

**SISTEM KAMERA PENGAWAS DENGAN MENGGUNAKAN
RASPBERRY PI DISERTAI *MOTION DETECTION* DAN
*AUTO BACKUP CLOUD (GOOGLE DRIVE)***



Disusun Oleh:

Nama : Ikhwan Alfath Nurul Fathony

NIM : 14523103

PROGRAM STUDI TEKNIK INFORMATIKA – PROGRAM SARJANA

FAKULTAS TEKNOLOGI INDUSTRI

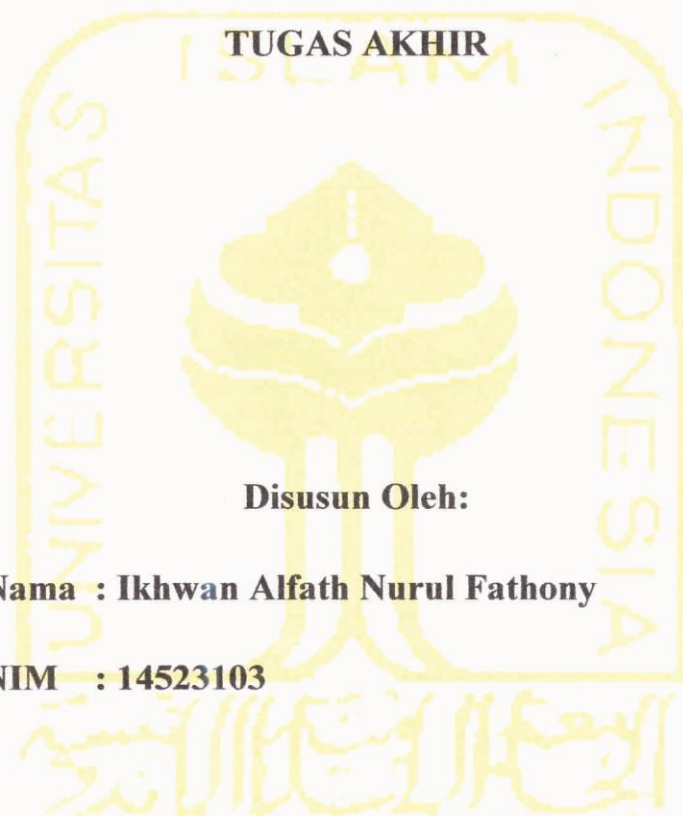
UNIVERSITAS ISLAM INDONESIA

YOGYAKARTA

2018

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**SISTEM KAMERA PENGAWAS DENGAN MENGGUNAKAN
RASPERRY PI DISERTAI MOTION DETECTION DAN
AUTO BACKUP CLOUD (GOOGLE DRIVE)**



Disusun Oleh:

Nama : Ikhwan Alfath Nurul Fathony

NIM : 14523103

Yogyakarta, 20 Juli 2018

Pembimbing

(Fietyata Yudha, S.Kom., M.Kom.)

HALAMAN PENGESAHAN DOSEN PENGUJI**SISTEM KAMERA PENGAWAS DENGAN MENGGUNAKAN
RASPERRY PI DISERTAI MOTION DETECTION DAN
AUTO BACKUP CLOUD (GOOGLE DRIVE)**

TUGAS AKHIR

Oleh:

Nama : Ikhwan Alfath Nurul Fathony

NIM : 14523103

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Teknik Informatika di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 20 Juli 2018

Tim Penguji

Fietyata Yudha, S.Kom., M.Kom.

Anggota 1

Mukhammad Andri Setiawan, S.T., M.Sc., Ph.D.

Anggota 2

Erika Ramadhani, S.T., M.Eng.

Mengetahui,

Ketua Program Studi Teknik Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia


(Dr. Raden Teduh Dirgahayu, S.T., M.Sc.)

HALAMAN PERNYATAAN KEASLIAN TUGAS

Yang bertanda tangan di bawah ini

Nama : Ikhwan Alfath Nurul Fathony

NIM : 14523103

Tugas Akhir dengan judul :

**SISTEM KAMERA PENGAWAS DENGAN MENGGUNAKAN
RASPBERRY PI DISERTAI *MOTION DETECTION* DAN
*AUTO BACKUP CLOUD (GOOGLE DRIVE)***

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung resiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 20 Juli 2018



(Ikhwan Alfath Nurul Fathony)

HALAMAN PERSEMBAHAN

Bismillahirrahmanirrahim

Saya persembahkan tugas akhir ini kepada bapak dan ibu saya yang selalu tiada henti mendoakan serta memberikan semangat dan dukungan yang tak henti – hentinya agar selalu dipermudahkan dalam segala urusan.

Untuk Sahabat – sahabat seperjuangan yang saya banggakan, sudah 4 tahun ini selalu bersama dalam menjalani masa kuliah. Terimakasih sahabat – sahabat seperjuangan lab informatika terpadu, semua teman teman UKM Merpati Putih, *Synaptic*, Robotika, teman – teman kelas B, serta teman – teman Magnifico 2014. Selalu jaga sportifitas dan tetap semangat semoga sukses untuk kita semua! Aamiin

HALAMAN MOTTO

Hati adalah raja seluruh organ lainnya, bak pasukan yang patuh dan setia menanti titahnya. Kemana organ tubuh hendak bergerak, dapat ditebak dari bagaimana kondisi hati.

KATA PENGANTAR

Assalamualaikum Wr. Wb.

Syukur Alhamdulillah segala rahmat yang telah diberikan oleh Allah SWT, sebab tiada hidayah yang lebih besar daripada hidayah yang telah diberikan oleh-Nya. Shalawat serta salam semoga tetap tercurah kepada junjungan kita Nabi Muhammad SAW beserta keluarga dan para sahabat. Sehingga atas ridho-Nya tugas akhir yang berjudul “*Sistem Kamera Pengawas dengan Menggunakan Raspberry Pi Disertai Motion Detection dan Auto Backup Cloud (Google Drive)*” dapat diselesaikan dengan baik.

Tugas Akhir ini disusun sebagai syarat terakhir yang harus ditempuh untuk menyelesaikan pendidikan pada jenjang Strata Satu (S1), pada Jurusan Informatika Universitas Islam Indonesia. Peneliti menyadari bahwa tanpa bimbingan, dorongan dan bantuan dari berbagai pihak tugas akhir ini tidak akan terwujud. Oleh karena itu dengan kerendahan hati peneliti mengucapkan terima kasih sebesar-besarnya kepada:

1. Bapak Fathul Wahid, S.T., M.Sc., Ph.D. selaku Rektor Universitas Islam Indonesia,
2. Bapak Prof. Dr. Ir. Hari Purnomo, M.T. selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia,
3. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc. selaku Ketua Program Studi Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia,
4. Bapak Fietyata Yudha, S.Kom., M.Kom. selaku pembimbing tugas akhir yang telah memberikan bimbingan dan pendampingan kepada penulis,
5. Bapak Chandra Kusuma Dewa, S.Kom., M.Cs. selaku dosen pembimbing akademik dan juga sebagai dosen pembimbing pelaksana tugas akhir yang telah membantu memberi arahan dan bimbingan, waktu dan ilmunya dan sangat membantu dalam membimbing skripsi,
6. Bapak dan ibu dosen Jurusan Informatika yang telah memberikan ilmunya kepada penulis, semoga bapak dan ibu dosen selalu dalam rahmat dan lindungan Allah SWT, sehingga ilmu yang telah diajarkan dapat bermanfaat di kemudian hari,
7. Ucapan terima kasih sebesar-besarnya kepada kedua orang tua yang tercinta, Bapak Mujiyono dan Ibu Susi Laningtyas Salindri dengan segala pengorbanannya

yang luar biasa, serta doa - doa yang tidak pernah putus untuk penulis serta nasihat dan petunjuk dari mereka yang menjadikan motivasi terbesar bagi penulis untuk dapat menyelesaikan tugas akhir ini.

8. Teman- teman seperjuangan yang telah memberikan dorongan dan semangat kepada penulis dalam menyelesaikan tugas akhir ini, terima kasih sahabat – sahabat seperjuangan lab informatika terpadu, semua teman teman UKM Merpati Putih, *Synaptic*, Robotika, teman – teman kelas B, serta teman – teman Magnifico, terimakasih atas dukungan yang telah diberikan.
9. Sahabat – sahabat kos: Mas Zulfikar, Mas Dhenta, Mas Bahar, Mas Laode. Semoga selepas lulus dari uii kita masih sering berkomunikasi, dan terimakasih telah menjadi keluarga kecil selama menempuh pendidikan di uii.
10. Sahabat – Sahabat KKN: Gilang, Fariz, Adit, Arie, Mardel, Kiki, Zahra. Terimakasih telah menjadi sahabat terbaik selama menjalani KKN, semoga tali persahabatan kita tetap terjaga hingga akhir hayat.
11. Sahabat – sahabat tim lomba Kaohsiung International Invention and Design EXPO (KIDE): Nastiti, Yudho, Nadhifah. Terimakasih telah memberikan dukungan, pengalaman, dan perjuangan hingga menumbuhkan semangat baru dan akhirnya penulis dapat menyelesaikan tugas akhir ini.

Tentunya sebagai manusia tidak pernah lepas dari kesalahan, sehingga dalam penyusunan tugas akhir ini masih banyak terdapat kesahalan dan kekurangan. Oleh karena itu penulis memohon maaf atas segala kesalahan dan kekurangan yang ada.

Wassalamualaikum Wr. Wb.

Yogyakarta, 20 Juli 2018

Ikhwan Alfath Nurul Fathony

SARI

Di era modern seperti saat ini sudah selayaknya masyarakat khususnya di perkotaan perlu berjaga-jaga dengan adanya tindak kriminalitas. Sebagaimana sudah banyak terjadi di lingkungan masyarakat utamanya perkotaan sangat rawan akan tindakan pencurian. Pemilik rumah ataupun toko sangat perlu berwaspada dengan hal ini. Untuk itu masyarakat perlu berwaspada dengan memasang sebuah perangkat kamera pengawas, sehingga tindakan kriminalitas bisa segera diketahui pemiliknya, sehingga dapat meminimalisir tindak kriminalitas yang ada di perkotaan.

Dengan memasang perangkat kamera pengawas saja ternyata belum cukup mengamankan lingkungan perkotaan. Sebab pencuri sekarang telah mengikuti perkembangan perangkat kamera pengawas, dimana mesin ataupun perangkat kamera pengawas ikut serta menjadi sasaran dari para pencuri. Sedangkan hasil rekam kamera tersebut dapat dijadikan barang bukti sebagai tindak kriminalitas di persidangan.

