

PERANCANGAN SISTEM KEAMANAN RUANGAN MENGUNAKAN RASPBERRY PI (bagian: aplikasi)

DESIGN OF ROOM SECURITY SYSTEM USING RASPBERRY PI (part: application)

Nandana Adya Samudera

Prodi S1 Teknik Komputer, Fakultas Teknik Elektro, Universitas Telkom
Jalan Telekomunikasi no.1 Ters.Buah Batu Bandung 40257, Indonesia
samudera.nanda@gmail.com

Abstrak

Sistem keamanan ini dibangun pada Raspberry Pi, sistem ini dilengkapi dengan modul picamera, modul sensor PIR (*passive infra red sensor*) untuk mendeteksi keberadaan manusia disekitar, dan modem gsm untuk mengirimkan notifikasi sms. Program untuk sistem ini dibangun menggunakan bahasa pemrograman Python dan Flask untuk framework web.

Sistem ini dibagi menjadi dua bagian, yaitu bagian Aplikasi dan Alert. Aplikasi berfungsi sebagai antarmuka pengguna, seperti streaming video, menampilkan laporan, melihat video yang terekam, mengeset konfigurasi sistem. Bagian Alert berperan mengirim sms, merekam video dan memasukan data-data sistem ke dalam database ketika manusia terdeteksi pada ruangan dan program aktif. Sedangkan pada penelitian ini difokuskan pada bagian Aplikasi

Setelah melakukan perancangan dan implementasi, tahap selanjutnya adalah dengan melakukan pengujian terhadap aplikasi. Pada pengujian yang telah dilakukan dapat disimpulkan bahwa aplikasi dapat mendukung fitur-fitur untuk sistem keamanan ruangan. Berdasarkan pengujian video streaming mempunyai rata-rata delay terendah 210 ms untuk resolusi dan fps terendah sampai dengan 1784 ms untuk resolusi dan fps tertinggi, penggunaan CPU cukup tinggi yaitu sekitar 98 persen.

Kata Kunci: Raspberry Pi, picamera, Python, Flask, PIR.

Abstract

This Security System based was built on Raspberry Pi, it is equipped with picamera module, passive infra red sensor module to detect presence of human around, and gsm modem to send notification SMS. Program for this system was built with Python programming language and Flask for web framework.

This system divided into two parts, application and alert. The Application serves as user interface, such as streaming video, view reports notification, watch recorded video, setting system configuration. Function of Alert is to send notification sms, record video, and input data system into database when human were detected around and the program is activated. This research focused on the application.

After doing design and implementation, next step is perform testing of application. Based on testing video streaming had average delay about 210 ms for lowest resolution and fps to 1789 ms for highest resolution and fps, CPU usage is high at about 98 percent.

I. Pendahuluan

Semakin kerasnya kehidupan membuat orang-orang menghalalkan segala cara untuk memenuhi kebutuhan hidup mereka seperti merampok, mencuri, dan tindakan kriminal lainnya. Salah satu perbuatan yang marak adalah tindakan pencurian rumah, hal ini meningkat apalagi ketika masa liburan dimana para pemilik rumah biasanya meninggalkan rumahnya untuk liburan. Pencuri jaman sekarang semakin nekat dan pintar untuk mencuri. Maka dibutuhkan kewaspadaan yang ekstra dalam menjaga keamanan rumah. Solusi yang biasa digunakan para pemilik rumah hanya dengan mengandalkan kunci atau gembok rumah untuk mengamankan rumah mereka. Untuk itu dibutuhkan sesuatu yang mendukung keamanan ekstra rumah kita agar terhindar dari para pencuri tersebut.

Teknologi kini semakin berkembang dengan pesat, dengan bantuan teknologi kita dapat membantu dan mempermudah kegiatan kita sehari-hari, salah satunya kita bisa memanfaatkannya untuk membantu mengamankan rumah kita dari para pencuri. Salah satu teknologi yang marak berkembang yaitu komputer

seukuran kartu kredit bernama Raspberry Pi. Dengan bantuan raspberry pi, kamera, dan koneksi internet kita bisa memanfaatkannya untuk menjadi sistem keamanan rumah yang murah dan bisa diakses dimana saja dan kapan saja, sehingga bisa melihat keadaan rumah kita tanpa perlu khawatir meskipun dalam keadaan kosong sekalipun. Dengan begitu kita berharap dapat membuat rumah kita lebih aman dari para pencuri.

Dari uraian di atas penulis tertarik untuk merancang dan mengaplikasikan sistem keamanan rumah dengan menggunakan Raspberry, kamera dan modul sensor. Raspberry berfungsi untuk memproses streaming, sehingga pemilik rumah bisa memantau keadaan rumah secara *live*. Selain streaming sistem ini juga mampu mendeteksi gerakan, dan menyimpan serta memberi notifikasi ke pemilik rumah.

2. Dasar Teori

2.1 Sistem Keamanan Ruang

Yang dimaksud dengan sistem keamanan ruangan adalah sebuah sistem atau alat yang dibuat untuk mengawasi atau yang berhubungan dengan keamanan pada sebuah ruangan pada rumah, kantor, toko atau lokasi tertentu yang memerlukan keamanan khusus.

Pada era modern ini *ada beberapa produk yang dibuat untuk menjaga keamanan suatu lokasi, contohnya* adalah CCTV, ip kamera, dan bahkan sekarang terdapat produk *surveillance* yang dipadu dengan smarthome, sebagai contoh adalah Piper dan Samsung Smartcam. Pada alat-alat tersebut mempunyai tujuan yang sama yaitu memantau pada suatu lokasi yang biasanya digunakan untuk alasan keamanan. Fitur yang ditawarkan dari alat-alat tersebut pun bermacam-macam seperti alarm, sensor untuk deteksi adanya pergerakan, perekam otomatis, pemantau dari jarak jauh melalui *video streaming* dan lain-lain. Dengan fitur-fitur tersebut diharapkan dapat menambah kualitas dan efektifitas pada keamanan, sehingga dapat mengurangi resiko pencurian.

