

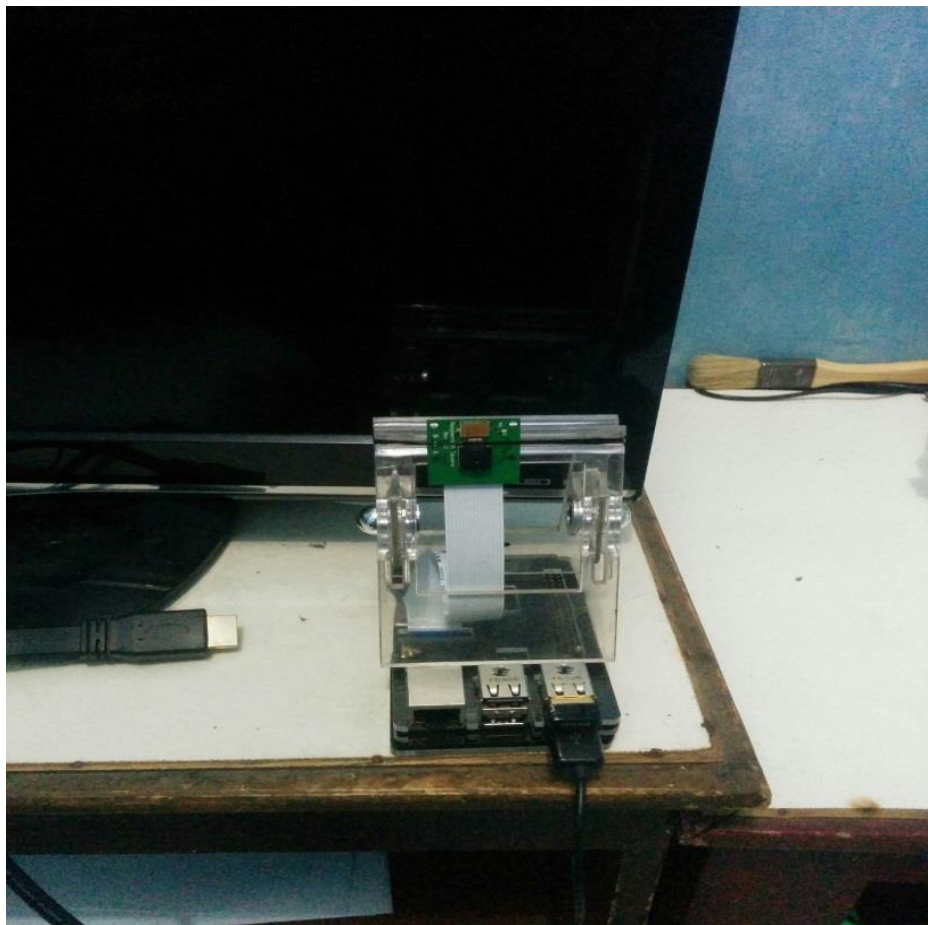
BAB IV

IMPLEMENTASI DAN PEMBAHASAN

4.1 Implementasi

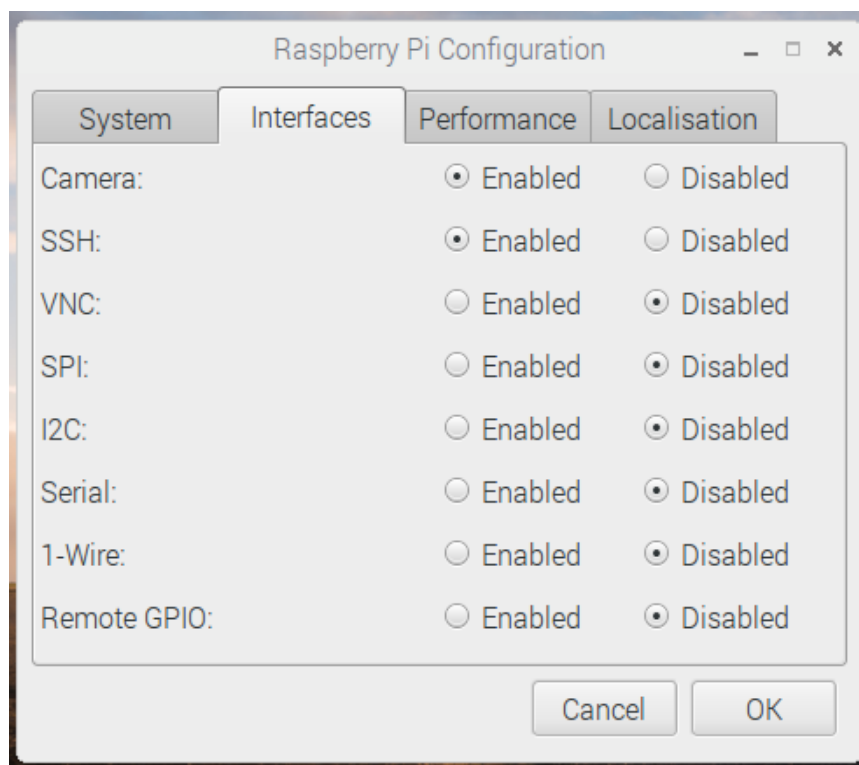
4.1.1 Konfigurasi Modul Kamera

Sebelum melakukan konfigurasi modul kamera pada Raspberry Pi hal pertama yang harus dilakukan adalah menghubungkan modul kamera ke port kamera pada yang tersedia pada Raspberry Pi.



Gambar 4.1 Raspberry Pi Dan Modul Kamera

Konfigurasi modul kamera pada Raspberry Pi agar dapat terbaca oleh system dengan cara pilih menu Preferences lalu pilih Raspberry Pi Configuration lalu pada tab interfaces klik enable untuk mengaktifkan modul kamera pada Raspberry Pi lalu klik OK. Untuk mengetahui apakah modul kamera telah aktif dapat dilakukan dengan mengetik perintah atau command pada terminal. `Raspistill -o gambar.jpg`, perintah untuk mengakses modul kamera untuk mengambil gambar dengan format .jpg. untuk merekam video dapat dilakukan dengan mengetik perintah `raspivid -o video.h264`, maka modul kamera pada raspberry pi akan merekam video dengan format .h264.



Gambar 4.2 Konfigurasi Modul Kamera

4.1.2 Instalasi OpenCV

