dan formatur IRC untuk mewujudkan VISI dan MISI IRC. Salah satu MISI IRC yaitu melakukan peningkatan tentang publikasi karya ilmiah dari mahasiswa dan dosen kedalam jurnal bereputasi internasional. Gambar dibawah ini merupakan logo dari IRC:



2. Visi dan Misi Perusahaan

2.1 Visi

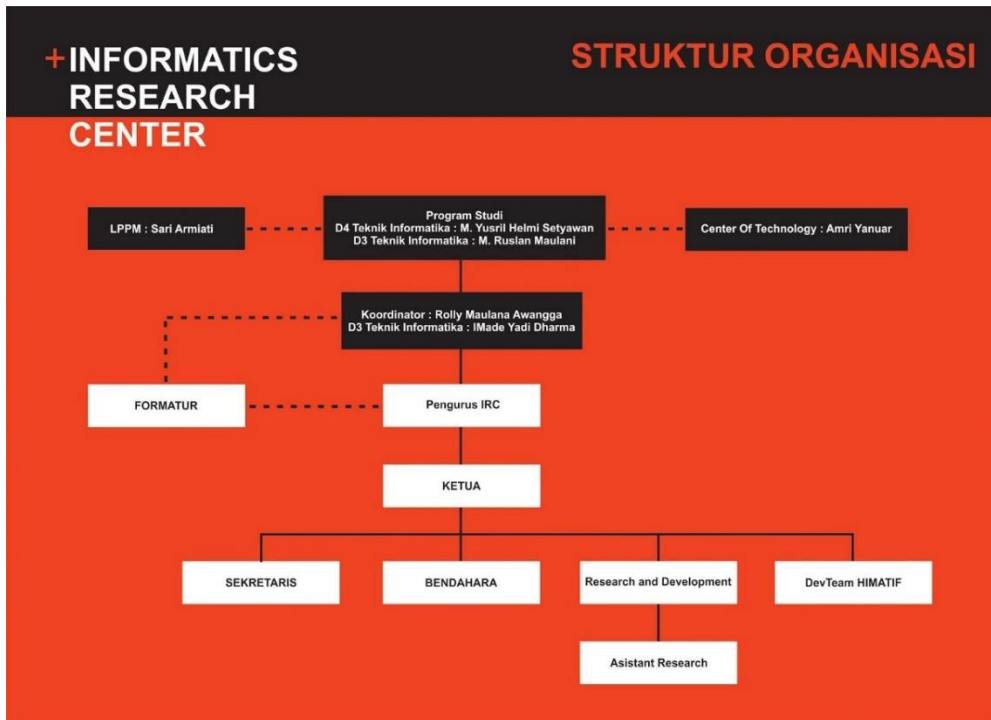
Menjadi Pusat Kolaborasi Nasional Tri Dharma Perguruan Tinggi se-Indonesia pada tahun 2020.

2.2 Misi

1. Peningkatan Publikasi Karya Ilmiah Mahasiswa dan Dosen dalam Publikasi Jurnal Bereputasi Internasional.
2. Pemanfaatan Teknologi hasil riset kepada masyarakat secara optimal.
3. Percepatan Pengembangan aplikasi Open Source.
4. Sistem Kerja Crowd Sourcing.
5. Selalu Update dengan berita dan informasi tentang teknologi.

3. Struktur Organisasi dan Job Description Perusahaan

Struktur Organisasi IRC gambar dibawah ini merupakan struktur organisasi di IRC



4. Job Description Perusahaan

1. Deskripsi pekerjaan yang ada di IRC meliputi:
2. Membuat Produk dalam bentuk alat
3. Menilai uts kelas 3a, 3b dan 3c
4. Menilai uts kelas 2b dan 2c
5. Menilai tugas mata kuliah pemrograman II kelas 2b
6. Menilai tugas mata kuliah pemrograman II kelas 2c
7. Masuk kelas mata kuliah pemrograman II
8. Mengolah data EEG
9. Membantu pembuatan buku dan haki dosen
10. Dan melakukan pekerjaan lainnya yang diperintahkan oleh Bapak Rolly Maulana Awangga S.T.,M.T

5. Deskripsi dan Ruang Lingkup Internship I

Deskripsi Pekerjaan Internship I

Dalam kegiatan magang kali ini, penulis melakukan kegiatan Internship II di IRC dan menempati posisi magang sebagai peneliti, karena penulis melakukan penelitian mengenai gelombang otak pada pemain game crash team racing.