Oleh karenanya penulis membuat sebuah sistem kamera pengawas dengan *ipcamera*, *webcam* serta menggunakan *mini computer (Raspberry Pi)* yang bertujuan untuk mengamankan barang bukti berupa gambar rekaman dengan cara sistem kamera rekam tersebut menangkap gambar ketika terdapat pergerakan, kemudian hasil tangkapan gambar tersebut secara otomatis terupload ke cloud (disini menggunakan *Google Drive*). Sehingga dengan adanya sistem ini, maka akan lebih mengamankan barang bukti apabila mesin daripada kamera rekam telah ikut dicuri ataupun dirusak oleh pelaku pencurian.

Kata Kunci :

Kamera, Pengawas, *Raspberry Pi*, Sistem, *Cloud*, *Google Drive*

GLOSARIUM

<i>IP</i>	<i>Internet Protokol (IP)</i> merupakan deretan angka biner yang dipakai sebagai alamat identifikasi untuk tiap host dalam jaringan internet.
<i>ipcamera</i>	jenis kamera video digital yang dapat digunakan untuk pemantauan keamanan dan dapat mengirim dan menerima data melalui jaringan komputer dan internet, sehingga kamera ini dapat diakses dengan alamat ip dan menggunakan <i>port RJ45</i> .
<i>Webcam</i>	jenis kamera yang dapat digunakan untuk mengambil gambar ataupun citra dan beberapa jenis kamera ini dapat digunakan untuk merekam suara dengan bantuan perangkat komputer dan dengan menggunakan port <i>Universal Serial Bus (USB)</i> .
<i>Cloud</i>	gabungan pemanfaatan teknologi computer dan pengembangan berbasis internet.
<i>Google Drive</i>	salah satu layanan yang ditawarkan oleh <i>Google</i> dan merupakan layanan untuk menyimpan dokumen atau file secara gratis maupun berbayar tergantung pada jumlah kapasitas memori penyimpanan itu sendiri.
<i>Motion Detection</i>	sebuah fitur video analisis yang berfungsi untuk mendeteksi adanya pergerakan (orang, mobil, dan lain-lain).
<i>Auto Backup</i>	merupakan proses yang dapat dilakukan oleh sistem secara otomatis membuat pencadangan data pada lokasi penyimpanan lain.
<i>Raspberry Pi</i>	perangkat komputer kecil yang dapat digunakan untuk menjalankan program layaknya PC desktop, namun dengan sumber daya yang terbatas.
<i>Css</i>	cascading style sheet, kode pemrograman untuk mengatur gaya tampilan <i>web</i>
<i>Hardware</i>	merupakan semua bagian fisik komputer, dan dibedakan dengan data yang berada di dalamnya atau yang beroperasi di dalamnya, dan dibedakan dengan perangkat lunak (<i>software</i>) yang menyediakan instruksi untuk perangkat keras dalam menyelesaikan tugasnya.

<i>Sistem</i>	merupakan suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi untuk mencapai suatu tujuan.
<i>Software</i>	secara umum dapat diartikan sebagai sekumpulan data-data elektronik yang tersimpan dan diatur oleh komputer yang berupa program atau instruksi untuk menjalankan dan mengeksekusi suatu perintah
<i>API</i>	(<i>Application Programming Interface</i>) merupakan sekumpulan perintah, fungsi, serta protokol yang dapat digunakan oleh programmer saat membangun perangkat lunak untuk sistem operasi tertentu.

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING.....	Error! Bookmark not defined.
HALAMAN PENGESAHAN DOSEN PENGUJI.....	Error! Bookmark not defined.
HALAMAN PERNYATAAN KEASLIAN TUGAS.....	Error! Bookmark not defined.
HALAMAN PERSEMBAHAN	v
HALAMAN MOTTO	vi
KATA PENGANTAR	vii
SARI	ix
GLOSARIUM.....	x
DAFTAR ISI.....	xii
DAFTAR GAMBAR	xv
DAFTAR TABEL.....	xvii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah	3
1.3. Batasan Masalah.....	3
1.4. Tujuan Penelitian.....	3
1.5. Manfaat Penelitian.....	4
1.6. Metodologi Penelitian	4
1.7. Sistematika Penulisan.....	4
BAB II LANDASAN TEORI.....	5
2.1 Metode pengembangan sistem atau perangkat lunak	5
2.2 <i>Waterfall</i>	5
2.2.1 Requirement Analisis.....	5
2.2.2 <i>System Design</i>	6
2.2.3 <i>Implementation</i>	6
2.2.4 <i>Integration and Testing</i>	6
2.2.5 <i>Operation and Maintenance</i>	6

2.3	<i>Usability</i>	7
2.3.1.	Pengukuran <i>Usability</i>	7
2.3.2.	Komponen <i>Usability</i>	8
2.4	Fungsionalitas Perangkat Lunak (<i>Software Functionality</i>)	9
2.4.1.	Aspek Fungsionalitas	9
2.4.2.	Pengujian Fungsionalitas	9
2.5	<i>Kamera Pengawas</i>	10
2.5.1	<i>Kamera Analog</i>	10
2.5.2	<i>IP Camera</i>	10
2.6	<i>Raspberry Pi</i>	11
2.7	<i>Cloud / Google Drive</i>	11
BAB III	METODOLOGI.....	12
3.1.	Tahapan Alur Penelitian	12
3.2.	Kebutuhan Sistem.....	12
3.3.	Perancangan Sistem.....	13
3.3.1.	Perancangan Hardware	13
3.3.2.	Perancangan Software.....	14
3.3.3.	Gabungan Perancangan Hardware dengan Software	15
3.4.	Software.....	15
3.4.1.	<i>Data Flow Diagram (DFD)</i>	15
3.4.2.	<i>Use Case</i>	17
3.4.3.	<i>Identifikasi Use Case Diagram</i>	18
3.4.4.	<i>Skenario Use Case Diagram</i>	19
3.4.5.	<i>Activity Diagram</i>	20
3.4.6.	<i>Perancangan Basis Data</i>	27
3.4.7.	<i>Perancangan Antarmuka</i>	27
3.4.8.	<i>Class Diagram</i>	30

3.4.9. <i>Pseudocode</i>	31
3.4.10. <i>Pengujian</i>	36
BAB IV HASIL DAN PEMBAHASAN	38
4.1. Hasil.....	38
4.1.1. Hasil Pembuatan Perangkat Keras (Hardware).....	38
4.1.2. Hasil Pembuatan Perangkat Lunak (Software)	40
4.2. Pembahasan Implementasi Penelitian	43
4.2.1. Pembahasan Implementasi Perangkat Keras (Hardware).....	43
4.2.2. Pembahasan Implementasi Perangkat Lunak.....	44
4.2.3. Pembahasan Modul Pada Perangkat Lunak	49
4.3. Pengujian Penelitian	52
4.4. Hasil Pengujian Sistem.....	52
4.4.1. Perhitungan Skala Likert.....	52
4.4.2. Hasil Total Skala Likert	53
4.5. Analisis Penelitian	55
BAB V KESIMPULAN DAN SARAN	56
5.1 Kesimpulan.....	56
5.2 Saran	56
DAFTAR PUSTAKA	57
LAMPIRAN.....	59

DAFTAR GAMBAR

Gambar 1.1 Data Statistik Kejahatan di Indonesia	2
Gambar 3.2 Tahapan Analisa Kebutuhan Sistem	12
Gambar 3.2 Skema Perangkat yang terhubung dengan Raspi	13
Gambar 3.3 Skema Sistem Pada Raspberry pi.....	14
Gambar 3.4 Skema Perangkat dan Sistem yang saling terhubung.....	15
Gambar 3.5 DFD Level 0.....	16
Gambar 3.6 DFD Level 1.....	16
Gambar 3.7 Use Case Diagram sistem kamera pengawas	17
Gambar 3.8 Activity Diagram Login Admin pada Sistem	21
Gambar 3.9 Activity Diagram Unduh Data Rekam pada Sistem	22
Gambar 3.10 Activity Diagram Melihat Data Rekam pada Sistem.....	23
Gambar 3.11 Activity Diagram Menghapus Data Rekam pada Sistem.....	24
Gambar 3.12 Activity Diagram Melihat Pesan.....	25
Gambar 3.13 Activity Diagram Pengguna Melihat Streaming Kamera	26
Gambar 3.14 Activity Diagram Pengguna Mengirim pesan.....	26
Gambar 3.15 Struktur Tabel Basis Data pada Sistem Kamera Pengawas	28
Gambar 3.16 Rancangan Halaman Login.....	28
Gambar 3.17 Rancangan Halaman Utama Admin.....	29
Gambar 3.18 Rancangan Halaman Streaming Admin.....	29
Gambar 3.19 Rancangan Halaman Rekaman Admin	30
Gambar 3.20 Rancangan Halaman Gambar Admin	30
Gambar 3.21 Rancangan Halaman Pengaturan Admin	31
Gambar 3.22 Rancangan Halaman Pesan Admin	31
Gambar 3.23 <i>Class Diagram</i>	31
Gambar 3.24 <i>Pseudocode</i> kamera.py.....	32
Gambar 3.25 <i>Pseudocode</i> kamera_ip.py	33
Gambar 3.26 <i>Pseudocode</i> upload.py	34
Gambar 3.27 <i>Pseudocode</i> main.py	36
Gambar 4.1 Perangkat Raspberry Pi.....	39
Gambar 4.2 Rangkaian komunikasi kamera usb dan ip kamera.....	40
Gambar 4.3 Rangkaian perangkat keseluruhan	40

Gambar 4.4 Halaman Login Admin.....	41
Gambar 4.5 Dashboard Halaman Admin.....	41
Gambar 4.6 Dashboard Halaman Streaming	42
Gambar 4.7 Dashboard Halaman Gambar	42
Gambar 4.8 Dashboard Halaman Pengaturan	43
Gambar 4.9 Dashboard Halaman Pesan.....	43
Gambar 4.10 Kode Prog. akses kamera webcam disertai Motion Detection.....	46
Gambar 4.11 Kode Program akses ip kamera dengan hasil format .avi	47
Gambar 4.12 Kunci rahasia untuk akses Google Drive Api.....	48
Gambar 4.13 Kode Program akses Google Drive Api.....	49
Gambar 4.14 Kode Program akses streaming video rekaman	50

DAFTAR TABEL

Tabel 3.1 Identifikasi Use Case Diagram	18
Tabel 3.2 Pengujian	37
Tabel 4.1 Tabel Prosentase	54
Tabel 4.2 Tabel Pengujian Skala Likert.....	54

BAB I

PENDAHULUAN

1.1. Latar Belakang

Di era modern seperti saat ini sudah selayaknya masyarakat khususnya di perkotaan perlu berjaga-jaga dengan adanya tindak kriminalitas. Sebagaimana sudah banyak terjadi di lingkungan masyarakat utamanya perkotaan yang padat akan penduduk. Suatu keamanan rumah untuk saat ini menjadi hal yang sangat penting yang mana selain tindak kriminalitas yang besar juga mobilitas dari pemilik rumah yang sangat tinggi menjadikan keamanan ruangan atau rumah di saat kita sedang tidak di rumah. Banyak upaya yang dilakukan oleh pemilik rumah guna mengamankan ruangan atau rumah salah satunya dengan kunci pengaman pada pintu rumah atau ruangan. Namun hal ini belum cukup untuk mencegah tindak kriminalitas. Untuk itu masyarakat perlu lebih berhati-hati dengan memasang sebuah perangkat kamera pengawas, sehingga apabila terdapat tindakan yang mencurigakan, pemilik rumah dapat segera mengetahuinya.

