

J-INTTECH

Journal of Information and Technology

Volume 06 Nomor 01, Bulan Juni Tahun 2018



STIKI

SEKOLAH TINGGI INFORMATIKA & KOMPUTER INDONESIA

Jl. Raya Tidar 100 Malang, 65146

Telp. (0341)560823, Fax (0341)562525

ISSN: 2303-1425 E-ISSN: 2580-720X

J-INTTECH

Journal of Information and Technology
Volume 06 Nomor 01, Bulan Juni 2018



LEMBAGA PENELITIAN DAN PENGABDIAN KEPADA MASYARAKAT

STIKI

SEKOLAH TINGGI INFORMATIKA & KOMPUTER INDONESIA
Jl. Raya Tidar 100, Malang; Phone: 0341-560823; Fax: 0341-562525; <http://www.stiki.ac.id>; mail@stiki.ac.id

PENGANTAR REDAKSI

J-INTECH merupakan jurnal yang diterbitkan oleh Sekolah Tinggi Informatika dan Komputer Indonesia Malang guna mengakomodasi kebutuhan akan perkembangan Teknologi Informasi serta guna mensukseskan salah satu program DIKTI yang mewajibkan seluruh Perguruan Tinggi untuk menerbitkan dan mengunggah karya ilmiah mahasiswanya dalam bentuk terbitan maupun jurnal online.

Pada edisi ini, redaksi menampilkan beberapa karya ilmiah mahasiswa yang mewakili beberapa mahasiswa yang lain, yang dianggap cukup baik sebagai media pembelajaran bagi para lulusan selanjutnya.

Tentu saja diharapkan pada setiap penerbitan memiliki nilai lebih dari karya ilmiah yang dihasilkan sebelumnya sehingga merupakan nilai tambah bagi para adik kelas maupun pihak-pihak yang ingin studi atau memanfaatkan karya tersebut selanjutnya.

Pada kesempatan ini kami juga mengundang pihak-pihak dari PTN/PTS lain sebagai kontributor karya ilmiah terhadap jurnal J-INTECH, sehingga Perkembangan IPTEK dapat dikuasai secara bersama-sama dan membawa manfaat bagi institusi masing-masing.

Akhir redaksi berharap semoga dengan terbitnya jurnal ini membawa manfaat bagi para mahasiswa, dosen pembimbing, pihak yang bekerja pada bidang Teknologi Informasi serta untuk perkembangan IPTEK di masa depan.

REDAKSI

J-INTECH

Journal of Information and Technology
Volume 06 Nomor 01, Bulan Juni 2018

DAFTAR ISI

Sistem Informasi Pelayanan Terpadu di Restoran Berbasis Android <i>Hery Kuswandi</i>	01-08
Pemanfaatan <i>Raspberry Pi</i> Dan Webcam Sebagai Kamera Pemantau Dan <i>Cloud Drive</i> Sebagai Media Penyimpanan <i>Ady Noegroho</i>	09-17
Sistem Penunjang Keputusan Berbasis <i>Webgis</i> Dengan Metode AHP Untuk Pemilihan Lokasi Usaha..... <i>Sya'roni</i>	18-22
Aplikasi Manajemen <i>Inventory</i> Berbasis <i>Mobile</i> <i>Angga Eka Syaputra</i>	23-32
Aplikasi Perencanaan Kebutuhan Produksi Menggunakan <i>Demand</i> <i>Forecasting</i> Dengan Pendekatan Proyektif..... <i>Samuel Pusirumang Makahanap</i>	33-42
Membangun Aplikasi <i>E-Commerce</i> Dengan Sistem Penunjang Keputusan Metode Apriori Untuk Memberikan Rekomendasi Kepada Calon Pembeli Di Toko Islam Malang <i>Alamsyah Ady Nugroho</i>	43-47
Sistem Informasi <i>Inventory</i> pada UD. MM GoDAM “NENENG” Berbasis Web Guna Memudahkan Pengolahan Data Barang..... <i>Widia Normalasari</i>	48-52
Aplikasi Pembelajaran Menulis Permulaan Berbasis Android Menggunakan <i>Unity 2D</i> <i>Andi Fiqqih Adiqro</i>	53-62
Sistem Pakar Identifikasi Penyakit Burung Puyuh Menggunakan Metode <i>Inferensi Forward Chaining</i> Berbasis Android <i>Mahartin Hendra Sukmawan</i>	63-77