Untuk instalasi OpenCV pada system operasi Raspbian pada Raspberry Pi langkah pertama yang dilakukan adalah mengunduh file OpenCV pada situs resmi www.opencv.org, lalu pilih versi yang dari OpenCV dan pilih platform dari system operasi. Setelah file berhasil diunduh lalu ekstrak file tersebut dan masuk ke direktori melalui terminal dimana tempat menyimpan file OpenCV disimpan.

```
cd ~/opencv
$ mkdir build
$ cd build
$ cmake -D CMAKE_BUILD_TYPE=RELEASE \
-D CMAKE_INSTALL_PREFIX=/usr/local \
-D INSTALL_C_EXAMPLES=ON \
-D INSTALL_PYTHON_EXAMPLES=ON \
-D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib/modules \
-D BUILD_EXAMPLES=ON ..
$ make -j4
$ sudo make install
$ sudo ldconfig
```

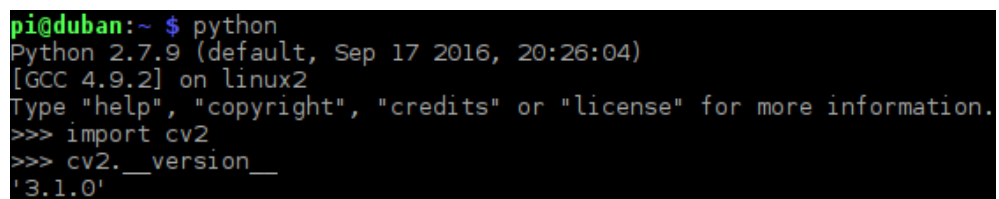
Gambar 4.3 Command Build OpenCV

Untuk memastikan bahwa OpenCV telah diinstall dengan benar maka dapat dilakukan verifikasi dengan mengetik beberapa kode program python pada python console melalui terminal, kode ini berfungsi untuk mengetahui versi dari OpenCV yang telah terintegrasi pada system operasi. Perintah atau kode program python tersebut adalah :

```
$ python
>>> import cv2
```

```
>>> cv2.__version__
```