2.1 Raspberry Pi

Raspberry Pi adalah sebuah komputer berpapan tunggal (*Single Board Computer*) seukuran kartu kredit yang dikembangkan oleh Yayasan Raspberry Pi di Inggris. Walaupun berukuran kecil, Raspberry Pi cukup handal untuk melakukan tugas-tugas yang dapat dilakukan oleh komputer pada umumnya seperti membuat laporan, bermain game, memutar video ataupun musik, bahkan Raspberry Pi dapat digunakan sebagai *web server*, dan *media server*.



Gambar 2.1 Raspberry Pi
(sumber: www.raspberrypi.com)

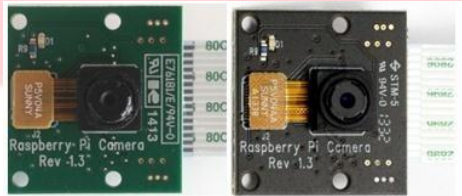
Selain komponen-komponen komputer pada umumnya, Raspberry Pi juga terdapat komponen pin GPIO, dengan adanya pin ini maka Raspberry Pi dapat mendukung beberapa modul salah satunya modul sensor infra merah, selain itu Raspberry Pi juga dapat dihubungkan dengan Arduino. Hingga saat ini Raspberry Pi memiliki beberapa model yaitu Raspberry Pi model A, model A+, model B, model B+, dan yang terbaru adalah model B generasi ke dua. Model B mendukung beberapa komponen yang tidak ada pada model A, seperti tambahan *port* USB, dan RAM yang lebih besar.

Berikut ini adalah komponen-komponen *input output* pada Raspberry Pi:

- HDMI, untuk dihubungkan ke monitor LCD yang mendukung port HDMI atau converter VGA to HDMI untuk LCD yang tidak mendukung port HDMI.
- *Video analog* (RCA *port*), dihubungkan ke monitor televisi.
- *Audio output*, keluaran untuk suara, dihubungkan ke *speaker*.
- *Port* USB 2.0.
- Pin GPIO, untuk menghubungkan dengan LED, sensor, alarm, dll.
- *Port* CSI (Camera Serial Interface)
- *Port* DSI (Display Serial Interface)
- *Ethernet output*, dihubungkan dengan kabel UTP/STP.
- *SD card slot*, untuk keperluan penyimpanan data.

2.2 Modul Kamera Raspberry Pi

Modul kamera Raspberry biasa disebut Picamera atau Rasicam adalah modul kamera yang didesain khusus untuk Raspberry Pi. Pada Picamera terdapat kabel pita yang dapat dihubungkan ke *CSI Connector* yang berada pada Raspberry Pi. Rasicam pertama dirilis pada 14 Mei 2013 dan pada 28 Oktober 2013 dirilis versi "Pi NoIR" yang merupakan versi Picamera tanpa infra merah, ciri-ciri dari Picamera versi ini adalah mempunyai PCB yang berwarna hitam. Pada keduanya mempunyai ukuran 25mm x 20mm x 9mm dengan berat sekitar 3 gram.



Gambar 2.2 Modul Picamera IR dan NoIR
(sumber: www.elinux.org)

Modul kamera Raspberry Pi dapat bekerja pada semua model Raspberry Pi. Pada pemrograman Python diperlukan *library* picamera untuk dapat menjalankan fungsi modul kamera ini.

2.3 Python

Python adalah sebuah bahasa pemrograman yang bisa digunakan pada beberapa *platform* (*multiplatform*), dan berifat sumber perangkat bebas terbuka (*opensource*), pertama kali dikembangkan oleh Guido van Rossum pada tahun 1990 di CWI, Belanda. Bahasa ini dikategorikan sebagai bahasa tingkat tinggi (*very-high-level language*) dan merupakan bahasa berorientasi objek yang dinamis (*object-oriented-dynamic language*).

Hal utama yang membedakan Python dengan bahasa lain adalah dalam hal aturan penulisan kode program. Python memiliki aturan yang berbeda dengan bahasa lain, seperti indentasi, tipe data, *tuple*, dan *dictionary*.

Python adalah bahasa pemrograman dinamis yang mendukung pemrograman berorientasi obyek. Python dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai sistem operasi seperti Linux, Windows, Unix, Symbian dan masih banyak lagi. Python merupakan salah satu bahasa pemrograman favorit saat ini, karena Python menawarkan banyak fitur seperti:

- Kepustakaan yang luas, menyediakan modul-modul untuk berbagai keperluan.
- Mendukung pemrograman berorientasi objek.
- Memiliki tata bahasa yang mudah dipelajari.
- Memiliki sistem pengelolaan memori otomatis.
- Arsitektur yang dapat dikembangkan (*extensible*) dan ditanam (*embeddable*) dalam bahasa lain, misal *objek oriented* Python dapat digabungkan dengan modul yang dibuat dengan C++.

Python telah digunakan pada berbagai aplikasi saat ini, contohnya adalah BitTorrent, Yum, Civilization 4, bahkan saat ini Python merupakan bahasa resmi dari Raspberry Pi. Kata "Pi" dalam Raspberry Pi merujuk pada kata Python. Python mendukung beberapa modul khusus untuk Raspberry Pi seperti modul picamera, dan modul gpio.

2.4 Flask

Flask adalah sebuah aplikasi *microframework* untuk bahasa Python yang dibuat dengan *toolkit* Werkzeug dan *template* Jinja2. Flask dibuat oleh Armin Ronacher. Pertama kali dirilis pada April 2010. Bila dibandingkan dengan Django, Flask jauh lebih ringan dan cepat karena Flask dibuat dengan ide menyederhanakan inti frameworknya seminimal mungkin, dengan *tagline* "*web development, one drop at a time*", oleh karena itu Flask disebut *microframework*. Flask dapat membantu kita membuat situs dengan cepat meskipun dengan *library* yang sederhana. Contoh aplikasi yang menggunakan *framework* Flask adalah Pinterest, dan LinkedIn. Flask saat ini berada pada lisensi 0.10.1 dan dilisensikan dengan lisensi BSD.