Sistem Keamanan <i>Database</i> Berbasis <i>Restfull</i> Pada <i>Content Management System Wordpress</i> (Studi Kasus : STIKI Malang).....	78-89
<i>Ridho Valentin</i>	
Sistem Informasi Pengolahan Data Surat Masuk dan Keluar di Kantor BARENLITBANG Kota Malang.....	90-93
<i>Antonius Lorensius</i>	
Sistem Penunjang Keputusan Pemilihan Jurusan Perguruan Tinggi Menggunakan Teori Psikologi <i>Rothwell Miller Interest Blank</i> (RMIB)	94-104
<i>Muhammad Hanifudin</i>	
Permainan Ular Tangga Berbasis Android Menggunakan <i>Unity</i>	105-118
<i>Novanda Bayhakky</i>	
Sistem Informasi Manajemen Pakan Guna Meningkatkan Indikator Keberhasilan Panen Ternak pada PT Berkah Benua Farm	119-140
<i>Burhannudin</i>	
Klasifikasi Artikel Berbahasa Indonesia untuk Mendeteksi <i>Clickbait</i> Menggunakan Metode Naïve Bayes	141-147
<i>Ali Fahnnur Yavi</i>	
Sistem Informasi Akademik SMK Bhakti Luhur Malang Berbasis Web	148-152
<i>Fransiskus Sina Witi</i>	
Pencarian Resep Masakan Menggunakan Metode <i>Vector Space Model</i> (VSM) Berbasis Android	153-160
<i>Bulan Dewi Gulita</i>	
Pemanfaatan Sensor Gyroscope pada Game Casual Berbasis Android.....	161-165
<i>Dionisius Aditya Remy Susanto</i>	
Penerapan Teknologi Augmented Reality pada <i>Game</i> Pengenalan Hewan Berdasarkan Jenis Makanannya Berbasis <i>Mobile</i>	166-172
<i>Herjuno Daud Pramono</i>	
Sistem Penunjang Keputusan Pemilihan Rumah Kontrakan untuk Keluarga di Kota Malang Menggunakan Metode Fuzzy Sugeno.....	173-176
<i>Slamet Nur Huda</i>	

ISSN: 2303-1425 E-ISSN: 2580-720X

J-INTTECH

Journal of Information and Technology
Volume 06 Nomor 01, Bulan Juni 2018

- Pelindung** : Ketua STIKI
- Penasehat** : Puket I, II, III
- Pembina** : Ka. LPPM
-
- Editor** : Subari, S.Kom, M.Kom
- Section Editor** : Daniel Rudiaman S.,ST, M.Kom
-
- Reviewer** : Dr. Eva Handriyantini, S.Kom, M.MT.
Evi Poerbaningtyas, S.Si, M.T.
Laila Isyriyah, S.Kom, M.Kom
Anita, S.Kom, M.T.
-
- Layout Editor** : Siti Aminah, S.Si, M.Pd
Nira Radita, S.Pd., M.Pd
Muh. Bima Indra Kusuma

PEMANFAATAN RASPBERRY PI DAN WEBCAM SEBAGAI KAMERA PEMANTAU DAN CLOUD DRIVE SEBAGAI MEDIA PENYIMPANAN

Ady Noegroho

Program Studi Teknik Informatika, Sekolah Tinggi Informatika Dan Komputer Indonesia
11170016@mhs.stiki.ac.id