Jika OpenCV telah diinstall dengan benar maka python console akan menghasilkan output berupa versi dari OpenCV yang telah terinstall pada system. Output dari kode diatas dalam dilihat pada gambar berikut :



```
pi@duban:~ $ python
Python 2.7.9 (default, Sep 17 2016, 20:26:04)
[GCC 4.9.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'3.1.0'
```

Gambar 4.4 Output Versi OpenCV

4.1.3 Instalasi FFMPEG

FFmpeg adalah program komputer yang dapat merekam, mengkonversikan dan streaming audio dan video digital dalam berbagai format. FFMpeg merupakan aplikasi command line yang terdiri dari kumpulan pustaka perangkat lunak bebas / open source. Langkah pertama yang dilakukan adalah mengunduh file ffmpeg pada situs resmi di <http://ffmpeg.org>. ekstrak file tersebut lalu masuk ke direktori melalui terminal dengan menambahkan perintah-perintah berikut :

```
wget http://ffmpeg.org/releases/ffmpeg-2.7.2.tar.bz2
cd ..
mkdir src
cd src
tar xvjf ../software/ffmpeg-2.7.2.tar.bz2
./configure --enable-gpl --enable-postproc --enable-
swscale --enable-avfilter --enable-libmp3lame --enable-
libvorbis --enable-libtheora --enable-libx264 --enable-
libspeex --enable-shared --enable-pthreads --enable-
libopenjpeg --enable-libfaac --enable-nonfree
Make -j4
sudo make install
```

Gambar 4.5 Build FFmpeg

Untuk mengetahui apakah ffmpeg telah terinstal dengan benar pada system dapat dilakukan dengan mengetik perintah `ffmpeg -version` pada terminal console. Jika ffmpeg telah diinstall dengan benar dan tidak terjadi kesalahan maka system akan menampilkan output berupa detail versi dari ffmpeg yang terinstall pada system.

```
pi@duban:~ $ ffmpeg -version
ffmpeg version 3.0.2 Copyright (c) 2000-2016 the FFmpeg developers
built with gcc 4.9.2 (Raspbian 4.9.2-10)
configuration: --enable-gpl --enable-postproc --enable-swscale --enable-avfilter
--enable-libmp3lame --enable-libvorbis --enable-libtheora --enable-libx264 --en
able-libspeex --enable-shared --enable-pthreads --enable-libopenjpeg --enable-li
bfaac --enable-nonfree
libavutil      55. 17.103 / 55. 17.103
libavcodec     57. 24.102 / 57. 24.102
libavformat    57. 25.100 / 57. 25.100
libavdevice    57.  0.101 / 57.  0.101
libavfilter     6. 31.100 /  6. 31.100
libswscale      4.  0.100 /  4.  0.100
libswresample  2.  0.101 /  2.  0.101
libpostproc   54.  0.100 / 54.  0.100
```

Gambar 4.6 Output Versi FFmpeg

4.1.4 Membuat Aplikasi pada Platform Dropbox



Saat Anda mulai membuat aplikasi di Platform Dropbox, Anda harus mendaftarkan aplikasi anda pada Dropbox di <https://www.dropbox.com/developers/apps> . Sebagai bagian dari proses, Anda harus memilih izin yang tepat untuk aplikasi Anda. Izin aplikasi Anda (terkadang disebut sebagai jenis akses dalam dokumentasi) menentukan data yang dapat diakses aplikasi Anda di Dropbox pengguna.

Folder aplikasi, Folder khusus yang dinamai sesuai aplikasi Anda dibuat dalam folder Apps Dropbox pengguna. Aplikasi Anda hanya bisa membaca dan menulis akses ke folder ini dan pengguna dapat memberikan konten ke aplikasi Anda dengan memindahkan file ke dalam folder ini.