Aplikasi yang dibuat dengan Flask disimpan dalam satu berkas ".py". Flask adalah *framework* yang sederhana namun dapat diperluas dengan beragam pustaka tambahan yang disesuaikan dengan kebutuhan penggunaannya. Meskipun Flask belum menyampai versi 1.0 namun dokumentasi yang dimilikinya sangat lengkap.

2.5 Video Streaming

Istilah video streaming terdiri dari dua buah suku kata yaitu video dan *streaming*, video berarti teknologi untuk menangkap, merekam, memposes, mentransmisikan dan menata ulang gambar yang bergerak, sedangkan *streaming* adalah sebuah proses penghantaran data dalam aliran berkelanjutan, dan tetap memungkinkan pengguna mengakses dan menggunakan file sebelum data dihantar sepenuhnya. Jadi *video streaming* adalah

sebuah teknologi yang memungkinkan kita untuk mentransmisi file video secara berkelanjutan yang memungkinkan video tersebut dapat dimainkan tanpa menunggu file video tersebut diunduh terlebih dahulu secara keseluruhan. File video yang *distreaming* akan ada sebuah *buffer* di komputer client, dan data video tersebut akan mulai didownload dalam buffer yang telah terbentuk di komputer client. Sistem akan membaca informasi dari buffer dan tetap melakukan proses download file, sehingga proses streaming akan berlangsung ke komputer client.

Terdapat dua jenis dalam video *streaming*, yaitu *live* dan *archived*, *Archived* atau *on-demand* adalah proses *streaming* pada sebuah video yang sudah terekam (sudah berbentuk sebuah file yang telah tersimpan dalam server), contohnya adalah ketika kita menonton video klip pada youtube. Sedangkan *live streaming* adalah konten yang dikirimkan secara langsung atau *realtime* melalui internet, hal ini biasanya membutuhkan media untuk merekam seperti video kamera. Jadi video yang didapatkan dari *live streaming*, sesegera mungkin ditransmisikan dan dapat diputar melalui internet.

2.6 MJPEG

MJPEG (*Motion Joint Photographic Expert Group*) adalah sebuah format kompresi video dengan cara kerjanya adalah memproses setiap *frame* video secara terpisah dan mengkompresnya menjadi gambar berformat JPEG secara berurutan *frame* demi *frame*. Setiap *frame* ditransmisikan diproses secara independen, sehingga dikirim *frame* demi *frame* ke tujuan dan kemudian dihubungkan ke dalam sebuah video, dengan kata lain MJPEG adalah sebuah kompresi video yang terbentuk dari gambar-gambar MJPEG yang disatukan menjadi sebuah video. Hal ini mengakibatkan perlunya kecepatan transfer data yang lebih besar dibandingkan dengan teknologi kompresi yang lain, namun memiliki kelebihan tidak menguras sistem *memory* secara berlebihan.

MJPEG biasa ditemukan dimana saja khususnya pada aplikasi ip kamera. MJPEG populer digunakan dimana saja khususnya pada ip kamera. Salah satu keuntungan menggunakan MJPEG adalah rendahnya *delay* pada *video streaming*, selain itu *stream* MJPEG dapat diputar pada kebanyakan browser pada saat ini, beberapa *device*-pun mendukung MJPEG seperti iPhone, iPad, Sony Playstation dan Android dapat memainkan format MJPEG.

Dikarenakan MJPEG sering kali di-*encode* melalui protokol http dan ditampilkan di web browser, maka MJPEG membutuhkan MIME khusus yaitu "*multipart/x-mixed-replace*". Berikut ini adalah struktur MJPEG pada protokol http:

Pada header MIME, ditentukan tipe dari MIME pada parameter "Content-Type", selanjutnya ditentukan parameter "boundary" sebagai pemisah antara frame jpeg.

3. Analisis dan Perancangan

3.1 Gambaran Umum Sistem Keseluruhan

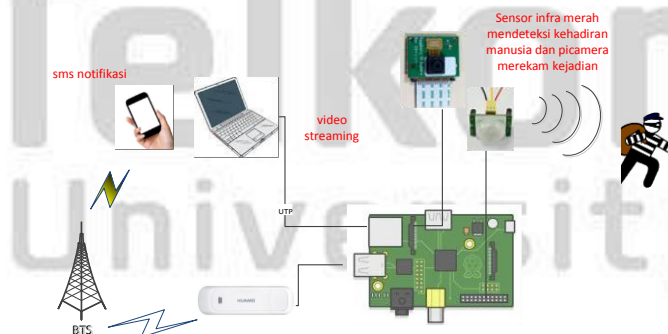
Sesuai dengan penjabaran sebelumnya, tujuan utama dari pembuatan sistem ini dikarenakan semakin maraknya pencurian sehingga menimbulkan kekhawatiran pemilik ruangan terutama ketika pemilik tersebut sedang berada di luar kota, maka dari itu perlu dibuat sebuah sistem yang diharapkan dapat mengatasi masalah tersebut. Adapun sistem ini mempunyai fitur sebagai berikut:

a. Mendeteksi kehadiran manusia dan merekam video

Merekam dan mengirim SMS apabila ada terdeteksi kehadiran manusia. Sistem ini juga dilengkapi dengan sensor infra merah yang dapat mendeteksi kehadiran manusia yang berada di area sensor, sehingga ketika terdeteksi sistem dapat merekam serta mengirimkan notifikasi sms ke pengguna.

b. Pemantauan dengan video streaming

Dengan sistem keamanan ruangan ini pengguna dapat mengawasi ruangan melalui PC/*device* dimana saja (selama terhubung internet) melalui fitur *live video streaming* sehingga tidak perlu adanya kekhawatiran apabila berada di luar kota.