Dengan memasang perangkat kamera pengawas saja masih belum cukup mengamankan lingkungan perkotaan. Sebab pencuri sekarang telah mengikuti perkembangan perangkat kamera pengawas, dimana mesin ataupun perangkat kamera pengawas ikut serta menjadi sasaran daripada pencuri. Sedangkan hasil rekam kamera tersebut dapat dijadikan sebagai barang bukti tindak kriminalitas di persidangan seperti yang tertera pada undang undang (ITE).

Undang – undang nomor 11 tahun 2008 tentang UU ITE yang merupakan pedoman hukum cyber di Indonesia ternyata tidak mencantumkan penjelasan tentang bukti digital ini. Namun terdapat dua istilah yang mirip dengan bukti digital ini, yaitu informasi elektronik dan dokumen elektronik.

Pasal 1 butir 1 UU ITE disebutkan bahwa informasi elektronik adalah satu atau sekumpulan data elektronik, termasuk tetapi tidak terbatas pada tulisan, suara, gambar, peta, rancangan, foto, electronic data interchange, surat elektronik (*electronic mail*), *telegram*, *teleks*, *telecopy* atau sejenisnya, huruf, tanda, angka, kode akses, simbol, atau perforasi yang telah diolah yang memiliki arti atau dapat dipahami oleh orang yang mampu memahaminya.

Sedangkan dalam pasal 1 butir 4 UU ITE menjelaskan bahwa dokumen elektronik adalah setiap Informasi Elektronik yang dibuat, diteruskan, dikirimkan, diterima, atau disimpan dalam bentuk analog, digital, elektromagnetik, optikal, atau sejenisnya, yang dapat

dilihat, ditampilkan, dan/atau didengar melalui Komputer atau Sistem Elektronik, termasuk tetapi tidak terbatas pada tulisan, suara, gambar, peta, rancangan, foto atau sejenisnya, huruf, tanda, angka, kode akses, simbol atau perforasi yang memiliki makna atau arti atau dapat dipahami oleh orang yang mampu memahaminya.

Berdasarkan penjelasan tersebut maka informasi dan dokumen elektronik tersebut dapat dibedakan namun tidak dapat terpisahkan. Selain itu informasi elektronik merupakan data atau sekumpulan data sedangkan dokumen elektronik merupakan tempat atau wadah dari informasi elektronik tersebut. Sebagai contoh sebuah gambar berformat .jpg, maka isi dari foto tersebut merupakan informasi elektronik, sedangkan file .jpg merupakan dokumen elektroniknya.

Oleh karenanya penulis membuat sebuah sistem kamera pengawas dengan berbekal perangkat sederhana yaitu *Raspberry Pi* yang terhubung dengan kamera *webcam* dan *ipcamera* untuk dapat menangkap gambar ketika terdapat pergerakan benda pada suatu ruangan ataupun rumah. Kemudian hasil rekaman gambar tersebut dengan berbekal *raspberry pi* yang terkoneksi ke internet akan di *upload* ke dalam *cloud* (di sini menggunakan *Google Drive*) secara otomatis, sehingga dapat dipantau jarak jauh dan hal ini dapat lebih mengamankan barang bukti gambar rekaman bilamana terjadi pencurian disertai pengrusakan perangkat kamera pengawas.

Negara Indonesia sendiri terdapat beberapa karakteristik kejahatan berdasarkan etnis atau suku bangsa. Untuk lebih jelasnya maka penulis berikan data statistik kejahatan di Indonesia sepanjang tahun 2010 hingga 2015 berikut ini (Somba, 2018).



Gambar 1.1 Data Statistik Kejahatan di Indonesia

1.2. Rumusan Masalah

- a. Apakah sistem kamera pengawas dapat mengakses kamera *webcam* dan *ipcamera*?
- b. Bagaimana kamera pengawas dapat menangkap gambar apabila terdapat pergerakan?
- c. Bagaimana hasil gambar dapat di *backup* pada *cloud storage* (*Google Drive*)?

1.3. Batasan Masalah

Dalam penyusunan tugas akhir ini, penyusun mempunyai batasan masalah sebagai berikut.

- a. *Raspberry pi* yang digunakan yaitu *Raspberry pi 3* model B.
- b. Memori internal yang digunakan yaitu sebesar 32 gb.
- c. Menggunakan satu kamera *webcam* dan satu *ipcamera*.
- d. Hasil rekaman pada kamera web berupa video format .avi dan berupa gambar .jpg yang akan menyimpan gambar ketika terdapat pergerakan.
- e. Hasil rekaman pada kamera ip berupa video format .avi yang akan disimpan pada memory raspberry pi.
- f. Bahasa pemrograman yang digunakan yaitu Python 2.7, PHP, Mysql, JSON, JS
- g. Data yang di-*upload* ke *Google Drive* secara otomatis merupakan data gambar .jpg

1.4. Tujuan Penelitian

- a. Sistem dapat mengakses kamera usb (*webcam*) ataupun dengan kamera *Internet Protocol* (*IP*) dengan menggunakan komputer kecil (*raspberry pi*) yang di dalamnya terinstall sistem operasi *linux raspbian stretch*.
- b. Sistem dapat menangkap gambar ketika terdapat pergerakan atau biasa disebut *motion detection*, hal ini diterapkan guna mendapatkan data gambar apabila terdapat pergerakan yang terdeteksi oleh kamera, sehingga akan lebih menghemat penyimpanan data.
- c. Dalam sistem penulis menambahkan fitur auto backup dengan tujuan untuk melakukan pencadangan data secara otomatis ke dalam *cloud* (*Google Drive*). Dengan adanya *motion detection* yang berarti dapat mengambil gambar dengan format .jpg secara otomatis, guna menghemat jalur data internet (*bandwidth*). Walau begini, sistem ini tetap tidak menghilangkan tujuan daripada sistem kamera pengawas yang mana tetap mengambil data rekam berupa video.

1.5. Manfaat Penelitian

Manfaat penelitian ini yaitu seseorang dapat memanfaatkan sistem kamera pengawas dengan perangkat raspberry pi, wecam dan ipcam yang dapat digunakan sesuai dengan tujuan yang diharapkan.

1.6. Metodologi Penelitian

- a. Studi pustaka metode *studi kasus*.
- b. Pemilihan perangkat dan program yang akan digunakan.
- c. Penentuan skenario sistem kamera pengawas.
- d. Bangun aplikasi web dengan kode program php, mysql, css
- e. Penyajian hasil sistem kamera pengawas.

1.7. Sistematika Penulisan

Dalam penulisan penelitian ini tersusun atas beberapa bab, sistematika penulisan adalah sebagai berikut:

Bab I Pendahuluan, berisi tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian dan sistematika penulisan.

Bab II Landasan Teori, memuat tentang penjelasan teori seperti pengertian dari *studi kasus*, *raspberry pi*, *webcam*, *ipcam*, serta parameternya.

Bab III Metodologi, memuat tentang rancangan sistem.

Bab IV Hasil dan Pembahasan, memuat tentang hasil implementasi dan hasil sistem yang telah dibuat.

Bab V Kesimpulan dan saran, berisi kesimpulan penelitian yang merupakan penjelasan hasil yang didapatkan dari penelitian. Saran akan memberikan masukan atau pendapat untuk penelitian selanjutnya agar lebih baik lagi.

BAB II

LANDASAN TEORI

2.1 Metode pengembangan sistem atau perangkat lunak

Metodologi pengembangan perangkat lunak (atau disebut juga model proses rekayasa perangkat lunak) merupakan salah satu strategi yang digunakan untuk memadukan proses, metode, dan perangkat (*tools*). Metode - metode rekayasa perangkat lunak dapat memberikan teknik untuk membangun suatu sistem perangkat lunak. Berkaitan dengan serangkaian tugas yang luas yang menyangkut analisis kebutuhan, konstruksi program, desain, pengujian, dan pemeliharaan (Pressman, 2002). Menurut Roger S Pressman Metode – metode tersebut adalah:

- a. Metode waterfall
- b. Model skuensial linier
- c. Model *Prototype*
- d. Model RAD

Waterfall merupakan salah satu model pengembangan software yang diperkenalkan oleh Winston Royce pada tahun 70-an dan merupakan model klasik yang sederhana dengan aliran sistem yang linier. Keluaran dari tahap sebelumnya merupakan masukan untuk tahapan berikutnya. Pengembangan dengan model ini merupakan hasil adaptasi dari pengembangan perangkat keras, karena pada saat itu belum ada metodologi perangkat lunak lainnya (Pressman, 2002).

2.2 Waterfall

Model Waterfall merupakan suatu proses pengembangan perangkat lunak berurutan, dimana kemajuan dipandang untuk terus mengalir ke bawah atau seperti air terjun, melewati fase – fase perencanaan, pemodelan, implementasi (konstruksi), hasil, dan pengujian.

2.2.1 Requirement Analisis

Tahapan ini pengembang sistem perlu berkomunikasi yang bertujuan untuk memahami perangkat apa saja yang harus digunakan serta memberikan batasan penggunaan perangkat yang dibutuhkan. Informasi ini bisa didapatkan melalui wawancara atau survey langsung. Informasi dianalisis untuk mendapatkan data yang dibutuhkan sesuai dengan keadaan di lapangan.

2.2.2 System Design

Spesifikasi kebutuhan dan tahapan sebelumnya akan dipelajari dalam fase ini dan menyiapkan rancangan desain sistem. Desain sistem membantu dalam menentukan kebutuhan perangkat keras (*hardware*) dan sistem persyaratan yang juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

2.2.3 Implementation

Pada tahap ini, sistem pertama kali dikembangkan pada program kecil yang disebut dengan unit, yang terintegrasi dalam tahapan selanjutnya. Setiap unit dikembangkan dan diuji untuk fungsionalitas yang sebut sebagai unit testing.

a. Representasi

Bentuk dari sebuah *system* dapat kita tentukan apakah hasilnya nanti akan dipresentasikan dalam sebuah tampilan web atau dilengkapi dengan tampilan visual perangkat mobile.

b. Jangkauan

System dapat berbentuk tampilan antar muka (*interface*) secara umum dapat digunakan dalam beberapa perangkat ataupun terbatas hanya pada perangkat computer saja.

c. Eksekutabilitas

Pada dimensi ini, *system* dengan kategori kompleks dapat berupa sebuah *program* “setengah jadi” yang benar – benar dapat dieksekusi sehingga terbentuk aplikasi yang dapat dioperasikan oleh *user* dan seakan – akan nyata bagi *user*.

d. Pematangan

Sebuah *system* dapat melalui tahap – tahap tertentu sebelum sampai pada tahapan yang dianggap “matang” sebelum memulai proses implementasi *system*.