Chooser atau Saver, Buat aplikasi App atau aplikasi Full Dropbox untuk membuat Chooser atau Saver Anda aktif dan berjalan. Sebenarnya, Chooser dan Saver tidak membutuhkan izin. Bila Anda menggunakan Chooser atau Saver, aplikasi Anda hanya mendapatkan akses khusus yang diberikan oleh pengguna.

Create a new app on the Dropbox Platform

1. Choose an API

<input checked="" type="radio"/> Dropbox API For apps that need to access files in Dropbox. Learn more		<input type="radio"/> Dropbox Business API For apps that need access to Dropbox Business team info. Learn more	
--	---	--	---

2. Choose the type of access you need

[Learn more about access types](#)

<input checked="" type="radio"/> App folder – Access to a single folder created specifically for your app.
<input type="radio"/> Full Dropbox – Access to all files and folders in a user's Dropbox.

3. Name your app

Create app

Gambar 4.7 Membuat Aplikasi Pada Dropbox Platform

Setelah aplikasi berhasil dibuat maka akan mendapat kode App key, App secret, dan kode token yang nantinya digunakan untuk menghubungkan antara aplikasi dengan Dropbox.

PiCam_Video

Settings	Branding	Analytics
----------	----------	-----------

Status	Development	Apply for production
--------	-------------	----------------------

Development users	Only you	Enable additional users
-------------------	----------	-------------------------

Permission type	App folder ⓘ
-----------------	--------------

App folder name	PiCam_Video	Change
-----------------	-------------	--------

App key	3fwp09zf95ive5a
App secret	bsd5oo0s659wp8v

Gambar 4.8 App key dan App secret Dropbox API

OAuth 2	Redirect URIs
---------	----------------------

https:// (http allowed for localhost)	Add
---------------------------------------	-----

Allow implicit grant ⓘ

Allow ▼

Generated access token ⓘ

0E3ESNkrinAAAAAAAAABRVzK-SaHpwvqvY03qJoxX27l8mM4mbtpj8lWLW2W3w9S

This access token can be used to access your account (duban.mblo@gmail.com) via the API. Don't share your access token with anyone.

Chooser/Saver domains	example.com	Add
-----------------------	-------------	-----

If using the [Chooser](#) or the [Saver](#) on a website, the domain of that site.

Webhooks	Webhook URIs ⓘ
----------	-----------------------

https://	Add
----------	-----

Gambar 4.9 Kode Token Dropbox API

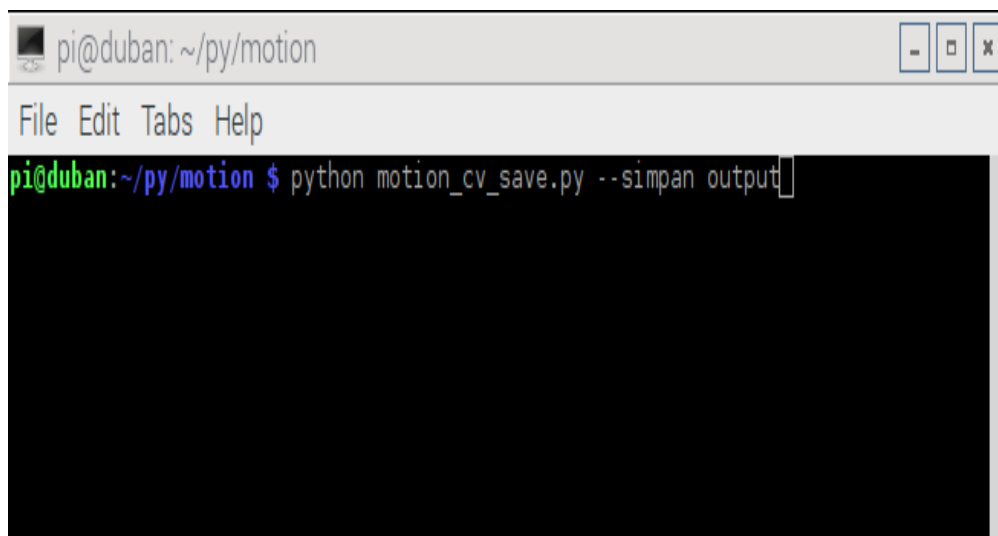
4.1.5 Kode Program

```
for frame in camera.capture_continuous(rawCapture,
format="bgr", use_video_port=True):
    frame = frame.array
    ts = dt.datetime.now().strftime('%d%m%Y_%H%M%S')
    text = "No Object Detected!"
    frame = imutils.resize(frame, width=400)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    gray = cv2.GaussianBlur(gray, (5,5), 0)
    if avg is None:
        print '[INFO] starting background
model...'
        avg = gray.copy().astype("float")
        rawCapture.truncate(0)
        continue
    cv2.accumulateWeighted(gray, avg, 0.6)
    frameDelta = cv2.absdiff(gray,
cv2.convertScaleAbs(avg))
    thresh = cv2.threshold(gray, 25, 255,
        cv2.THRESH_BINARY_INV)[1]
    (gray,cnts, _) = cv2.findContours(thresh.copy(),
cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
cv2.imshow("Frame", frame)
cv2.imshow("Frame Delta", frameDelta)
cv2.imshow("Frame Thresh", thresh)
```