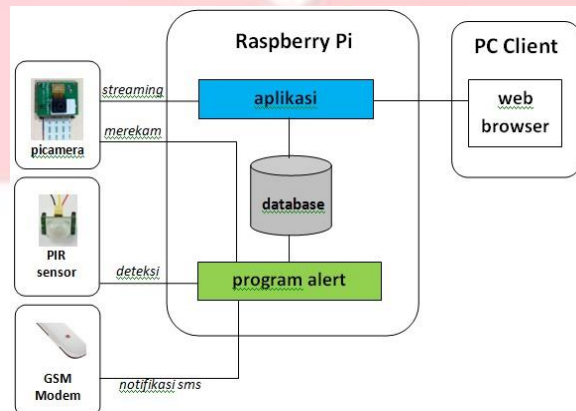


Gambar 3.1 Gambaran Umum Sistem Keseluruhan

(sumber: shutterstock.com, raspberrypi.org, trendmicro.com, storybird.com, huawei.com,)

3.1.1 Arsitektur Sistem Keseluruhan

Sistem keamanan ruangan ini berupa program Python yang ditanamkan pada Raspberry Pi yang dilengkapi dengan beberapa modul guna mengoptimalkan sistem. Berikut ini adalah gambaran arsitektur sistem secara keseluruhan:



Gambar 3.2 Arsitektur Sistem Keseluruhan

Berikut ini adalah bagian-bagian perangkat keras yang terdapat pada sistem:

- Raspberry Pi, digunakan untuk pemrosesan data, *video streaming*, mengolah *input* dan *output*.
- Pi-Camera, digunakan untuk merekam video dan *video streaming*.
- PIR sensor, digunakan untuk mendeteksi kehadiran manusia.
- Modem GSM, digunakan untuk mengirim sms peringatan ke telepon genggam pengguna.
- PC pengguna yang dilengkapi dengan web browser, digunakan untuk mengakses aplikasi yang ada di dalam raspberry pi.

Sedangkan pada sisi perangkat lunak terbagi menjadi dua, yaitu:

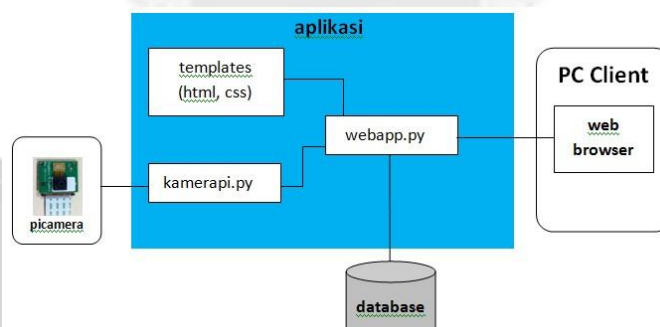
- Program aplikasi, lebih difokuskan kepada interaksi kepada pengguna, seperti menampilkan *video streaming*, menampilkan video yang terekam, melihat laporan, dan konfigurasi sistem.
- Program alert, program ini berfungsi mengirimkan sms peringatan dan merekam video bila sensor infra merah mendeteksi adanya kehadiran manusia di area sensor.

Diantara program aplikasi dan alert terpisahkan oleh database, database ini berfungsi untuk menyimpan data konfigurasi, data video dan data log. Pada data-data inilah yang nantinya menjadi penentu dari program alert apakah program merekam atau mengirim notifikasi sms atau tidak.

3.2 Perancangan Aplikasi

Karena dalam tugas akhir ini lebih difokuskan kepada bagian aplikasi maka buku ini tidak membahas mengenai program alert. Dalam poin ini akan dijelaskan mengenai perencanaan dan rancangan pembuatan aplikasi.

3.2.1 Arsitektur Aplikasi



Gambar 3.3 Arsitektur Aplikasi

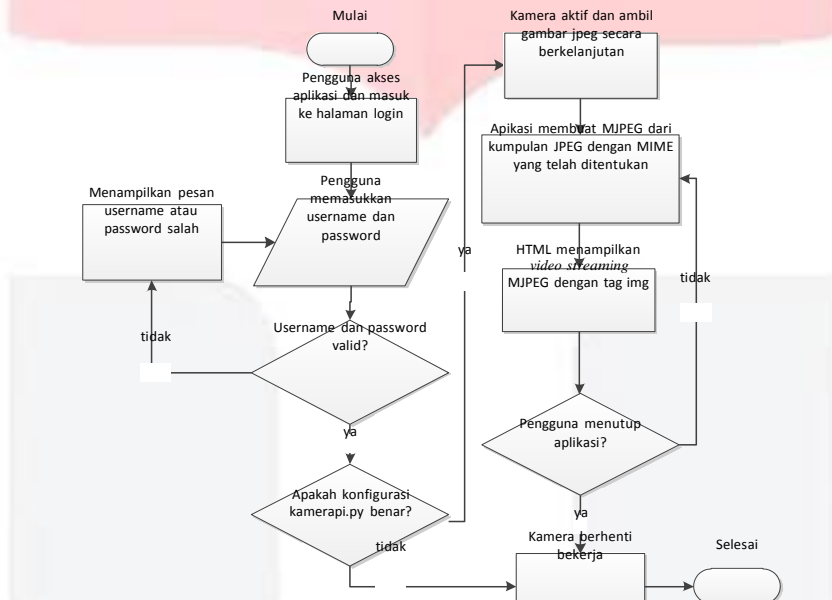
Dalam aplikasi terdapat tiga bagian yaitu *webapp*, *kamerapi* dan *templates*. Saat Raspberry Pi dinyalakan maka secara otomatis sistem akan memanggil program *webapp* dan program alert. Program *webapp* merupakan program utama dari sistem aplikasi, program ini bertugas untuk memanggil program *kamerapi* yang berguna untuk melakukan fitur *video streaming* berbentuk *motion jpeg*, hasil output dari program ini ditampilkan lewat halaman *html* yang tersimpan dalam folder *templates*. Folder *templates* ini berisi file-file halaman *html* dan *css*, file-file inilah yang menampilkan *user interface* dan *video streaming* pada *web browser* pengguna.