6. Ruang Lingkup Internship I

Pada internship I kali ini, penulis melakukan riset untuk meneliti gelombang otak, menggunakan alat EEG Neurosky dan merekam datanya ke dalam format csv dan nanti akan diolah ke dalam bentuk grafik. Internship I ini, kasus yang diambil adalah meneliti gelombang otak pada pemain game crash team racing.

7. Faktor Pendukung dan Penghambat

7.1 Faktor Pendukung

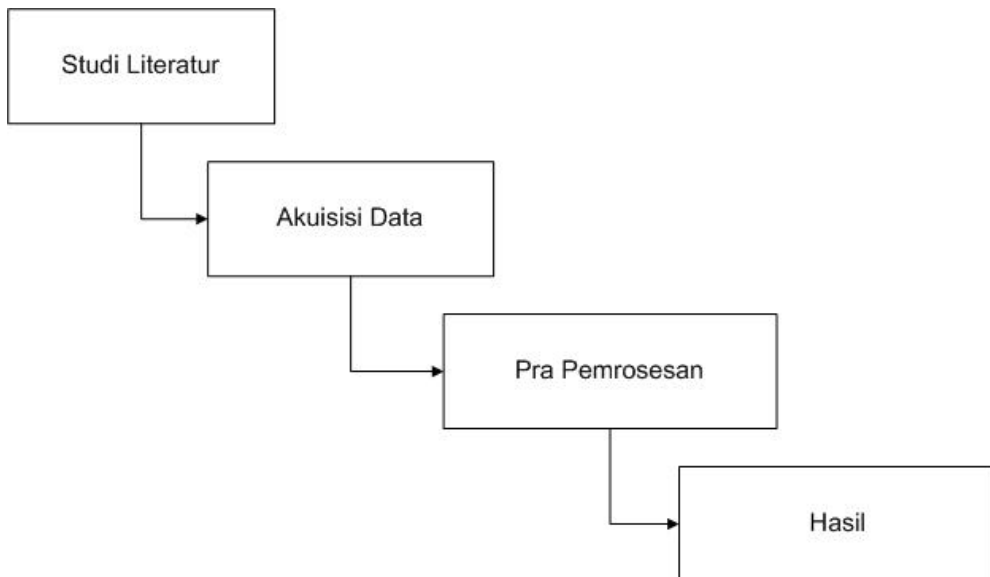
1. Tersedia Buku Python dan buku pemrograman lainnya.
2. Tersedia Wifi untuk mendukung pembelajaran yang dilakukan di IRC.
3. Ruangan yang bersih dan tertata rapih.
4. Jam masuk 09.00 sesuai jadwal yang ditentukan.
5. Pembelajaran khusus dari rekan-rekan IRC.

7.2 Faktor Penghambat

1. Jika terlalu ramai membuat ruangan sedikit tidak kondusif.
2. Belum menguasai pemrograman python.
3. Seringnya terjadi mati listrik apabila semua peralatan dinyalakan.

METODOLOGI PENELITIAN

Diagram Alur Metodologi Penelitian



1. Deskripsi Metode

Tahapan alur yang digunakan peneliti dalam analisis data EEG adalah sebagai berikut:

1.1 Studi Literatur

Studi Literatur merupakan tahap awal persiapan untuk mempersiapkan kegiatan penelitian. Persiapan yang dilakukan berupa persiapan judul, proposal, surat perizinan dan pengumpulan data. Tahap ini dilakukan studi pustaka atau mencari referensi beberapa buku jurnal-jurnal, informasi-informasi, serta laporan-laporan sebagai pendukung kegiatan penelitian EEG.