Gambar 4.10 Kode Untuk Menampilkan Frame

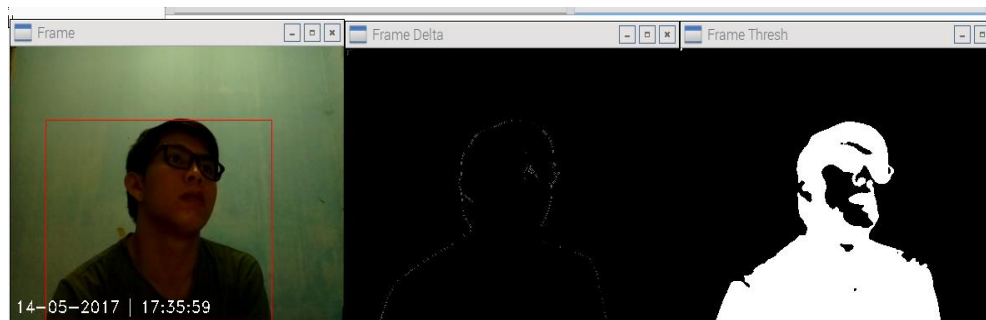
Cuplikan kode program memanfaatkan modul atau pustaka dari OpenCV yang berfungsi untuk menampilkan frame video. Teknik pengambilan video menggunakan perulangan dan disimpan pada variable array dan menggunakan format warna standar bgr agar tekstur warna dari video dalam tampil sesuai dengan aslinya. Pada tiga baris terakhir dari program berfungsi untuk menampilkan frame dari video.

Frame video yang ditampilkan sebanyak tiga frame yang masing-masing frame memiliki teknik yang berbeda dalam menampilkan frame video. Frame pertama adalah frame normal, frame kedua menggunakan teknik *edge detection* dengan memanfaatkan modul yang telah disediakan oleh OpenCV, lalu frame ketiga menggunakan teknik *Threshold* atau teknik yang berbasis pada image processing dimana melakukan pengolahan pixel pada suatu citra atau menghilangkan beberapa pixel dan juga mempertahankan beberapa pixel sehingga menghasilkan suatu citra baru hasil sortir pixel yang telah dilakukan.

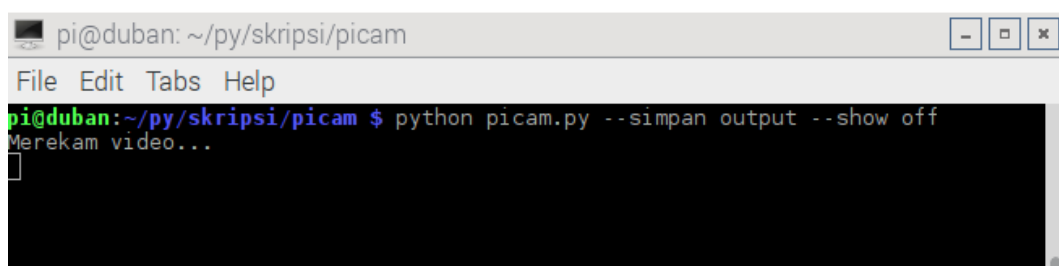


```
pi@duban: ~/py/motion
File Edit Tabs Help
pi@duban:~/py/motion $ python motion_cv_save.py --simpan output
```