ABSTRAK

Sistem yang dirancang dan dibangun dalam penelitian ini adalah sistem kamera pemantau menggunakan Raspberry Pi dengan menggunakan webcam sebagai alat perekamnya dan wifi dongle atau modem sebagai alat untuk mengirim data hasil rekaman melalui internet untuk disimpan di penyimpanan awan atau cloud storage. Sistem ini juga dirancang untuk dapat membedakan kondisi ruangan atau area yang dipantau ketika ada perubahan pergerakan di dalamnya dan akan langsung merekam perubahan pergerakan tersebut, jadi tidak semua yang ditangkap oleh kamera akan direkam. Bisa dikatakan bahwa sistem ini memiliki artificial intelligence untuk membedakan mana yang perlu dan tidak untuk direkam dan disimpan. Sehingga dengan memiliki kemampuan tersebut tidak akan memenuhi ruangan penyimpanan lokal yang memang sudah terbatas. Selain itu hasil rekaman gambar akan langsung diupload ke penyimpanan awan dan memberikan nama file sesuai dengan tanggal dan waktu gambar tersebut direkam, jadi memudahkan pengguna untuk mencari kapan gambar terakhir yang diupload oleh sistem. Selain itu secara otomatis sistem akan menghapus file rekaman tersebut dari tempat penyimpanan lokal, sehingga ruang penyimpanan lokal tidak akan menjadi penuh dengan hasil rekaman gambar oleh sistem, bisa diibaratkan bahwa kondisi penyimpanan lokal tetap kapasitasnya ketika sistem melakukan pekerjaannya. Sistem ini juga dirancang untuk langsung dapat berjalan ketika terjadi reboot misalnya ketika saat pemadaman listrik dan ketika listrik menyala kembali. Sehingga pengguna tidak akan direpotkan untuk mensetting dari awal agar sistem berjalan seperti semula.

Kata Kunci: Raspberry Pi, Webcam, OpenC

1. PENDAHULUAN

Perkembangan teknologi khususnya teknologi informasi telah mengubah peradaban dunia. Terhitung pada empat dasawarsa terakhir dimulai dengan digunakannya transistor yang lebih kecil pada arsitektur *processor* komputer, maka dimulailah era dimana teknologi informatika menjadi sebuah teknologi yang menunjang bidang-bidang ilmu pengetahuan lain untuk memudahkan manusia dalam menjalani kehidupannya di berbagai aspek dan fungsi.

Ilmu komputer merupakan disiplin termuda dalam bidang science yang berbeda dari Fisika, Matematika dan *science* klasik lainnya yang berdasar dari Yunani kuno (G. Dodig-Crnkovic, 2002). Walaupun beberapa ilmuwan masih memperdebatkan apakah computer science adalah science (P.J. Denning, 2005). Bahkan sebuah bidang baru yang diberi nama *computing* diperkenalkan untuk menjembatani hal tersebut (D.E. Comer, 1989). Belajar ilmu komputer memiliki konsekuensi untuk belajar banyak konsep dan banyak bidang ilmu lainnya, mengintegrasikan teori dan praktek, abstraksi (umum) dan desain (khusus).

Dalam dunia digital, kamera pun juga mengalami perkembangan yang cukup pesat

sekarang ini, ditandai dengan munculnya berbagai macam kamera digital, selain untuk mengambil gambar diam yang sering disebut dengan foto, juga dapat merekam atau menangkap gambar yang bergerak dalam bentuk video. Webcam adalah salah satu jenis kamera digital yang mampu melakukan hal-hal tersebut. Dari sinilah muncul pemikiran untuk memanfaatkannya sebagai kamera yang berfungsi untuk mengawasi suatu ruangan perkantoran atau pertokoan. Teknologi seperti ini telah banyak terjual di pasaran dengan metode yang berbeda. Pembuatan sistem keamanan ini dibuat dengan memanfaatkan kamera digital, yakni webcam sebagai pengambil gambar, dimana webcam dapat dengan mudah berkomunikasi data melalui port *USB* dan mudah dicari di pasaran. Salah satu kelemahan dari webcam adalah kurang mampu mengambil obyek pada ruang gelap yang menyebabkan obyek yang ditangkap menjadi kurang jelas.