2.2.4 Integration and Testing

Seluruh *unit* yang dikembangkan dalam tahap implementasi kemudian diintegrasikan ke dalam sistem setelah pengujian yang dilakukan pada masing-masing *unit*. Setelah integrasi seluruh sistem diuji untuk dilakukan pengecekan pada setiap kegagalan maupun kesalahan.

2.2.5 Operation and Maintenance

Tahap terakhir dalam model *waterfall* yaitu perangkat lunak yang sudah jadi, dijalankan kemudian dilakukan pemeliharaan secara berkala. Pemeliharaan di sini termasuk

dalam memperbaiki kesalahan yang belum ditemukan pada langkah sebelumnya. Perbaikan implementasi *unit* sistem dan peningkatan jasa sistem sebagai kebutuhan baru.

2.3 Usability

Usability adalah pengoptimasian penggunaan sistem oleh pengguna, sebuah *system* yang baik dapat dipergunakan secara maksimal oleh pengguna sehingga kemampuan *system* dapat dimanfaatkan. Selain itu *usability* merupakan sebuah attribute kualitas yang dapat menilai tingkat kemudahan *user interface* dari sebuah *system* untuk digunakan (Nielsen, 1993).

2.3.1. Pengukuran Usability

Dalam melakukan pengukuran *usability* pada sebuah perangkat lunak dapat dilakukan dengan dua cara (Falahah, 2012), yaitu:

- a. Mengandalkan sebuah asumsi berdasarkan pembuat program atau diri sendiri.
- b. Pengukuran berdasarkan *usability* matriks.

Hasil yang diperoleh dari pengukuran *usability* ini dapat dimanfaatkan untuk beberapa hal diantaranya (Tullis, Tom, & Albert, 2008).

- a. Mendapat masukan dari data, lebih obyektif daripada diri sendiri.
- b. Dapat digunakan untuk membandingkan *usability* dua produk.
- c. Dapat mengklasifikasi permasalahan (jika ada).
- d. Membuat prediksi penggunaan produk yang sebenarnya.
- e. Memberikan ilustrasi pada manajemen berdasarkan fakta.

Pada matriks *usability* sendiri saat ini terdapat beberapa jenis matrik atau teknik pengukuran *usability*, yang secara umum dapat dibagi menjadi dua kategori, yaitu (Falahah, 2012) :

- a. ***Desired quality***: merupakan sebuah pengamatan berupa ukuran selesai atau tidaknya suatu tugas (*yes/no*), tercapai tidaknya suatu hasil yang ada, atau diterima tidaknya suatu pernyataan (*agree/disagree*).
- b. **Pengukuran kuantitatif**: mengukur dalam sebuah skala tertentu, misalnya X% *user* dapat menyelesaikan tugasnya tepat waktu atau bisa kurang dari satu menit.

Pada penelitian ini matrik pengukuran yang digunakan adalah *desired quality* yang sesuai dengan bentuk dan jenis penelitian yang digunakan.

2.3.2. Komponen *Usability*

Usability juga memiliki beberapa komponen, di antara komponen tersebut yaitu (Nielsen, 1993) :

a. *Learnability*

Menjelaskan tentang tingkat kemudahan *user* atau pengguna untuk memenuhi tugas dasar ketika *user* pertama kali melihat dan menggunakan *system*. Pada *learnability* terdapat juga matrik ukuran yang dapat digunakan sebagai acuan pada saat dilakukan pengujian *learnability*, yaitu (ISO/IEC 9126 - 2) :

1. ***Ease of function learning***: melihat tingkat kemudahan *user* dalam mempelajari fungsi perangkat lunak.
2. ***Understandability***: melihat kemampuan perangkat lunak untuk cepat dipahami oleh *user*.
3. ***Efectiveness of user documentation***: melihat keefektifan dokumentasi penggunaan yang diberikan perangkat lunak untuk *user*.

Sedangkan sebelum memulai pengujian *learnability* diperlukan penjelasan tentang kemampuan user berdasarkan pada poin – poin yang ada pada *user definition* (Grossman, George, Attar, & Ramtin, 2009).

1. ***Level of experience with computers***

Karena penilaian berdasarkan dari sudut pandang *user* jadi dirasa perlu menjelaskan tingkat kepemahamannya mengenai pengoperasian komputer terlebih dahulu.

2. ***Level of experience with interface***

Kemudian menjelaskan bagaimana pemahaman *user* terkait dengan *interface* dari perangkat lunak selama penggunaan.

3. ***Quality of domain knowledge***

Setelah itu menjelaskan tentang pengetahuan *user* mengenal domain permasalahan, yaitu tentu saja mengenai manfaat dan kegunaan dari aplikasi yang dibahas.

4. ***Experience with similiar software***

Terakhir menjelaskan tentang perangkat lunak yang serupa dan pernah digunakan oleh *user* diluar perangkat lunak yang sedang digunakan sebagai penelitian ini.

b. *Efficiency*

Kemampuan perangkat lunak untuk memberikan kinerja yang sesuai dan relatif terhadap melayani kebutuhan pengguna serta jumlah sumber daya yang diperlukan pada saat keadaan tersebut. Untuk matrik pengukuran yang dapat digunakan sebagai acuan pada saat melakukan pengujian (P Botella, 2004), yaitu:

1. **Time behavior:** kemampuan perangkat lunak dalam memberikan respon dan waktu pengolahan yang sesuai dengan fungsinya masing - masing
2. **Resource behavior:** kemampuan perangkat lunak dalam menggunakan sumber daya yang dimilikinya untuk melakukan fungsi yang ditentukan.

c. *Memorability*

Menjelaskan tentang tingkat kemudahan pengguna dalam menggunakan sebuah *system* dengan baik, apakah sistem tersebut mudah dioperasikan dan diingat oleh pengguna dilihat dari segi fitur atau menu – menu yang ada.

d. *Errors*

Kemampuan perangkat lunak dalam memberikan sebuah koreksi atau peringatan kepada pengguna terhadap kondisi dan situasi yang tidak diinginkan ketika sedang mengoperasikan sistem.

2.4 Fungsionalitas Perangkat Lunak (*Software Functionality*)

Fungsionalitas perangkat lunak yaitu kemampuan perangkat lunak untuk menyediakan fungsi – fungsi yang dapat memenuhi kebutuhan *user* dalam menyelesaikan tugasnya. Fungsionalitas merupakan *attribute – attribute* yang menjaga suatu fungsi dan sifat spesifik suatu perangkat lunak untuk memenuhi dan mengandung suatu kebutuhan. Fungsionalitas dapat dinilai berdasarkan hasil evaluasi fitur dan kemampuan dari perangkat lunak, serta sifat umum dari fungsi yang ada secara keseluruhan (Khosravi, 2004).

2.4.1. Aspek Fungsionalitas

Aspek fungsionalitas yaitu kemampuan perangkat lunak berfokus pada kesesuaian satu set fungsi untuk dapat melakukan tugas – tugas tertentu atau fungsi utama (Zrymiak, 2001). Aspek fungsionalitas berfokus pada ketepatan hasil keluaran (*output*) yang sesuai dengan maksud dan fungsi dari suatu perangkat lunak. Selain itu, perangkat lunak memiliki kemampuan untuk dioperasikan diberbagai macam *system*, seperti suatu aplikasi harus memiliki kemampuan untuk dijalankan oleh pengguna pada sistem operasi yang berbeda tanpa mengurangi fungsi yang ada (Ottensosser, 1999).

2.4.2. Pengujian Fungsionalitas

Merupakan suatu metode pengujian yang berfokus pada persyaratan fungsional perangkat lunak. Pengujian fungsional sering dikenal juga dengan istilah *black box testing* yang merupakan pengujian yang mengabaikan mekanisme internal suatu perangkat lunak dan

berfokus hanya pada *output* yang dihasilkan dalam menanggapi *input* dan kondisi eksekusi yang dipilih (Beizer, 1995).

Pada saat melakukan pengujian *functionality* (fungsionalitas) perangkat lunak terdapat beberapa matrik ukuran yang harus diperhatikan selama pengujian, yaitu (Botella *et al*, 2004):

- a. ***Suitability***: melihat kemampuan perangkat lunak dalam menyediakan serangkaian fungsi yang sesuai untuk tugas – tugas tertentu untuk pengguna.
- b. ***Accuracy***: Kemampuan perangkat lunak dalam memberikan hasil yang presisi dan benar sesuai dengan kebutuhan.

Kemudian kedua matrik ukuran tersebut digunakan untuk menilai kesesuaian karakteristik dari *system* dengan melihat fungsi atau fitur yang disediakan pada pembuatan *system*, jika salah satu tidak terpenuhi maka akan dianggap tidak sesuai.

2.5 ***Kamera Pengawas***

Perangkat *kamera pengawas* sudah menjadi tren masyarakat akhir – akhir ini. Kamera pengawas yang awalnya hanya diminati oleh beberapa yang memiliki usaha, pada kenyataannya sekarang juga mulai diminati oleh perseorangan serta keluarga. Dari mulai mengawasi aktivitas anak dirumah, mengawasi orangtua berumur lanjut, mengawasi aktifitas perniagaan, dll. Kamera pengawas untuk saat ini menjadi alat yang diandalkan membuat perlindungan beberapa orang yang disayangi tanpa dibatasi oleh ruangan. Terdapat beberapa macam kamera pengawas yang dapat digunakan dan dapat terhubung dengan *raspberry pi*, yaitu (Warsito, 2015)

2.5.1 ***Kamera Analog***

Kamera analog dapat diakses dari jarak jauh dengan bantuan perangkat tambahan atau perangkat yang dapat mengolah gambar rekaman serta dapat terkoneksi dengan jaringan internet. Disini penulis menggunakan perangkat kamera webcam dan ipcam untuk dapat mengambil gambar rekaman serta terhubung dengan perangkat Raspberry pi sebagai pusat manajemen atau pengolahan data hasil rekamannya.

2.5.2 ***IP Camera***

Alamat IP (*Internet Protocol Address* atau sering disingkat sebagai *IP*) adalah deretan angka biner antara 32 bit sampai 128 bit yang dipakai sebagai alamat identifikasi untuk tiap host dalam jaringan internet. Jadi *IP Camera* atau kamera ip merupakan sebuah kamera yang

dapat diakses langsung dengan menggunakan alamat ip. Jadi pada dasarnya kamera ip sudah dapat langsung di akses tanpa menggunakan perangkat tambahan sebagai contoh di sini perangkat Raspberry Pi. Kamera ip dapat langsung di akses dari jarak jauh tanpa perangkat pengolah data, yang penting kamera tersebut dapat terkoneksi dengan jaringan internet.