Gambar 4.11 Command Untuk Menjalankan File Python



Gambar 4.12 Output Frame



Gambar 4.13 Menjalankan Program Tanpa Menampilkan Frame

```

for c in cnts:
    if cv2.contourArea(c) < args["min_area"]:
        continue
    (x, y, w, h) = cv2.boundingRect(c)
    cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 0,
255), 1)
    text = "Object Detected"
    if text == "Object Detected":
        if (timestamp - lastUploaded).seconds >=
min_upload_seconds:
            motionCounter += 1
            if motionCounter >= min_motion_frames:
                t = TempImage()
                n = Notif()
cv2.imwrite(t.path, frame)
filename = dt.datetime.now().strftime('%d%m%Y_%H%M%S')
timestamp = dt.datetime.now()
print '[UPLOAD] {}'.format(ts)
path = "{base_path}/{filename}.jpg".format(
base_path=args["simpan"], filename=ts)
client.put_file(path, open(t.path, "rb"))
t.cleanup()
n.kirim()
    else:
        motionCounter = 0

```

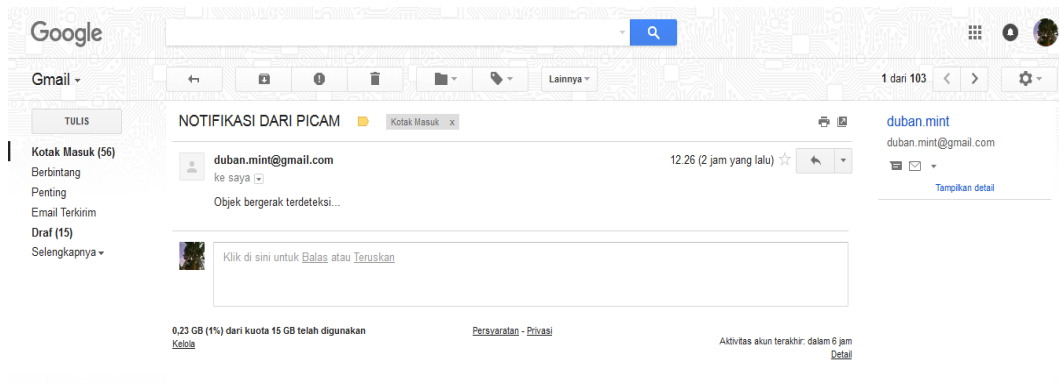
Gambar 4.14 Fungsi untuk Deteksi Gerak

Penjelasan :

Kode program yang berfungsi untuk mendeteksi gerakan suatu objek dengan memanfaatkan pustaka dari OpenCV yang dapat menampilkan suatu kontur tertentu yang baik ukuran panjang dan lebarnya mengikuti ukuran objek yang terekam oleh kamera serta kontur tersebut secara otomatis mengikuti gerakan dari objek atau dikenal dengan istilah *motion detection* atau deteksi gerakan. Jika kamera mendeteksi adanya gerakan maka system secara otomatis akan mengambil gambar dari frame video dan menyimpan pada temporer direktori lalu mengirim gambar ke akun dropbox yang telah terintegrasi dengan system kemudian system secara otomatis menghapus file gambar tersebut.

```
import smtplib
from email.MIMEMultipart import MIMEMultipart
from email.MIMEText import MIMEText
class Notif:
    def kirim(self):
        fromaddr = "duban.mint@gmail.com"
        toaddr = "mikael.duban@gmail.com"
        msg = MIMEMultipart()
        msg['From'] = fromaddr
        msg['To'] = toaddr
        msg['Subject'] = "NOTIFIKASI DARI PICAM"
        body = "Objek bergerak terdeteksi...!!!"
        msg.attach(MIMEText(body, 'plain'))
        server = smtplib.SMTP('smtp.gmail.com', 587)
        server.starttls()
        server.login(fromaddr, "mika199126")
        text = msg.as_string()
        server.sendmail(fromaddr, toaddr, text)
        server.quit()
```