Adapun dalam rancang bangun ini juga memanfaatkan metode deteksi gerak pada teknik pengolahan citra digital (*image processing*) yang diolah dalam *library OpenCV* dan menggunakan bahasa *python*.

Ditambah dengan teknologi baru yaitu komputer mini sebesar kartu *ATM*, yang

dinamakan *Raspberry Pi* yang merupakan penyempurna dari purwarupa *mikrocontroller* menjadi *mikroprocessor*, namun tanpa menyertakan local *harddrive* di dalamnya, penulis mencoba untuk membuat suatu sistem kamera pemantau dengan menggunakan *webcam* dan *Raspberry Pi*, dimana hasil dari rekaman kamera tersebut langsung tersimpan di penyimpanan awan (*cloud storage*). Sehingga sewaktu-waktu diperlukan, data yang tersimpan dapat dilihat kembali.

2. ANALISA DAN PERANCANGAN

Diagram Alir Sistem

Tahap selanjutnya dalam penelitian ini adalah membuat rancangan sistem yang nantinya menjadi acuan dalam implementasi sistem. Untuk memperjelas gambaran proses kerja oleh sistem kamera pemantau ini dapat dilihat pada *flowchart* di bawah ini.



Gambar 1. *Flowchart* Sistem Kamera Pemantau

Penjelasan dari gambar *flowchart* di atas sebagai berikut. Sistem akan melakukan autentikasi pada *dropbox*. Jika autentikasi gagal, sistem akan berhenti. Jika autentikasi berhasil maka sistem akan menjalankan proses selanjutnya. Kamera mengambil gambar. Sistem menganalisa gambar yang diambil oleh kamera, dan menentukan status ruangan, apakah ada pergerakan atau pada kondisi sepi atau tidak ada pergerakan. Jika status ruangan sepi, sistem

kembali melakukan proses pengambilan gambar oleh kamera. Jika status ruangan ada gerakan, sistem menuju proses selanjutnya. Sistem akan menyimpan gambar ke dalam *folder temporary*. Sistem memberi nama *file* sesuai waktu ketika gambar diambil oleh kamera. Sistem mengupload *file* ke *dropbox*. Setelah file terupload, sistem akan menghapus file. Sistem kembali ke proses ketika gambar diambil oleh kamera. Sistem akan berhenti ketika ada *Interrupt*.

Perancangan Otomatisasi Sistem

Sebagai sistem kamera pemantau, dengan berbasis linux yang tidak semua orang bisa dan mengerti cara mengoperasikan, maka sistem ini dirancang untuk terus bekerja, dan berhenti apabila ada interupsi yang disengaja dengan menekan tombol keyboard pada perintah tertentu oleh pengguna. Namun seperti diketahui, karena sistem ini berjalan di *Raspberry Pi*, maka kemungkinan sistem untuk berhenti karena interupsi oleh pengguna dengan memakai papan keyboard sangat kecil kemungkinannya. Tetapi ada kemungkinan sistem akan berhenti akibat pasokan energi ke sistem berhenti, misalnya karena pemadaman listrik. Untuk itu perlu dipikirkan rancangan agar sistem akan bekerja ketika listrik menyala lagi tanpa harus melakukan perintah atau setting agar sistem kembali bekerja seperti sedia kala.

Berdasarkan hal tersebut, maka penulis memakai *crontab* untuk mengatasi permasalahan itu. *Crontab* adalah aplikasi daemon (berjalan dibalik layar) yang digunakan untuk menjalankan tugas yang dijadwalkan pada suatu waktu di sistem operasi linux. Setiap user di sistem yang memiliki file *crontab*, mengijinkan file tersebut untuk melakukan suatu aksi yang telah dispesifikasikan sesuai waktu yang telah ditentukan. *Crontab* biasa digunakan untuk membuat backup secara otomatis, sinkronisasi files, dan lain-lain.