2.6 Raspberry Pi

Raspberry Pi, sering disingkat dengan nama *Raspi* adalah computer papan tunggal (*single-board circuit; SBC*) yang seukuran dengan kartu kredit dan dapat digunakan untuk menjalankan program perkantoran, permainan, computer, dan sebagai meutar media hingga video beresolusi tinggi. Raspberry Pi dikembangkan oleh yayasan nirlaba, Raspberry Pi Foundation yang di gawangi sejumlah pengembang dan ahli computer dari Universitas Cambridge, Inggris. Raspberry pi memiliki dua model yaitu model A dan model B. Secara umum Raspberry pi Model B memiliki kapasitas penyimpanan *RAM* sebesar 1 GB, sedangkan model A memiliki kapasitas penyimpanan *RAM* sebesar 512 MB. Selain itu, model B sudah dilengkapi dengan port *Ethernet* (untuk *LAN*) yang tidak terdapat pada model A. Penyimpanan data tidak didesain untuk menggunakan cakram keras atau solid state drive melainkan mengandalkan kartu penyimpanan tipe SD untuk menjalankan sistem dan sebagai media penyimpanan jangka panjang.

2.7 Cloud / Google Drive

Google Drive adalah layanan *cloud storage* dari *Google* yang diluncurkan pada akhir April 2012, yaitu layanan untuk menyimpan file di internet pada storage yang disediakan oleh Google. Dengan menyimpan file di *Google Drive* maka pemilik file dapat mengakses file tersebut kapanpun dimanapun dengan menggunakan komputer desktop, laptop, komputer tablet ataupun smartphone. Dan file tersebut dapat di share dengan orang lain untuk berbagi pakai dan juga kolaborasi pengeditannya. Kapasitas yang disediakan oleh *Google Drive* untuk layanan gratis adalah 15 GB, untuk menggunakan kapasitas lebih dari itu maka akan dikenakan biaya tambahan.

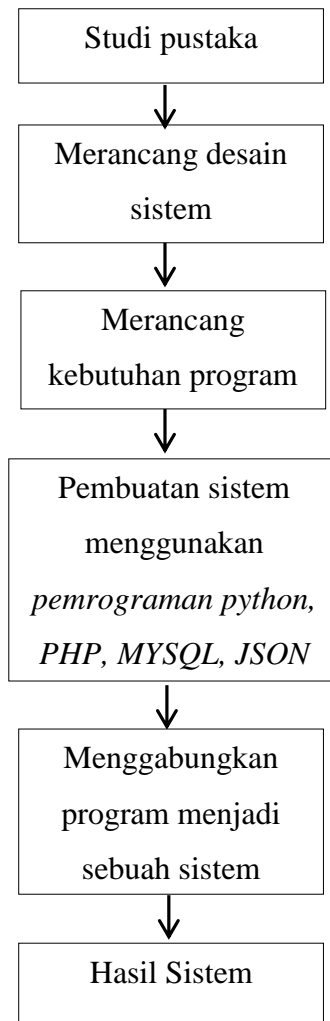
Dengan menggunakan Google Drive, berarti pemilik file telah memiliki backup filenya di internet, sehingga jika terjadi kerusakan perangkat maka data masih aman di internet.

BAB III

METODOLOGI

3.1. Tahapan Alur Penelitian

Dalam melakukan pembuatan perangkat lunak (*software*) ini ada beberapa tahapan yang dilakukan sebelum nantinya diperoleh hasil dari pembuatan sistem tersebut. Tahapan – tahapannya seperti pada gambar 3.1.



Gambar 3.1 Tahapan Analisa Kebutuhan Sistem

3.2. Kebutuhan Sistem

Pada tahapan kebutuhan sistem langkah pertama yang dilakukan dalam penelitian adalah pengumpulan data dan informasi dari buku – buku referensi, modul, jurnal, serta artikel – artikel baik yang disediakan oleh perguruan tinggi ataupun yang di luar dari

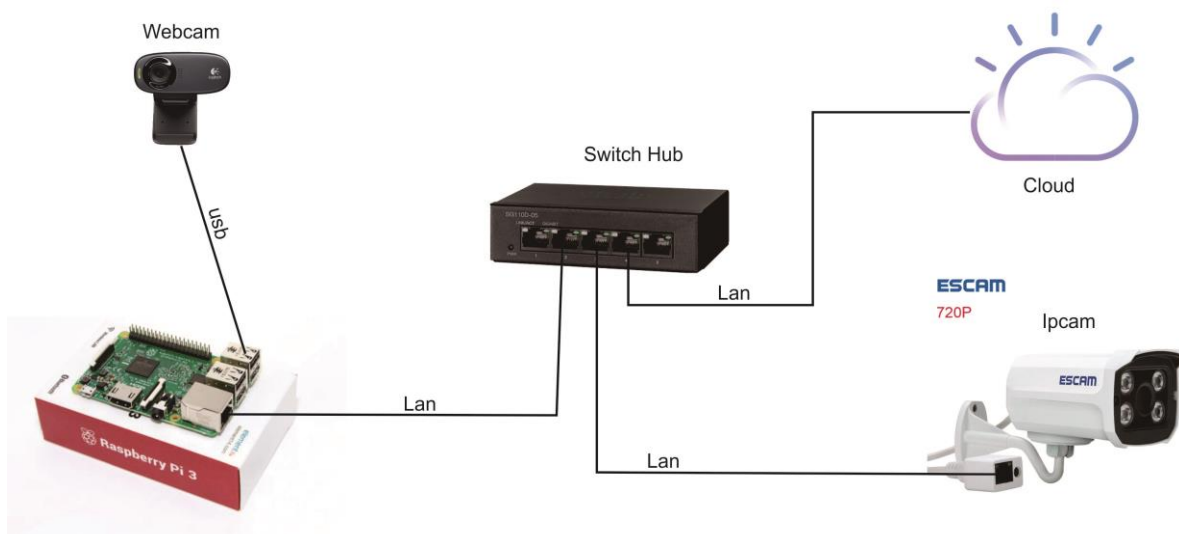
perguruan tinggi yang sesuai dengan objek penelitian. Menggunakan metode studi pustaka ini untuk mengumpulkan bahan – bahan dalam melakukan penelitian ini, apa itu pengujian perangkat lunak, bagaimana melakukan sebuah pengujian perangkat lunak dan lain sebagainya.

3.3. Perancangan Sistem

Pada tahapan perancangan sistem, yang merupakan kumpulan dari beberapa elemen yang saling berkaitan, kemudian beberapa elemen tersebut saling bertanggung jawab dalam memproses input sehingga menghasilkan sebuah output. Untuk dapat menghasilkan input dan output maka diperlukan elemen pendukung yang saling terkoneksi satu sama lain. Oleh karenanya di sini sangat diperlukan adanya perancangan hardware yang bertugas sebagai perangkat keras atau alat yang dapat menunjang penelitian ini, kemudian perlu juga adanya perancangan software yang bertugas sebagai perangkat lunak untuk menjalankan hardware atau perangkat keras sebagaimana mestinya.

3.3.1. Perancangan Hardware

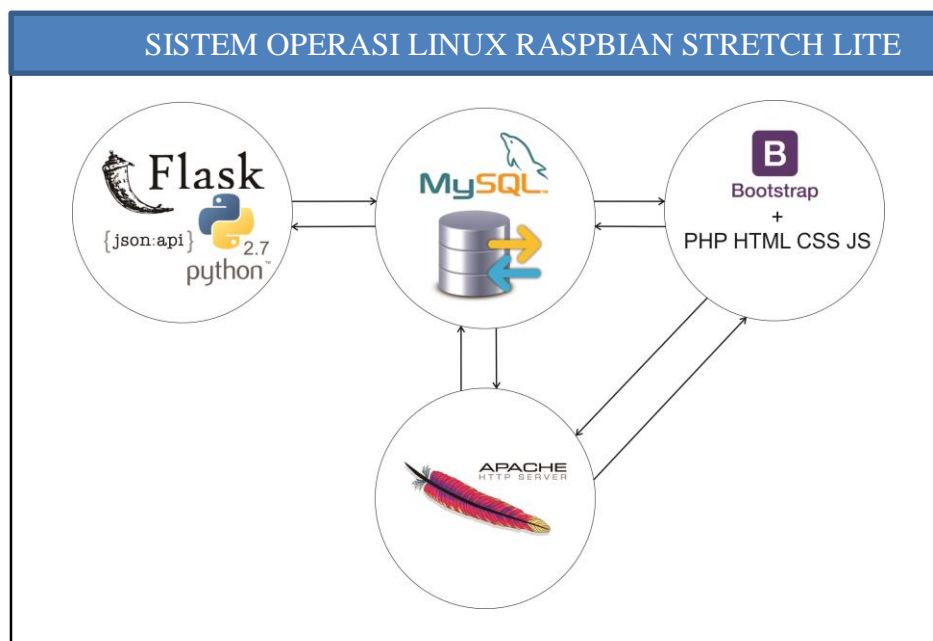
Untuk dapat menjalankan sistem kamera pengawas ini dibutuhkan beberapa perangkat yang terhubung dengan raspberry pi, di antaranya yaitu kamera web dan kamera ip. Khusus kamera ip dibutuhkan perangkat tambahan berupa switch hub agar dapat di remote dari perangkat *raspberry pi*. Dan juga agar implementasinya berjalan dengan semestinya, maka penulis menambahkan perangkat router outdoor *TP-LINK* dengan kode jenis TL-WA7210N untuk koneksi internetnya. Seperti tergambar pada gambar 3.2.