1.2 Akuisisi Data

Akuisisi Data merupakan tahap yang digunakan untuk memperoleh data yang nantinya data tersebut digunakan pada pemrosesan sistem. Pengambilan data dilakukan pada subjek dengan menggunakan alat perekam EEG. Dalam penelitian ini, metode yang digunakan pengumpulan data dilakukan pada subjek yang berusia sekitar 22-24 tahun berjumlah 7 orang. Semua subjek akan direkam gelombang otaknya menggunakan Neurosky Mindwave untuk mendapatkan parameter yang akan diproses oleh sistem nantinya. \cite{simbolon2015experiment}.

Proses perekaman dilakukan sebanyak 10 sesi pada setiap subjek yang berjumlah 7 orang. Setiap subjek direkam sebanyak 10 kali ketika bermain easy dan bermain hard. Proses perekaman sinyal EEG ini merupakan tahap untuk mengumpulkan data EEG yang bertujuan untuk menganalisa sinyal otak. Proses perekaman sinyal EEG dilakukan dengan cara memasang alat Neurosky Mindwave dibagian kepala subjek. Perekaman sinyal EEG menggunakan komputer yang telah terpasang aplikasi python dan terkoneksi dengan port USB Neurosky Mindwave.

Eksperimen dilakukan di sebuah ruangan yang sudah dikondisikan untuk bermain game crash team racing.

Setelah alat terpasang, pastikan pengaturan \textit{port} USB nya sesuai, keadaan baterainya juga harus memadai dan posisi operator yang merekam data melalui laptop juga tidak boleh terlalu jauh melebihi sekitar 20 m dari subjek yang telah dipasang alat \textit{Neurosky Mindwave}. Selain itu pastikan juga tidak ada benda yang menghalangi di area tersebut, agar pancaran sinyal dari USB \textit{Neurosky Mindwave} dapat terdeteksi oleh alat EEG \textit{Neurosky Mindwave} dengan baik.

1.3 Pra Pemrosesan

Pra-pemrosesan sinyal ini dilakukan setelah mendapatkan sinyal mentah dari percobaan perekaman sinyal EEG. Data sampel melalui perekaman menggunakan program Python otomatis tersimpan dengan format file *.csv. Data tersebut kemudian diolah. Setelah perekaman selesai, hasil rekaman EEG dapat diproses ke tahap berikutnya. Format csv (comma separated values) merupakan suatu format data dalam basis data dimana setiap record dipisahkan dengan tanda koma (,) atau titik koma (;). Format csv dapat dibuka dengan berbagai text-editor seperti Notepad, Wordpad, dan ms.excel .

1.4 Pemrosesan Data

Pada tahap ini data RAW hasil perekaman menggunakan Neurosky Mindwave akan diolah. Data yang akan diolah adalah data hasil perekaman sinyal otak yang berbentuk format .csv. Gambar dibawah ini merupakan data .csv subjek yang telah direkam menggunakan alat EEG Neurosky.

gambar 1

gambar 2

1.5 Hasil

Pada tahap ini merupakan hasil dari studi literatur hingga pengolahan data dan disusun menjadi sebuah laporan. Dari hasil pengolahan sinyal EEG, dapat kita lihat perbedaan kondisi aktivitas otak pemain game crash team racing.

KESIMPULAN

1. Kesimpulan Masalah

Berdasarkan hasil analisis dari penelitian ini, ekstraksi sinyal hasil rekaman alat Neurosky Mindwave pada pemain game crash team racing mampu diidentifikasi dengan baik berdasarkan gelombang otak yang terekam oleh alat.

2. Kesimpulan metode

Berdasarkan penelitian, pengambilan data sampai ke analisis untuk melakukan validasi apakah atribut yang dipilih bisa dilakukan klasifikasi atau tidak dengan decision tree.

3. Kesimpulan Penelitian

Analisis sinyal EEG pada pemain game crash team racing, berdasarkan hasil analisis kondisi gelombang otak subjek saat bermain game crash team racing mampu dilihat karakteristiknya. Akan terlihat bagaimana orang berpikir ketika mereka bermain game crash team racing. Berdasarkan hasil analisis penelitian ini, hasil dari pada pemain game crash team racing lebih fokus pada game dengan tingkat kesulitan easy dibandingkan dengan tingkat kesulitan hard.