Dengan menggunakan *crontab*, penjadwalan kerja otomatis dari system ketika sistem melakukan reboot bisa dibuat. Dan untuk menggunakan *crontab*, penulis melakukan instalasi aplikasi *webmin*, dimana selain mudah dalam penggunaannya dan dapat dilakukan tanpa harus tersambung dengan unit sistem

3. IMPLEMENTASI DAN PEMBAHASAN

Instalasi sistem operasi pada *Raspberry Pi*

Karena *Raspberry Pi* pada dasarnya adalah sebuah komputer mini, maka dibutuhkan system operasi untuk menjalankan perangkat tersebut. Namun karena bentuk dan kegunaan dari *Raspberry*

Pi berbeda dengan umumnya komputer yang ada di pasaran, maka cara instalasi sistem operasi yang akan dipakai oleh *Raspberry Pi* juga berbeda dengan komputer pada umumnya.

Hal-hal yang dilakukan untuk menginstall sistem operasi *Raspberry Pi* sebagai berikut. Unduh sistem operasi *Raspberry Pi*, dalam hal ini adalah Raspbian Wheezy di laman berikut, <https://www.raspberrypi.org/downloads/>. Masukkan SD Card ke laptop. Buka Win32 Disk Imager dan klik gambar folder untuk memilih Raspbian OS yang akan di install.



Gambar 2. Instalasi sistem operasi *Raspberry Pi*

Klik *Write*, kemudian tunggu sampai proses penulisan sistem operasi *Raspbian* pada *SD Card* selesai.

Sharing koneksi internet antara laptop dan *Raspberry Pi*

Menghubungkan antara laptop dengan *Raspberry Pi* bertujuan untuk memudahkan instalasi dan implementasi sistem selanjutnya. Dengan membuat koneksi antara laptop dengan *Raspberry Pi*, dikarenakan tidak adanya *display device* dan *hardware* lainnya seperti *keyboard* dan *mouse* pada *Raspberry Pi*, maka proses selanjutnya dapat dilakukan dengan cara *remote desktop* sistem operasi *Raspberry Pi* pada *windows*.

Dan umum diketahui untuk instalasi suatu aplikasi maupun *library* pada *linux* berbeda dengan *windows* yang *user-friendly*. Di *linux* untuk menginstall aplikasi maupun *library* salah satu caranya menggunakan *repository*.

Dan untuk menggunakan *repository* maka dibutuhkan koneksi internet, sedangkan sampai pada tahap ini *Raspberry Pi* belum terkoneksi dengan internet. Maka cara yang paling mudah agar *Raspberry Pi* terkoneksi dengan internet untuk mengunduh *repository* yang akan digunakan untuk proses sistem selanjutnya adalah *sharing* koneksi internet antara laptop dengan *Raspberry Pi* menggunakan jaringan yang telah dibuat yaitu LAN. Langkah-langkah untuk *sharing* koneksi internet laptop dan *Raspberry Pi* sebagai berikut:

1. IP Address antara laptop dan *Raspberry Pi* harus dalam satu jaringan yang sama. Cara mengubah *ip address Raspberry Pi* yaitu mengedit file *cmdline.txt* dengan notepad



Gambar 3. Langkah 1

2. *Sharing* koneksi internet di laptop dengan *ip address Raspberry Pi*



Gambar 4. Langkah 2

3. Buka *software Xming* dan *Putty*, atur konfigurasi *Putty* dan masukkan *ipaddress Raspberry Pi* tekan *Open* dan akan muncul *login terminal*. Pengaturan *default*-nya adalah *login* sebagai *pi*, *password*-nya adalah *raspberry*. Lalu pada *terminal putty* ketik *lxsession*, maka *desktop Raspbian* akan ditampilkan di *desktop windows*.