Gambar 3.2 Skema Perangkat yang terhubung dengan Raspi

3.3.2. Perancangan Software

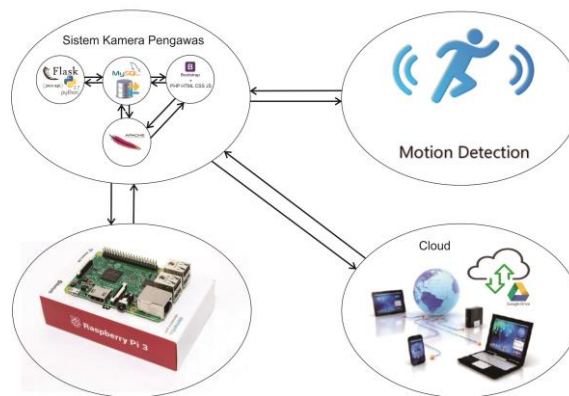
Agar sistem pada raspberry pi dapat berjalan dengan sempurna, maka dibutuhkan beberapa elemen pendukung yang pertama yaitu install sistem operasi linux ubuntu lite di dalam raspberry pi agar dapat menjalankan kode program dengan sumber daya perangkat yang minimalis, kemudian buat kode program sesuai dengan kebutuhan di antaranya menggunakan bahasa pemrograman python 2.7. Dalam program python ini digunakan untuk mengakses kamera ip dan kamera web. Selain itu dalam program python juga terdapat proses untuk merekam video dalam format .avi, kemudian terdapat logika percabangan juga untuk *motion detection* yaitu akan merekam gambar .jpg ketika terdapat pergerakan, kemudian hasil gambar tersebut di upload ke dalam Google Drive secara otomatis, sehingga admin akan mendapatkan data rekam gambar cadangan di akun *Google Drive*-nya. Di dalam program python juga terdapat fungsi flask server yang berfungsi agar admin dan pengguna dapat menampilkan data streaming di halaman sistem (Jay, 2015). Seluruh proses tersebut dapat berjalan bersamaan dengan menggunakan metode pemrograman thread. Agar data dapat masuk ke dalam database, maka dibutuhkan service mysql yang tersinkronisasi dengan bahasa pemrograman python. Kemudian agar data yang telah dikirim dari bahasa pemrograman python tersebut dapat di tampilkan ke halaman admin dan halaman pengguna, maka dibutuhkan web service yang berjalan bersamaan di dalam sistem, disini penulis menggunakan web service apache. Setelah semua proses dan service berjalan semua, maka sistem akan dapat di akses oleh admin dan pengguna. Seperti rancangan pada gambar 3.3 di bawah ini.



Gambar 3.3 Skema Sistem Pada Raspberry pi

3.3.3. Gabungan Perancangan Hardware dengan Software

Dalam penelitian ini dengan menggunakan perangkat *Raspberry Pi* dengan sistem operasi linux, dan kemudian di dalamnya dapat menjalankan sistem kamera pengawas yang telah terdapat kode program dengan logika apabila terdapat pergerakan, maka sistem akan menangkap gambar dengan format .jpg yang kemudian gambar tersebut secara otomatis di upload ke dalam *Google Drive*. Sehingga disini dengan menggunakan perangkat mini pc berupa *raspberry pi* sudah dapat menjalankan sistem terintegrasi dengan *Google Drive Api*. Sebagai gambaran perancangan perangkat keras (*Hardware*) dengan elemen perangkat lunak (*Software*) dapat tergambarkan pada gambar 3.4 di bawah ini.



Gambar 3.4 Skema Perangkat dan Sistem yang saling terhubung

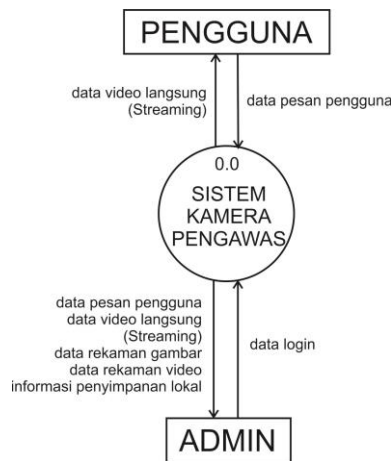
3.4. Software

3.4.1. Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah alat pembuatan model yang dapat digunakan untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi. Data Flow Diagram ini sering juga disebut dengan nama *Bubble chart*, *Bubble diagram*, model proses, diagram alur kerja, atau model fungsi. Dalam Data Flow Diagram juga terdapat tingkatan levelnya, yang mana diawali dari diagram level 0, level 1, dst sesuai dengan kebutuhan detail fungsi dari tiap proses pada sistem. Untuk itu, akan dijelaskan secara lengkap di bawah ini:

a. DFD Level 0

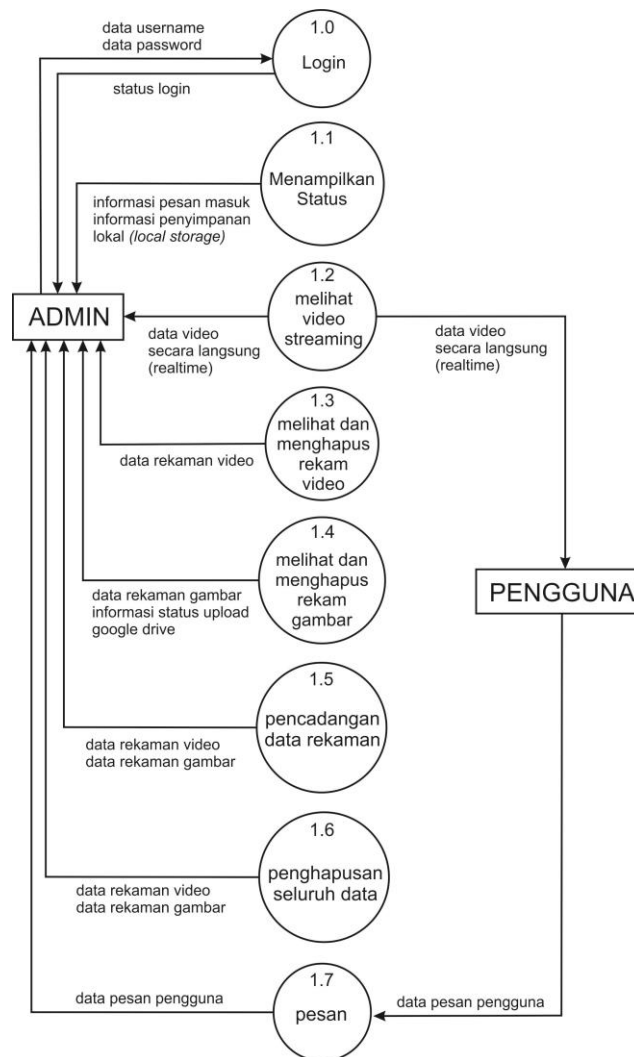
DFD level 0 merupakan gambaran model sistem secara umum yang mana menjelaskan proses alur data yang mengalir antara sistem dengan user ataupun pengguna. Untuk lebih jelasnya dapat dilihat pada gambar 3.5



Gambar 3.5 DFD Level 0

b. DFD Level 1

DFD level 1 merupakan gambaran model sistem yang lebih mendetail untuk proses alur datanya. Sehingga pada DFD level 1 ini dapat lebih jelas dan mudah untuk dipahami. Untuk lebih detailnya dapat dilihat pada gambar 3.6



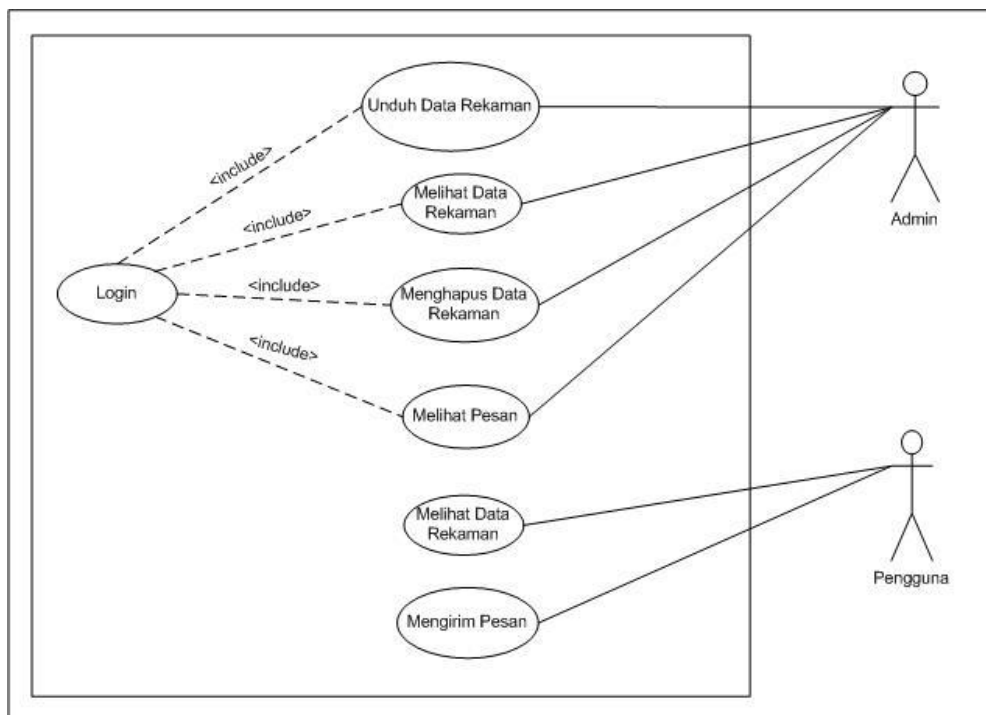
Gambar 3.6 DFD Level 1

3.4.2. Use Case

Use case diagram merupakan sebuah representasi visual yang mewakili interaksi antara pengguna dan system dalam *UML* (Rosenblatt, 2012) atau juga merupakan sebuah diagram untuk menunjukkan peran dari berbagai pengguna dan bagaimana peran – peran menggunakan system (Satzinger, 2009). Dari beberapa pengertian para ahli tersebut secara umum *use case diagram* merupakan sebuah diagram yang menggambarkan interaksi tentang apa saja yang dapat dilakukan antara *system* dan pengguna.

Manfaat serta tujuan dari penggunaan *use case diagram* yaitu untuk dapat melihat siapa saja yang akan berinteraksi dengan system, dapat melihat ruang lingkup system yang akan dibuat, serta melihat interaksi apa saja yang terjadi antara pengguna dan proses pada *system*.

Dalam pembuatan sistem kamera pengawas dengan menggunakan *Raspberry Pi*, saebelumnya sangat perlu membuat sebuah rancangan untuk menunjang kegunaan dari sebuah sistem ini. Berdasarkan rancangan dan analisa pengguna dari sistem kemudian dapat ditentukan kebutuhan sistemnya.



Gambar 3.7 Use Case Diagram sistem kamera pengawas

Gambar 3.7 Merupakan rancangan interaksi pada *system* dengan para *actor* (pengguna). Dalam rancangan tersebut terdapat 2 orang actor selaku *user* yang akan berinteraksi langsung dengan *system*. Pada actor admin terdapat interaksi proses, yaitu mengunduh data rekaman, melihat data rekaman, menghapus data rekaman, melihat pesan.

Gambar 3.7 Terdapat sebuah proses *login*, yang terdapat tanda *include relation* yang terkait pada beberapa aktifitas. Maksudnya ialah pada aktifitas – aktifitas tersebut perlu akses *login* sebelum memulainya, seperti pada *admin* yang perlu melakukan *login* agar dapat melakukan interaksi proses. Untuk aktor pengguna juga terdapat proses melihat data rekaman dan mengirim pesan.