3.2.2 Spesifikasi Aplikasi

Aplikasi ini dirancang memiliki spesifikasi sebagai berikut:

- Stream: Pengguna mampu melakukan *live streaming video* dan menampilkannya di browser pengguna.
- Log: Program mampu menampilkan *log history* mengenai laporan tentang sensor yang mendeteksi adanya kehadiran manusia, terkirimnya sms, dan video yang terekam.
- Video: Menampilkan video yang telah terekam.
- Config: Halaman konfigurasi yang dapat mengaktifkan dan menonaktifkan rekam video.

3.2.3 Skema Kerja Video Streaming

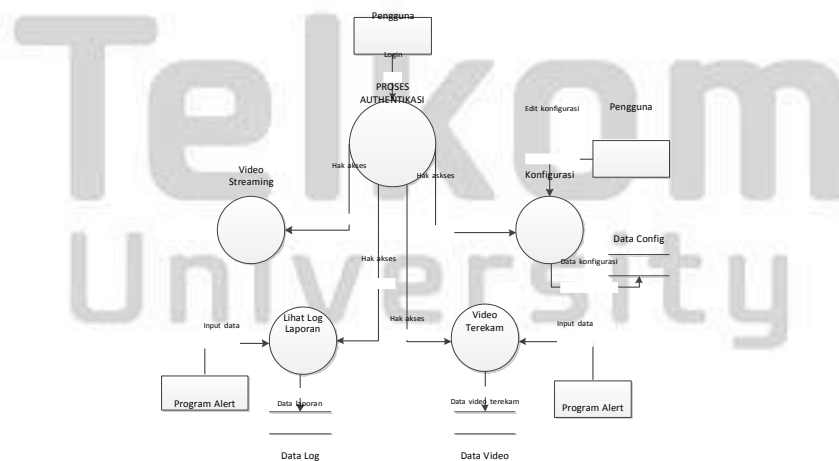


Gambar 3.4 Skema kerja video streaming

Berikut ini adalah tahapan-tahapan alur kerja *video streaming* pada sistem keamanan ruangan dari mulai pengambilan *frame* oleh program pada picamera sampai gambar tersebut ditangkap menjadi *video streaming* secara langsung. Pada saat pengguna mengakses aplikasi, program akan mengecek konfigurasi untuk video streaming, apakah valid atau tidak? Jika valid maka akan dilanjutkan ke tahap halaman login, sebelum mengakses *video streaming* pengguna diharuskan untuk mengecek autentikasi pengguna dengan memasukkan username dan password pengguna, jika login benar maka aplikasi akan menampilkan halaman *video streaming*, dan kamera mulai aktif bekerja, pada tahap ini kamera mengambil gambar jpeg secara terus menerus dan pada tahap ini html menampilkan *video streaming* dalam bentuk video MJPEG dengan tag “img”. Jika pengguna menutup aplikasi atau mengganti halaman maka kamera berhenti bekerja dan aplikasi berhenti melakukan *video streaming*.

3.2.4 Data Alir Diagram

Data Alir Diagram (disingkat DAD) aplikasi sitem keamanan ini terdiri dari dua level, yaitu level 0 da level 1. Berikut ini adalah diagram konteks dan data alir diagram level 0 aplikasi sistem keamanan ruangan:



Gambar 3.5 Data Alir Diagram Aplikasi Level 0

Seperti yang telah disebutkan pada sub-bab Spesifikasi Aplikasi, aplikasi keamanan ruangan ini terdiri dari empat bagian, yaitu Stream, Log, Video dan Config. Yang dapat mengakses aplikasi hanyalah pengguna. Untuk dapat mengakses bagian-bagian tersebut pengguna harus melakukan autentikasi terlebih dahulu dengan login sesuai username dan password yang telah ditentukan. Setelah proses autentikasi berhasil dilakukan, pengguna dapat mengubah konfigurasi sistem didalamnya, konfigurasi tersebut antara lain nomer sms notifikasi pengaktifan program Alert dan pada bagian Log dan Video pengguna hanya dapat melihat dan memainkan video yang telah terekam tanpa bisa mengubah data yang ada di dalamnya, bagian ini diinputkan oleh program Alert, hal ini dilakukan ketika sensor mendeteksi adanya kehadiran manusia dan opsi program alert yang berada pada halaman config telah diaktifkan.

3.2.5 Perancangan Basis Data

Basis data pada sistem ini sangat sederhana, karena basis data hanya untuk menyimpan data video, log laporan dan data konfigurasi. Basis data yang digunakan adalah sqlite3.

Tabel yang digunakan pada aplikasi ini berjumlah empat tabel yaitu tabel autentikasi, log, video dan config.

3.2.6 Perancangan Antarmuka

Salah satu hal penting yang menjadi perhatian pada sebuah perancangan adalah bahwa rancangan yang dibuat diharapkan dapat digunakan dengan mudah oleh semua pengguna. Sesuai dengan pembagian spesifikasi, aplikasi ini mempunyai 4 halaman, yaitu:

1. Stream, berfungsi untuk menampilkan video dari picamera dengan *live streaming*.
2. Log, menampilkan laporan seperti laporan sensor mendeteksi panas tubuh manusia, laporan sms terkirim, laporan video terekam.
3. Video, menampilkan video-video yang terekam ketika sensor mendeteksi adanya panas tubuh manusia
4. Config, merupakan halaman konfigurasi, konfigurasi ini berupa checkbox apakah video akan merekam dan/atau sistem mengirim sms notifikasi apabila sensor terdeteksi.

3.3 Kebutuhan Sistem

Berikut ini adalah perangkat lunak dan perangkat keras yang dibutuhkan untuk menunjang pembuatan aplikasi.