Gambar 5. Langkah 3

Instalasi OpenCV

Buka terminal dan ketikkan perintah berikut:

```
$ sudo apt-get update
• $ sudo apt-get upgrade
• $ sudo rpi-update
• $ sudo apt-get install build-essential cmake pkg-config
• $ sudo apt-get install libgtk2.0-dev
• $ sudo apt-get install libjpeg8-dev libtiff4-dev libjasper-dev libpng12-dev
• $ sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev
• $ sudo apt-get install libatlas-base-dev gfortran
• $ wget https://bootstrap.pypa.io/get-pip.py
• $ sudo python get-pip.py
• $ sudo pip install virtualenv virtualenvwrapper
• $ sudo rm -rf ~/.cache/pip
• $ sudo nano source ~/.profile Tambahkan tiga
```

baris di bawah ini dalam ~/.profile

```
o # virtualenv and virtualenvwrapper
o Export
WORKON_HOME=$HOME/.virtualenvs
o source /usr/local/bin/virtualenvwrapper.
s
h
simpan dan tutup
• $ mkvirtualenv cv
• $ sudo apt-get install python2.7-dev
• $ sudo pip install numpy
• $ sudo pip install imutils
• $ sudo apt-get install python-opencv
• $ sudo apt-get install libcv-dev
• $ sudo apt-get install libopencv-dev
• $ sudo apt-get install libcv2.3
• $ cd ~/.virtualenvs/cv/lib/python2.7/site-packages/
• $ ln -s /usr/local/lib/python2.7/site-packages/cv2.so cv2.so
• $ ln -s /usr/local/lib/python2.7/site-packages/cv.py cv.py
```

Instalasi Dropbox API di Raspberry Pi

Untuk bisa menggunakan Dropbox API, harus registrasi dulu di <https://www.dropbox.com/developers/apps> kemudian klik *create app* seperti gambar di bawah ini:



Gambar 6. Dropbox API

Kemudian akan menuju halaman yang harus diisi tentang aplikasi yang akan dibuat. Di halaman tersebut akan diperoleh nomor *app key* dan *app secret* yang akan diisikan untuk mendapatkan token.



Gambar 7. Konfigurasi Dropbox API

Instalasi *Webmin*

Dikarenakan *repository Raspberry Pi* tidak mempunyai paket untuk instalasi *webmin*, maka ada beberapa hal yang perlu dilakukan agar paket *webmin* dapat terinstall di *Raspberry Pi*.



Gambar 8. Edit *Repository Raspbian*

- Ketik `sudo nano /etc/apt/sources.list`
- Tambahkan untuk sumber *repository* dari :
 - o `deb http://download.webmin.com/download/repository sarge contrib`
 - o `deb http://webmin.mirror.somers-ettechsolutions.co.uk/repository sarge contrib`
- Kemudian di *terminal* ketikkan perintah
 - o `cd /root`
 - o `wget http://www.webmin.com/jcameron-key.asc`
 - o `apt-key add jcameron-key.asc`
 - o `apt-get update`
 - o `apt-get install webmin`
- *Webmin* telah terinstall, sekarang tinggal mengatur konfigurasinya