3.4.3. Identifikasi Use Case Diagram

Terdapat tujuh case yang telah dipetakan oleh kebutuhan fungsionalitas perangkat lunak sistem kamera pengawas. Keterangan use case diagram dari sistem tersebut pada tabel 3.1

Tabel 3.1 Identifikasi Use Case Diagram

Kode	Nama dan Deskripsi
SKP01	<u>Login</u> Deskripsi: Admin harus login ke halaman dashboard untuk dapat melakukan proses melihat, menghapus, membackup data rekaman.
SKP02	<u>Mengunduh Data Rekaman</u> Deskripsi: Admin dapat melakukan proses unduh data rekaman dengan mengunduh data dengan format .jpg, .avi ataupun di unduh dalam format .zip
SKP03	<u>Melihat Data Rekaman</u> Deskripsi: Admin dapat melihat data rekaman secara <i>streaming</i> atau melihat dengan pilihan format video .avi atau pilihan format .jpg yaitu data rekaman berdasarkan hasil tangkapan gambar ketika terdapat pergerakan
SKP04	<u>Menghapus Data rekaman</u> Deskripsi: Admin dapat menghapus data rekaman secara terpilih ataupun menghapus data secara keseluruhan
SKP05	<u>Melihat Pesan</u> Deksripsi: Admin dapat melihat pesan yang di kirim dari pengguna, dimana juga terdapat pilihan untuk melihat detail pesan masuk

Kode	Nama dan Deskripsi
SKP06	<u>Melihat Data Rekaman</u> Deskripsi: Pengguna dapat melihat data rekaman secara <i>streaming</i>
SKP07	<u>Mengirim Pesan</u> Deskripsi: Pengguna dapat mengirim pesan kepada admin

3.4.4. Skenario Use Case Diagram

a. SKP01 Login

Aktor: Admin

Prerequisit: -

1. Sistem aplikasi menampilkan halaman awal dashboard.
2. Admin melihat notifikasi pesan, dan besaran penyimpanan lokal yang masih tersedia.
3. Sistem memberikan notifikasi pesan jika terdapat pesan masuk.
4. Admin melihat streaming hasil rekaman pada sistem.

b. SKP02 Mengunduh Data Rekaman

Aktor: Admin

Prerequisit: Login

1. Admin melihat dan mengunduh file data rekaman pada menu rekaman dalam format video .avi.
2. Menu gambar dapat mengunduh dalam format gambar .jpg.
3. Pada menu pengaturan terdapat pilihan backup unduh pada format .zip.

c. SKP03 Melihat Data Rekaman

Aktor: Admin

Prerequisit: Login

1. Admin melihat data rekaman video dengan format .avi.
2. Admin melihat data rekaman gambar dengan format .jpg yang di simpan ketika terdeteksi adanya pergerakan.

d. SKP04 Menghapus Data Rekaman

Aktor: Admin

Prerequisit: Login

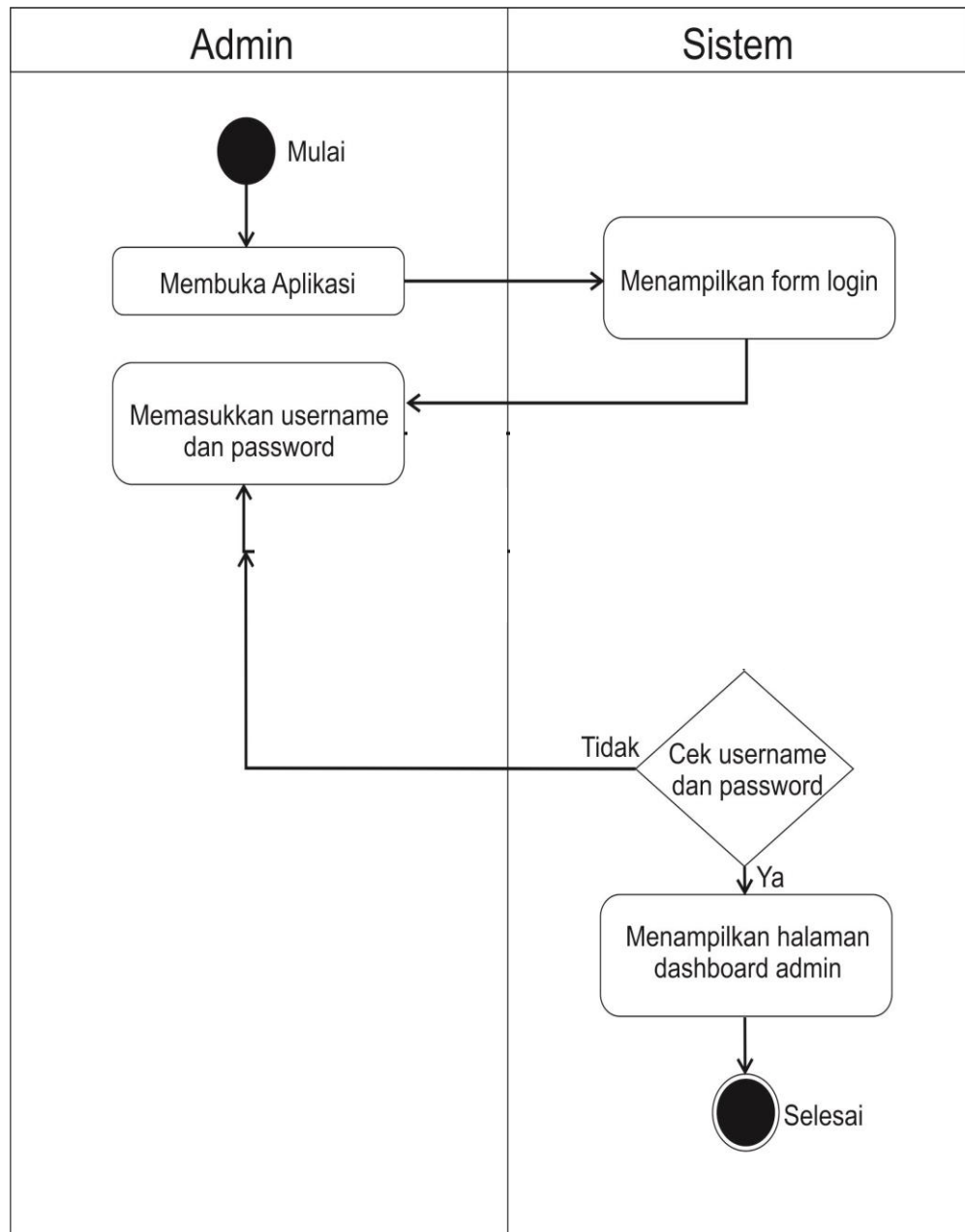
1. Sistem menampilkan data rekaman.
 2. Admin memilih data rekaman yang akan dihapus.
 3. Admin dapat menghapus semua data rekaman sekaligus.
- e. SKP05 Melihat Pesan
- Aktor: Admin
- Prerequisit: Login
1. Sistem menampilkan notifikasi jika terdapat pesan masuk.
 2. Terdapat fitur pencarian pesan, detail pesan, dan hapus pesan.
 3. Admin hanya dapat membaca pesan dan tidak dapat membalas pesan tersebut melalui sistem ini, namun dalam pesan, pengirim diminta untuk memasukkan alamat email. Sehingga admin tetap dapat berkomunikasi dengan pengguna yang mengirim pesan.
- f. SKP06 Melihat Data Rekam
- Aktor: Pengguna
- Prerequisit: -
1. Pengguna melihat data rekam secara realtime, namun tidak dapat melakukan penghapusan dan hanya dapat melihat data rekam pada saat itu juga.
- g. SKP07 Mengirim Pesan
- Aktor: Pengguna
- Prerequisit: -
1. Pengguna mengirim pesan dengan syarat memasukkan nama, alamat email, dan isi pesan yang diminta untuk dikirim kepada admin.
 2. Pesan dikirim melalui sistem dan untuk dapat membacanya diperlukan masuk ke halaman admin.

3.4.5. *Activity Diagram*

Activity diagram adalah suatu metode yang menggambarkan logika prosedural, proses bisnis, dan jalur kerja (Fowler, 2005). *Activity diagram* akan menggambarkan berbagai aliran aktifitas dalam rancangan *system*, bagaimana aliran berawal, *decision* yang mungkin terjadi, dan bagaimana aktifitas itu berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

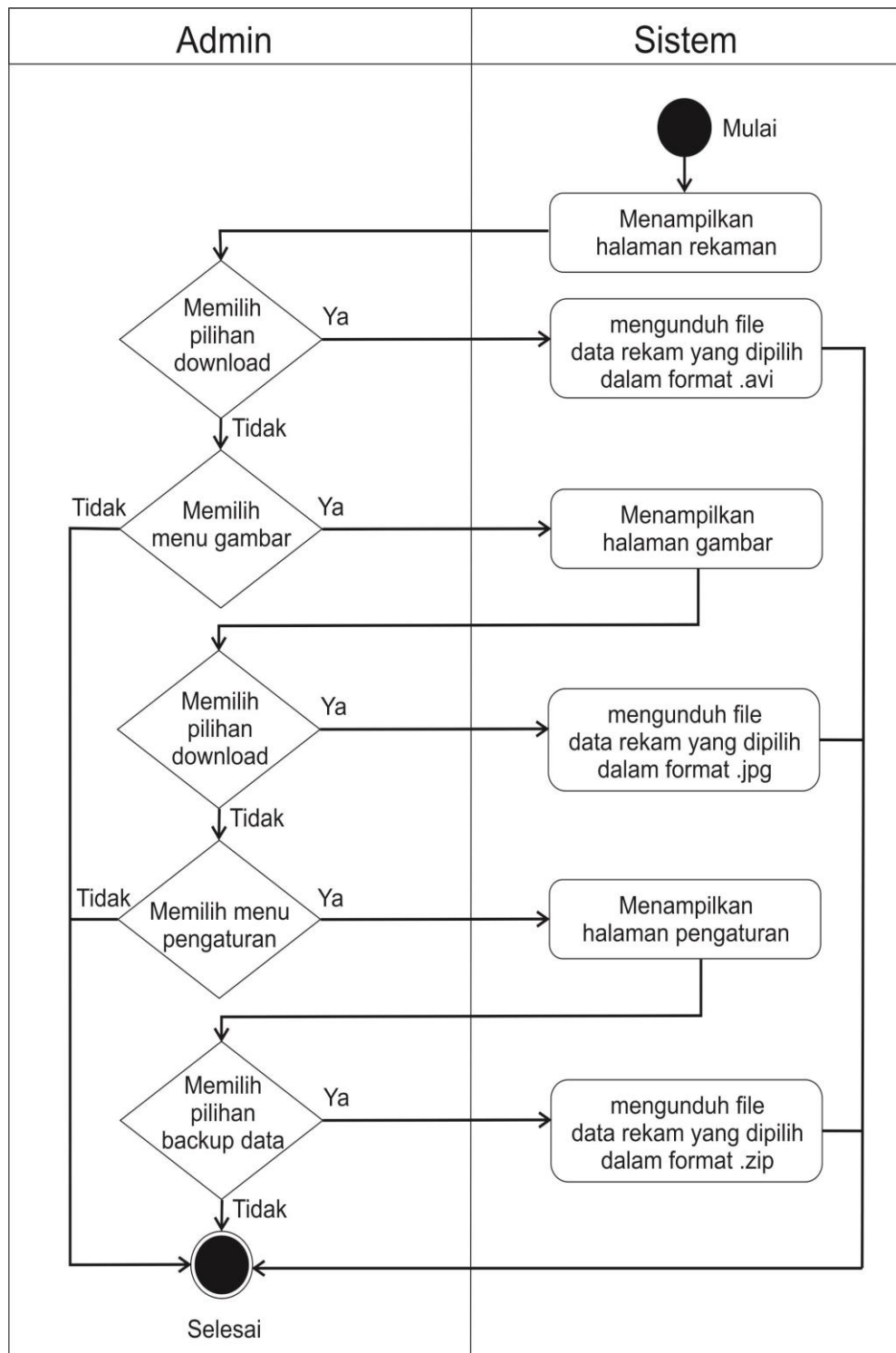
Manfaat serta tujuan dari penggunaan *activity diagram* adalah untuk menggambarkan semua aktifitas secara global dalam sebuah *system*. Dengan melihat *activity diagram*, pengguna dapat mengetahui apa saja yang dapat dilakukan pada sebuah *system*. Selain itu,