3.3.1 Kebutuhan Perangkat Lunak

Perangkat lunak yang digunakan dalam pembuatan aplikasi ini terdiri dari:

- Server:
 - Debian Wheezy OS
 - Flask 0.10.1
 - Python 2.7
 - Sqlite 3.7.9
- Client
 - Putty
 - Chrome web browser

3.3.2 Kebutuhan Perangkat Keras

Perangkat keras yang digunakan untuk pembuatan aplikasi ini terdiri dari:

- Server:
 - Raspberry Pi Model B Rev 2, dengan spesifikasi:
 - RAM 512 MB
 - Prosesor ARM1176JZF-S 700 MHz
 - Video output: RCA, HDMI
 - Ethernet port
 - Power: 700 mA
 - Picamera Rev 3.0
- Client:
 - Laptop Asus A46C, dengan spesifikasi:
 - RAM 4 GB
 - Prosesor Intel Core i5-3317U CPU @1,7GHz

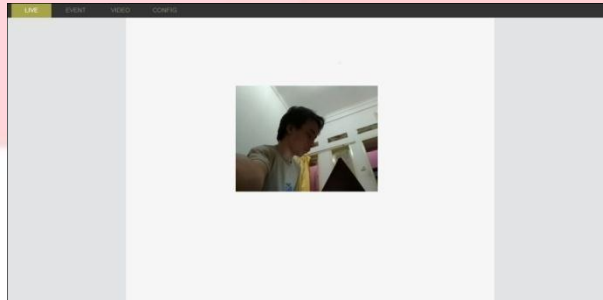
4. Implementasi dan Pengujian Sistem

4.1 Implementasi Sistem

Tahap implementasi dilakukan setelah perancangan selesai dilakukan dan selanjutnya diimplementasikan pada bahasa pemrograman yang digunakan. Implementasi sistem ini merupakan tahap meletakkan sistem sehingga siap untuk dioperasikan.

4.1.1 Implementasi Antarmuka

Implementasi antarmuka adalah *interface* yang menghubungkan pengguna dan aplikasi sehingga dapat mengakses aplikasi dengan mudah. Untuk implementasi antarmuka lebih lengkap dapat dilihat dalam lampiran.



Gambar 4.1 Implementasi Antarmuka

4.1.2 Implementasi Program

Pada tahap ini dilakukan pembuatan program pada aplikasi agar dapat berjalan sesuai dengan yang diharapkan. Pada pembuatan program dilakukan dengan menggunakan bahasa Python 2.7. Untuk lebih lengkap implementasi program dapat dilihat di halaman lampiran.

4.2 Pengujian Sistem

Setelah tahap implementasi maka tahap selanjutnya adalah pengujian sistem, hal ini dilakukan untuk menguji performa sistem. Pada tahap ini dibagi menjadi dua, yaitu pengujian *video streaming* dan pengujian aplikasi.

4.2.1 Pengujian Video Streaming

Pengujian ini dilakukan untuk menguji kemampuan Raspberry Pi untuk mengolah *video streaming*. Pada pengujian streaming sendiri dibagi menjadi tiga bagian yaitu:

1. Pengujian *delay video streaming*.
2. Pengujian batas maksimum resolusi dan *frame rate* Picamera.
3. Pengujian beban CPU dan memory pada Raspberry Pi.

Pada pengujian *video streaming* dilakukan dengan menghubungkan antara laptop dan Raspberry Pi menggunakan kabel UTP secara *peer-to-peer*.

4.2.1.1 Pengujian Delay Video Streaming

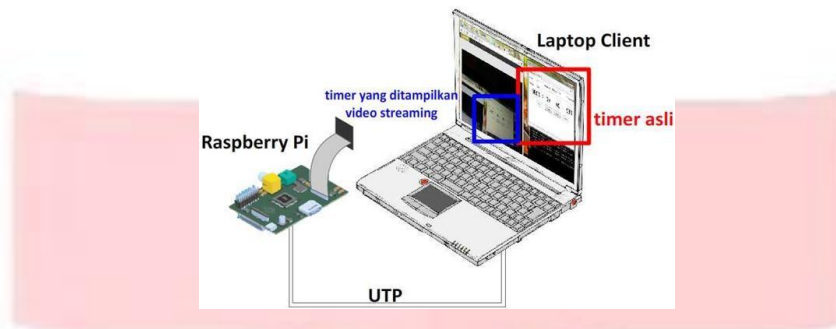
Pengujian ini dilakukan untuk mengetahui jeda waktu antara saat pengambilan gambar dan saat gambar ditampilkan pada laptop pengguna.

a. Skenario Pengujian

Skenario pengujian ini adalah dengan mengubah-ubah *frame rate* dan resolusi dari *video streaming*, dengan pengujian ini diharapkan dapat mengetahui delay video streaming dari masing-masing *frame rate* dan resolusi yang telah ditentukan. Frame rate dalam skenario ini antara lain 10 fps, 20 fps, dan 30 fps sedangkan macam resolusi adalah dengan rasio 16:9, yaitu 320 x 180, 640 x 320, 1280 x 720, dan 2560 x 1440 piksel. Pada pengujian ini juga menggunakan *tools* untuk yang dipasang pada laptop pengguna. *Tools* tersebut antara lain:

1. Countdown Timer, aplikasi *stopwatch* berbasis desktop. Digunakan sebagai pembandingan antara waktu timer yang disorot oleh kamera dan yang ditampilkan pada *video streaming*.
2. AutoCap, aplikasi untuk *screenshot* pada selang waktu yang telah ditentukan. Digunakan untuk meng-capture timer setiap detik, sehingga didapat data *delay*.

Pengujian ini dilakukan dengan picamera yang menyorot ke laptop pengguna yang telah dijalankan program Countdown Timer. Pada tahap ini maka hasil sorotan akan ditampilkan pada laptop pengguna dengan video streaming, dengan begitu akan terlihat jeda waktu video streaming dengan membandingkan antara waktu yang ditunjukkan pada Countdown dan waktu Countdown Timer yang ditampilkan pada video streaming, dengan cara ini diperlukan AutoCap untuk *screenshot* pada selang waktu tertentu untuk didapatkan data pengujian *delay*.