Gambar 9. *Webmin* telah terinstal



Gambar 10. Konfigurasi *Webmin*

Pilih *system* -> *schedule cron job* -> *create new schedule cron job*

Program Kamera Pantau

Listing program lengkapnya sebagai berikut :

```
# USAGE
# python coba_camera.py --conf
# CONF.JSON
# import the necessary packages
from picamtemp.tempimage import
TempImage from dropbox.client
import
DropboxOAuth2FlowNoRedirect
from dropbox.client import
DropboxClient import dropbox
import
argparse
import
warnings
import
datetime
import
imutils
import json
import time
import cv2

# construct the argument parser
and parse the arguments
ap =
argparse.ArgumentParser
r() ap.add argument("-
c", "--conf",
required=True,
help="path to the JSON
configuration file")
```

```

conf = json.load(open(args["conf"]))
client = None

# check to see if the Dropbox
should be used if
conf["use_dropbox"]:
    # connect to dropbox
    and start the session
    authorization process
    flow =
    DropboxOAuth2FlowNoRedirect(conf
    ["dropbo x key"],
    conf["dropbox secret"])
    print "[INFO]
    Authorize this
    application:
    {}format(flow.start())
    authCode = raw input("Enter
    auth
    code here: ")_strip()

    finish the authorization and
    grab the
    Dropbox client
    (accessToken, userID) =
    flow.finish(authCode)
    client =
    DropboxClient(accessToken)
    print "[SUCCESS] dropbox
    account
    args =
    vars(sys.argv)

    # filter warnings, load
    the configuration and
    initialize the Dropbox
    # client
    warnings.filterwarnings("ig
    nore")
;
;

linked"

camera = cv2.VideoCapture(0)

# initialize the first frame in the
video stream print "[INFO] warming
up..."
time.sleep(conf["camera_warmup_time
"]) firstFrame = None
lastUploaded = datetime.datetime.now()
motionCounter = 0

# loop over the frames of
the video while True:
    # grab the current frame and
    initialize the occupied/unoccupied
    # text
    (grabbed, frame) = camera.read()

    timestamp =
    datetime.datetime.now()
    text = "Sepi"

    # if the frame could not be
    grabbed, then we have reached the
    end
    # of the video
    if not grabbed:
        break

    # resize the frame,
    convert it to grayscale, and
    blur it
    frame =
    imutils.resize(frame,
    width=500)
    gray = cv2.cvtColor(frame,
    cv2.COLOR_BGR2GRAY)
    gray = cv2.GaussianBlur(gray, (21,
    21), 0)

    # if the first frame is None,
    initialize it

```

```

if firstFrame is None:
    print "[INFO]
starting background
model..."
    firstFrame =
    gray.copy().astype("float")
    #camera.release()
    #cv2.destroyAllWindows()
    ws()
    contin
    ue

    # accumulate the
    weighted average between the
    current frame and
    # previous frames, then
    compute the difference between
    the current
    # frame and running
    average
    cv2.accumulateWeighte
    d(gray,
    firstFrame, 0.5)
    frameDelta =
    cv2.absdiff(gray,
    cv2.convertScaleAbs(firstFr
    ame))

    # compute the absolute
    difference between the
    current frame and
    # first
    frame

    thresh =
    cv2.threshold(frameDelta,
    conf["delta_thresh"], 255,
    cv2.THRESH_BINARY)[1]

    # dilate the thresholded
    image to fill in holes, then
    find contours
    # on thresholded image

```

```

thresh =
cv2.dilate(thresh, None,
iterations=2)
(cnts, _) =
cv2.findContours(thre
sh.copy(),
cv2.RETR_EXTERNAL,

cv2.CHAIN_APPROX_SIMPLE)

# loop over the
contours for c
in cnts:
    # if the contour is
    too small,
    ignore it
    if
    cv2.contourArea(c) <
    conf["min_area"]:
        contin
        ue

```

```

# compute the
bounding box for the
contour, draw it on the
frame,

# and update
the text
(x, y, w, h) =
cv2.boundingRect(c)
cv2.rectangle(f
rame, (x, y),
(x
+ w, y + h), (0, 255, 0),
2)

text = "Ada
Gerakan"

# draw the text and
timestamp on the

()".format(text), (10, 20),

cv2.FONT_HERSHEY_SIMPLEX,
0.5, (0, 0, 255), 2)
cv2.putText(frame,
datetime.datetime.now().strftime("%A
%d %B
%H %I:%M:%S%p"),
(10, frame.shape[0] -
10), cv2.FONT_HERSHEY_SIMPLEX, 0.35,
(0, 0,
255), 1)

# jika status ruangan : Ada
Gerakan if text == "Ada
Gerakan":

# check to see if enough
time has passed between uploads
if (timestamp -
lastUploaded).seconds >=
conf["min_upload_seconds"]:
# increment the
motion counter
motionCounter +=
1

# check to see if
the number of frames with consistent
motion is
# high enough

if motionCounter
>= conf["min_motion_frames"]:
frame
cv2.putText(frame,
"Status Ruangan:

# check
to see if dropbox should be
used

i

f conf["use_dropbox"]:
#
write the image to temporary file
t
= TempImage()

cv2.imwrite(t.path, frame)

print 'save temporary file to
'+ t.path

#

SAVE TO DROPBOX

f

= open(t.path, 'rb')

ufolder =

'/camera/'

ufilename =

'KAMERA-' +
datetime.datetime.now().strftime('%Y
-%m-%d-
%H-%M-%S') + '.jpg'

upload_files_to_dropbox =
ufolder +
ufilename

response =