Gambar 4.15 Fungsi Mengirim Notifikasi Email



Gambar 4.16 Output Notifikasi Email

Penjelasan :

Modul `smtplib` mendefinisikan objek sesi klien SMTP yang dapat digunakan untuk mengirim email ke mesin Internet manapun dengan daemon pendengar SMTP atau ESMTP. Modul `smtplib` berguna untuk berkomunikasi dengan mail server untuk mengirim email. Untuk mengirim email, kita harus mengubah objek menjadi string, dan kemudian menggunakan prosedur yang sama seperti di atas untuk mengirim menggunakan server SMTP.

Protokol SMTP menyertakan perintah untuk meminta server apakah sebuah alamat valid. Biasanya `VRFY` dinonaktifkan untuk mencegah spammer menemukan alamat email yang sah, namun jika diaktifkan, Anda dapat meminta server tentang alamat dan menerima kode status yang menunjukkan validitas beserta nama lengkap pengguna.

```

import uuid
import os
class TempImage:
    def __init__(self, basePath="./gambar", ext=".jpg"):
        app_key = '3fwpo9zf95ive5a'
        app_secret = 'bsd5oo0s659wp8v'
        flow =
        dropbox.client.DropboxOAuth2FlowNoRedirect(app
        _key, app_secret)
        authorize_url = flow.start()
        print '1. Kunjungi : ' + authorize_url
        print '2. Klik "Allow" (Anda mungkin harus
        login)'
        print '3. Copy kode.'
        code = raw_input("Masukan kode disini :
        ").strip()
        access_token, user_id = flow.finish(code)
        client =
        dropbox.client.DropboxClient('0E3ESNkrinAAAAAA
        AAAAH-
        fSw7tGWCKxheHShpDk3ukVa2o2drV1C_LDNhux0OBu')
        print 'Dropbox berhasil terkoneksi!'

        self.path =
        "{base_path}/{rand}{ext}".format(base_path=bas
        ePath,
            rand=str(uuid.uuid4()), ext=ext)
    def cleanup(self):
        os.remove(self.path)

```

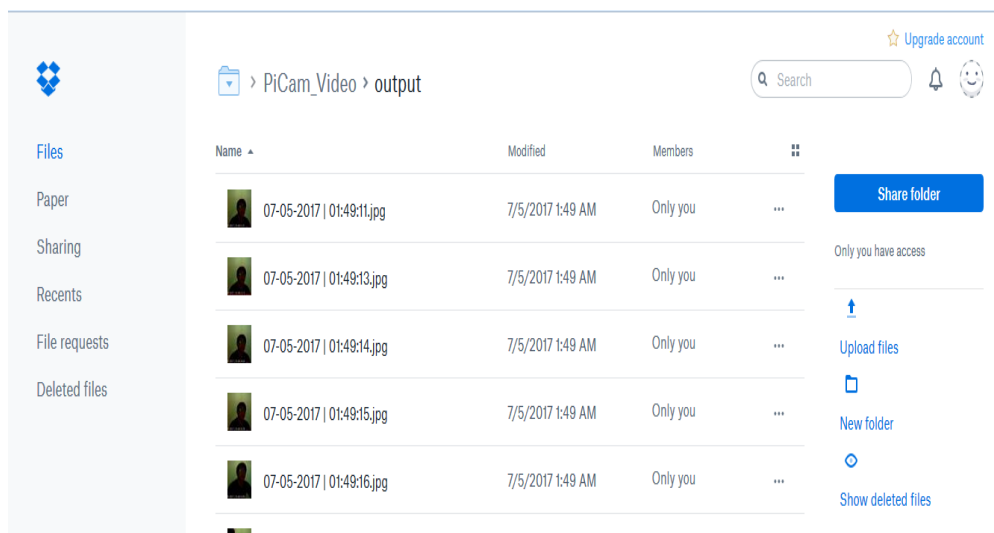
Gambar 4.17 Fungsi Mengirim Gambar Ke Dropbox

Penjelasan :