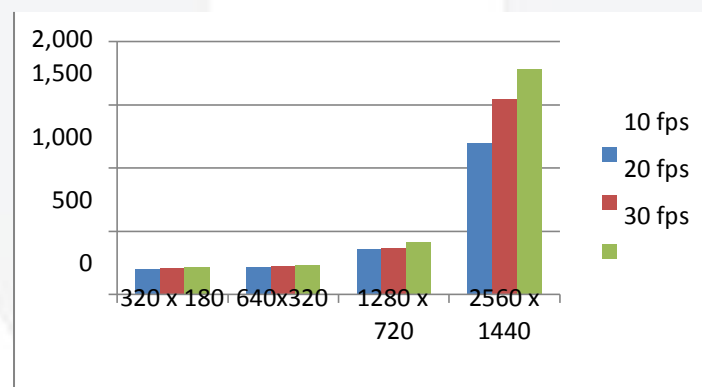


Gambar 4.2 Skenario Pengujian *Delay Video Streaming*

Pada contoh gambar di atas didapatkan *delay* salah satu *screenshot* adalah waktu yang ditunjukkan Countdown Timer asli ($T_2 = 00:530$) dikurangi dengan waktu Countdown Timer yang ditampilkan pada video streaming ($T_1 = 00:320$), maka hasilnya adalah 210 *milisecond*.

b. Hasil Pengujian

Setelah pengujian dilakukan didapatkan data nilai besar *delay* dari masing-masing resolusi dan *frame rate* pada pengujian *video streaming* (tabel ada pada lampiran). Data dari hasil pengujian itu dirata-rata dan dibuat grafik. Berikut ini adalah hasil rata-rata pengujian *delay video streaming*:



Gambar 4.3 Grafik hasil pengujian *delay video streaming*

Berdasarkan pengujian di atas didapatkan hubungan *delay* dengan besar resolusi dan *frame rate*, yaitu besarnya resolusi dan *frame rate* berbanding lurus dengan *delay*, semakin besar resolusi dan *frame rate*, maka *delay* yang didapat juga semakin besar, begitu juga sebaliknya. Hal ini dikarenakan semakin besar jumlah resolusi dan *frame rate* maka *resource* yang digunakan untuk mentransfer *video streaming* ke laptop pengguna semakin besar pula.

Menurut grafik hasil pengujian di atas perbandingan *delay* antara resolusi 320 x 180 dengan resolusi 640 x 320 tidak begitu besar, pada resolusi 1280 x 720 *delay* bertambah kurang lebih dua kali lipat dari resolusi sebelumnya, sedangkan pada resolusi 2560 x 1440 *delay* bertambah cukup drastis hingga mencapai kurang lebih empat kali lipat dari resolusi sebelumnya.

4.2.1.2 Pengujian Waktu Proses Streaming Raspberry Pi

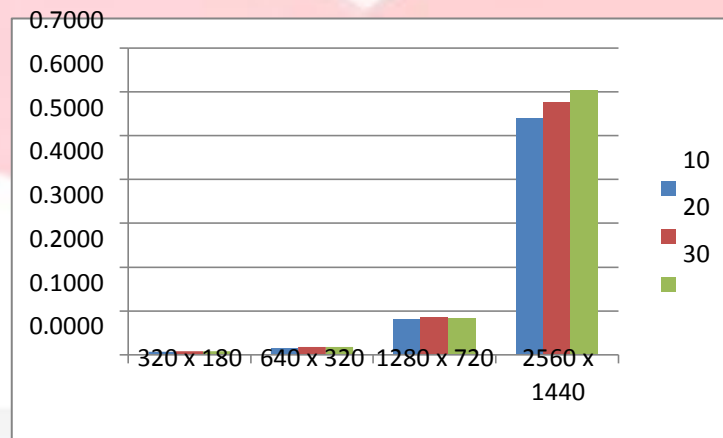
Pengujian ini dilakukan untuk mengetahui lama waktu Raspberry Pi dalam proses pengolahan *video streaming* sebelum ditransfer ke client.

a. Skenario Pengujian

Ketika mjpeg dirangkai dari kumpulan jpeg proses streaming mulai dilakukan, pada saat proses streaming dicatat berapa waktu stream yang dihasilkan dari proses *video streaming* tersebut. Setelah waktu setiap stream tercatat maka didapat data selisih dari setiap stream, selisih waktu tersebut adalah waktu proses pengolahan pada setiap streaming

b. Hasil Pengujian

Setelah pengujian dilakukan maka didapat data selisih waktu streaming dari setiap konfigurasi. Data-data tersebut lalu dimasukkan ke dalam tabel dan dari tabel tersebut diambil rata-rata selisih waktu setiap stream dari setiap konfigurasi.



Gambar 4.4 Grafik Pengujian Waktu Proses Streaming Raspberry Pi

Dari pengujian lama waktu proses Raspberry Pi ini didapat ketika resolusi diset pada 2560 x 1440 waktu lama proses Raspberry Pi meningkat menjadi di atas 500 milidetik.

4.2.1.3 Pengujian Frame Rate

Pengujian ini dilakukan untuk mengetahui berapa *frame rate* yang dikirimkan Raspberry Pi ke komputer *client*, apakah sesuai dengan konfigurasi atau tidak pada resolusi berapa jumlah frame rate menurun.

a. Skenario Pengujian

Skenario pengujian ini adalah dengan mencatat waktu setiap frame yang dihasilkan dari proses *streaming*. Setelah data waktu setiap frame dicatat maka didapatkan berapa *frame* yang dihasilkan dari *video streaming* pada setiap detiknya.

b. Hasil Pengujian

Tabel 4.3 Tabel pengujian jumlah frame rate

Resolusi	Frame rate pada konfigurasi	Frame rate hasil pengujian
320 x 180	10 fps	10 fps
	20 fps	17 fps
	30 fps	17 fps
640 x 320	10 fps	10 fps
	20 fps	15 fps
	30 fps	15 fps
1280 x 720	10 fps	10 fps
	20 fps	15 fps
	30 fps	15 fps
2560 x 1440	10 fps	5 fps
	20 fps	5 fps
	30 fps	5 fps

Dari pengujian di atas diketahui jumlah frame rate yang dihasilkan tidak sesuai dengan konfigurasi, dan ketika resolusi diset pada 2560 x 1440 frame rate yang dihasilkan mulai menurun drastis hingga 5 fps. Dari pengujian ini dapat disimpulkan bahwa resolusi optimum video streaming pada Raspberry Pi dan picamera adalah 1280 x 720.