```

```

client.put_file(upload_files_to
dropbox, f)

    print 'File Saved to
    Dropbox: ' +
upload_files_to_dropbox

Cleanup the temporary image    #

t.clean
ur()

#
update the last uploaded
timestamp and reset the motion

# counter

lastUploaded =
timestamp

motionCounter = 0

# otherwise, the room is
not occupied else:
    motionCounter = 0

# show the frame and
record if the user presses a
key
    if conf["show_video"]:
        # display the kamera
        security
        cv2.imshow("Kamera
Security",
frame)
        key = cv2.waitKey(1) &
0xFF

# if the `q` key is pressed,
the break from
loop
    if key == ord("q"):
        break

# cleanup the camera and
close any open windows
camera.release()
cv2.destroyAllWindows()
    
```

Uji Kamera Pantau

Hasil perekaman kamera ketika status ruangan sepi



Gambar 11. Uji Coba dengan Status Ruangan Sepi

Hasil perekaman kamera ketika status ruangan ada gerakan



Gambar 12. Uji Coba dengan Status Ruangan Ada Gerakan

Hasil upload di dropbox



Gambar 13. Hasil upload sistem di Dropbox

4. KESIMPULAN DAN SARAN

Sebagai sebuah mini komputer dengan kapasitas hardware yang terbatas Raspberry Pi dengan mampu bekerja untuk digunakan sebagai kamera pemantau dengan menggunakan webcam. Keterbatasan hardware yang ada pada Raspberry Pi antara lain tidak tersedianya kapasitas penyimpanan lokal yang memadai untuk menyimpan hasil, keterbatasan pasokan daya sehingga bila menggunakan device melalui USB port nya butuh tambahan daya lain. Sistem kamera yang dibuat mempunyai beberapa kekurangan seperti ketersediaan jaringan internet untuk mengirim data, sehingga sistem berjalan tidak sesuai dengan rancang bangun seperti yang dikehendaki dari sebuah fungsi utama kamera pemantau. Karena menggunakan linux proses instalasi aplikasi yang akan dipasang bagi orang awam sedikit rumit dan tetap membutuhkan ketersediaan jaringan internet untuk mendownload repositorynya. OpenCV sebagai sebuah library pengolahan gambar yang open source sangat membantu bagi seorang programmer untuk melakukan image processing tanpa harus memikirkan algoritma-algoritma yang rumit tentang pengolahan gambar. Penyimpanan

awan yang ada sangat membantu user untuk dapat memenuhi kebutuhan tentang tempat penyimpanan selama terkoneksi dengan jaringan internet. API yang dimiliki oleh Dropbox (contoh) sebagai penyimpanan awan membantu programmer tanpa perlu rumit memikirkan algoritma pengiriman file dari program yang dibuat untuk disimpan di *cloud storage*.

Untuk pengembangan penelitian lebih lanjut, *Raspberry Pi* lebih cocok digunakan untuk sistem kontrol sensor tanpa harus menggunakan media internet untuk pengiriman data. Namun penulis menyadari bahwa dengan perkembangan teknologi yang ada kendala-kendala yang ada pada suatu mini komputer dapat diatasi.

5. REFERENSI

- [1] Ahmad Usman, 2005, Pengolahan Citra Digital dan Teknik Pemrogramannya, Yogyakarta: Graha Ilmu.
- [2] Brendan Horan, 2013, Practical *Raspberry Pi*, ISBN-13 (electronic): 978-1-4302-4972-6
- [3] Maik Schmidt, 2012, *Raspberry Pi - A Quick-Start Guide*, ISBN-13: 978-1-937785-04-8
- [4] Mike Cook, 2014, *Raspberry Pi Projects*, ISBN 978-1-118-55553-8