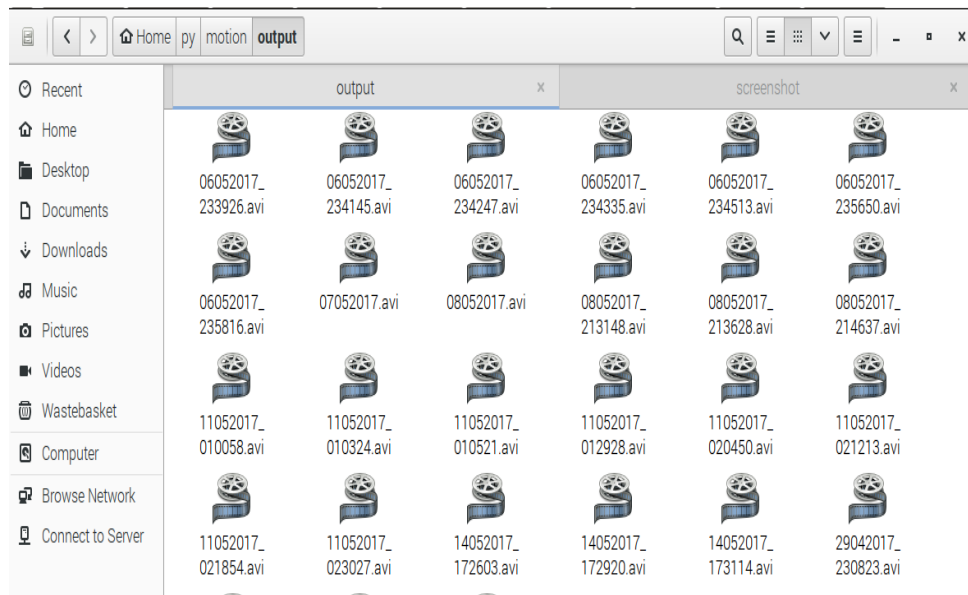
Saat membuat API memanggil API Dropbox, setiap permintaan memerlukan tingkat otentikasi tertentu. Jika Anda menggunakan SDK Dropbox resmi, ini akan menangani masalah spesifik ini untuk Anda. App Authentication, Ini adalah jenis otentikasi yang paling umum. Jenis ini menggunakan token akses untuk pengguna dan pasangan aplikasi tertentu, agar

dapat beroperasi di akun pengguna tersebut, sejauh diizinkan oleh izin aplikasi itu.

Jika system mendeteksi gerakan maka kemudian akan menulis gambar dari frame dan menyimpan pada penyimpanan internal Raspberry Pi menggunakan kelas TempImage , mengunggahnya melalui API Dropbox, dan kemudian jika file gambar berhasil diunggah ke Dropbox maka file gambar akan dihapus menggunakan fungsi cleanup pada kelas TempImage.



Gambar 4.18 Output Gambar Pada Dropbox



Gambar 4.19 Output File video di Internal Raspberry Pi

4.2 Pembahasan Sistem

Pada tahapan pembahasan system ini akan di tampilkan hasil pengujian dari sistem terhadap fungsi-fungsi yang digunakan pada system untuk mengetahui apakah fitur-fitur telah berjalan sebagaimana mestinya dan untuk memastikan tidak ada kesalahan baik dari sisi interface atau tampilan maupun dari sisi logika dan algoritma. Pada tahap pengujian dilakukan menggunakan metode Black Box dan White Box.

Tabel 4.1 Tabel Pengujian

No	Pengujian	Hasil yang diharapkan	Hasil Pengujian	Keterangan
1	Perintah atau command untuk eksekusi file python	Nama dan format file video secara default mengikuti waktu video direkam	Sesuai	
2	Deteksi gerakan	Kontur mengikuti pergerakan objek,	Sesuai	Menyesuaikan dengan kecepatan dari pergerakan objek
3	Mengirim notifikasi via email	Jika ada objek bergerak maka sistem mengirim notifikasi via email	Sesuai	
4	Simpan file gambar secara temporer	Mengambil gambar secara otomatis lalu menghapus setelah gambar diupload ke Dropbox	Sesuai	
5	Upload file gambar ke dropbox	Mengupload file gambar setelah gambar diambil	Sesuai	