4.2.1.4 Pengujian beban CPU dan memory pada Raspberry Pi

Pengujian yang ketiga ini bertujuan untuk mengetahui berapa CPU dan *memory* yang dikonsumsi Raspberry Pi untuk *video streaming*.

a. Skenario Pengujian

Pengujian ini dilakukan dengan menggunakan *tools* htop, htop adalah sebuah aplikasi untuk mengetahui berapa besar *memory* dan berapa persen CPU yang digunakan saat proses komputer berjalan. Skenario pengujian ini adalah dengan mengubah *frame rate* dan resolusi sesuai dengan pada pengujian *delay*, yaitu dengan resolusi 320 x 180, 640 x 320, 1280 x 720, 2560 x 1440 dan *frame rate* 10 fps, 20 fps, dan 30 fps dan saat *video streaming* tidak dijalankan lalu melihat presentase CPU dan besar *memory* yang digunakan saat proses video streaming berlangsung selama kurang lebih satu menit. Masing-masing hasil dari pengujian dengan htop lalu dicatat, berapa sampai berapakah *memory* dan CPU yang dikonsumsi Raspberry Pi untuk *video streaming*. Berikut ini adalah contoh tampilan htop ketika video streaming belum dijalankan:

b. Hasil Pengujian

Setelah masing-masing pengujian dilakukan selama kurang lebih satu menit, didapatkan nilai CPU dan *memory* yang dikonsumsi Raspberry Pi, data tersebut dikumpulkan dan dibuat tabel.

Tabel 4.4 Tabel CPU dan *memory* yang dikonsumsi Raspberry Pi

Idle		CPU (%)	Memory (MB)
320 x 180	10 fps	94 - 99,5	74-76
	20 fps	98,5 - 100	74-76
	30 fps	98,5 - 100	76
640 x 320	10 fps	98,5 - 100	76
	20 fps	98,5 - 100	76
	30 fps	98,5 - 100	77
1280 x 720	10 fps	98,5 - 100	78-79
	20 fps	98,5 - 100	78-79
	30 fps	98,5 - 100	78-79
2560 x 1440	10 fps	98,5 - 100	85-87
	20 fps	98,5 - 100	85-87
	30 fps	98,5 - 100	85-87
<i>idle</i>		15,5 - 18,7	74-76

Dari pengujian di atas didapatkan ketika *video streaming* dijalankan Raspberry Pi mengonsumsi CPU kurang lebih 98,5% dan konsumsi *memory* yang digunakan adalah 74 sampai 87 MB.

Dari percobaan di atas didapatkan bahwa saat video streaming dijalankan membutuhkan sumber daya yang sangat besar mencapai 98% - 100%. Penggunaan *memory* meningkat seiring dengan besarnya resolusi yang digunakan, karena resolusi yang besar membutuhkan *memory* yang besar pula.

4.2.2 Pengujian Aplikasi

a. Skenario Pengujian

Pengujian ini dilakukan secara *alpha*. Metoda yang digunakan dalam pengujian *alpha* adalah metode *Black Box testing*. Metoda ini yang berfokus pada fungsionalitas sistem terutama hasil inputan dan keluaran yang dilakukan oleh pengguna. Pengujian dikatakan berhasil jika *output* sesuai dengan kendali input yang dimasukkan pada tiap tampilan. Pengujian ini dilakukan pada spesifikasi:

1. Login

2. Video Terekam
3. Aktivasi Program Alert
4. Nomer Handphone Tujuan

b. Hasil Pengujian

Berdasarkan hasil dari pengujian *blackbox* yang telah dilakukan, menunjukkan bahwa secara fungsional aplikasi yang telah dibangun sudah dapat menghasilkan keluaran yang diharapkan

5. Saran dan Kesimpulan

5.1 Saran

5.1 Kesimpulan

Dari pengujian yang telah dilakukan, maka dapat ditarik kesimpulan sebagai berikut:

1. Berdasarkan pengujian Black Box aplikasi dapat mendukung fitur-fitur untuk sistem keamanan ruangan.
2. Dari hasil pengujian video streaming, didapatkan bahwa resolusi optimum untuk video streaming adalah 320 x 180 sampai 1280 x 720, sedangkan pada resolusi 2560 x 1440 performa Raspberry Pi menurun drastis, hal ini dapat dilihat dari delay yang meningkat hingga 4 kali lipat dari resolusi sebelumnya dan jumlah frame rate yang dihasilkan menurun menjadi 5 fps meskipun frame rate diset pada 30 fps.
3. Saat pengguna mengakses aplikasi dan video streaming dijalankan meningkatkan rata-rata kinerja CPU 99% dan penggunaan RAM 76 - 86 MegaBytes.

5.2 Saran

Dari aplikasi yang telah dibangun tentunya masih perlu pengembangan agar aplikasi ini bisa lebih baik. Saran untuk melakukan pengembangan pada aplikasi ini yaitu sebagai berikut:

1. Untuk meningkatkan sistem keamanan dan kenyamanan pengguna hendaknya sistem ini menunjang fitur-fitur lain seperti alarm, konfigurasi *brightness*, *scheduling*, dan lain-lain.
2. Hardware yang digunakan hendaknya lebih ditingkatkan untuk mengurangi beban pada *server*

DAFTAR PUSTAKA

- [1] Rakhman, E., Candrasyah, F., & Sutera, F. D. (2014). *Raspberry Pi Mikrokontroler mungil yang serba bisa*. Bandung: Andi.
- [2] Grinberg, M. (2014). *Flask Web Development*. California: O'Reilly Media.
- [3] Hartono, Jogiyanto, 1999, Analisis dan Desain Sistem Informasi, Penerbit Andi, Yogyakarta.
- [4] Picamera Documentation (2015). <http://picamera.readthedocs.org>, Diakses 5 April 2015
- [5] Joel W King, (2009). Cisco IP Video Surveillance Design Guide.