activity diagram juga bermanfaat untuk menggambarkan *parrarel behavior* atau menggambarkan interaksi dari beberapa *use case*.



Gambar 3.8 Activity Diagram Login Admin pada Sistem

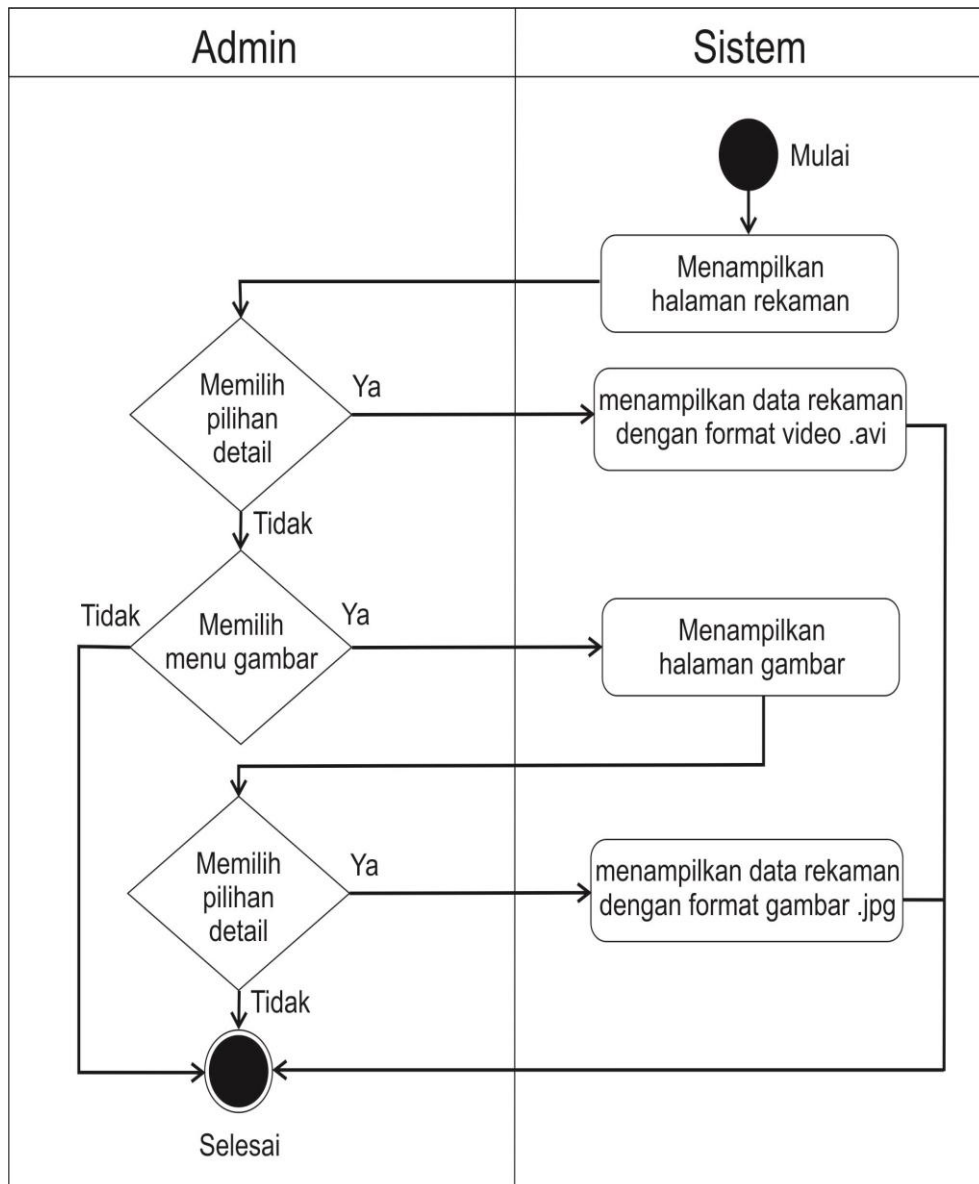
Gambar 3.8 Adalah gambaran aktifitas admin ketika melakukan login ke dalam halaman utama sistem. Pada tampilan login terdapat masukan yang berupa *username* serta *password* yang kemudian dilakukan pengecekan terhadap database pada sistem. Apabila *username* serta *password* yang dimasukkan sesuai dengan yang ada pada database, maka pengguna dapat masuk sebagai admin dan dapat mengelola isi dari pada sistem ini. Namun jika tidak, atau terdapat kesalahan input pada username ataupun password, maka akan dialihkan pada halaman login.



Gambar 3.9 Activity Diagram Unduh Data Rekam pada Sistem

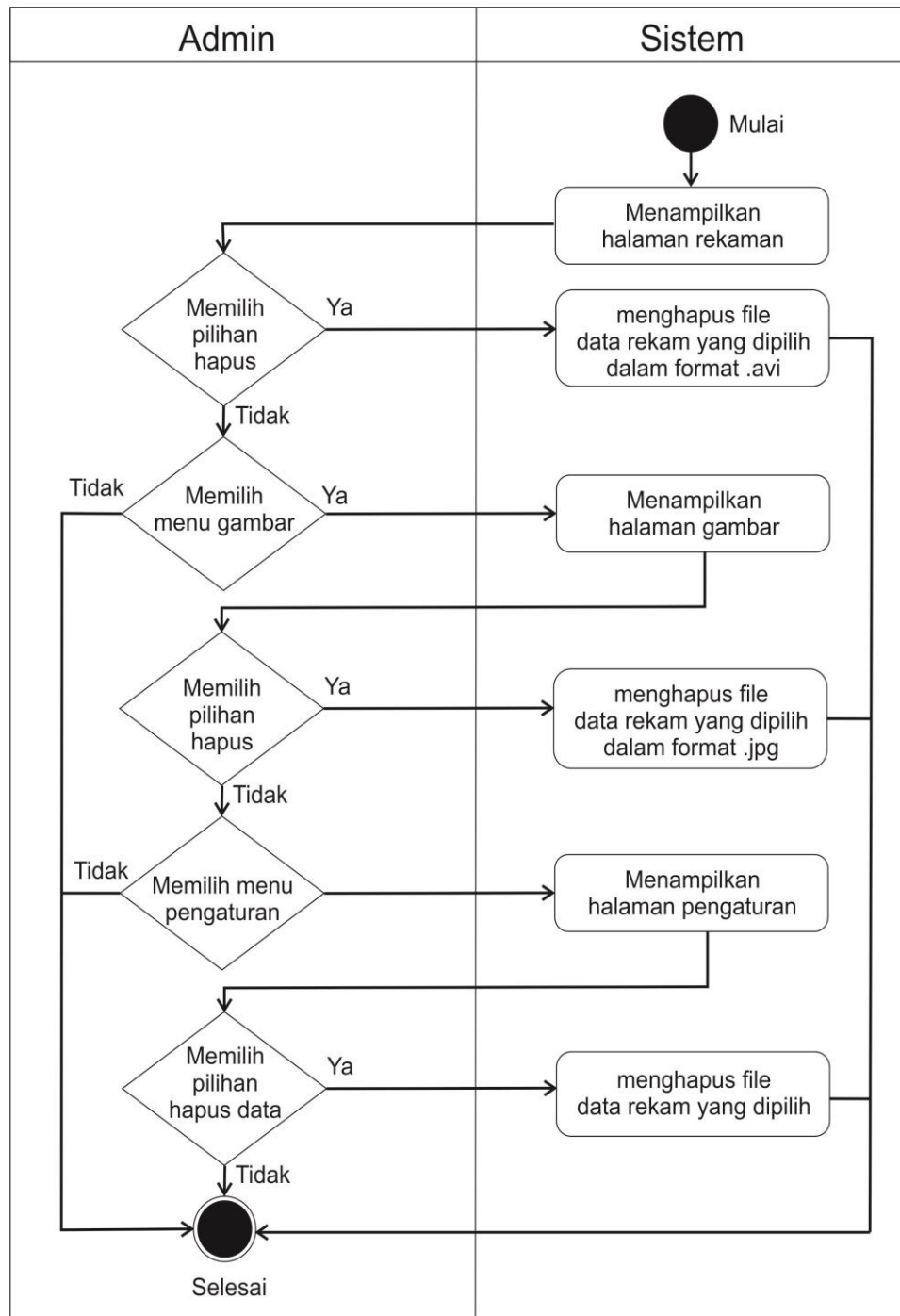
Gambar 3.9 Merupakan gambaran aktifitas admin ketika mengunduh data rekaman yang terdapat pada sistem. Pada aktifitas ini terdapat beberapa pilihan untuk dapat mengunduh data rekaman dalam beberapa format, yang pertama admin dapat mengunduh data rekaman pada pilihan menu rekaman dengan format data rekaman .avi yang berarti file

data rekaman berupa video. Kemudian admin juga dapat mengunduh data rekaman dalam format .jpg yang berarti data rekaman gambar yang direkam ketika terdeteksi adanya pergerakan. Pilihan terakhir yaitu admin dapat meunduh data rekaman pada pilihan pengaturan dengan format .zip yang berarti admin mengunduh semua data rekaman yang ada pada penyimpanan lokal.



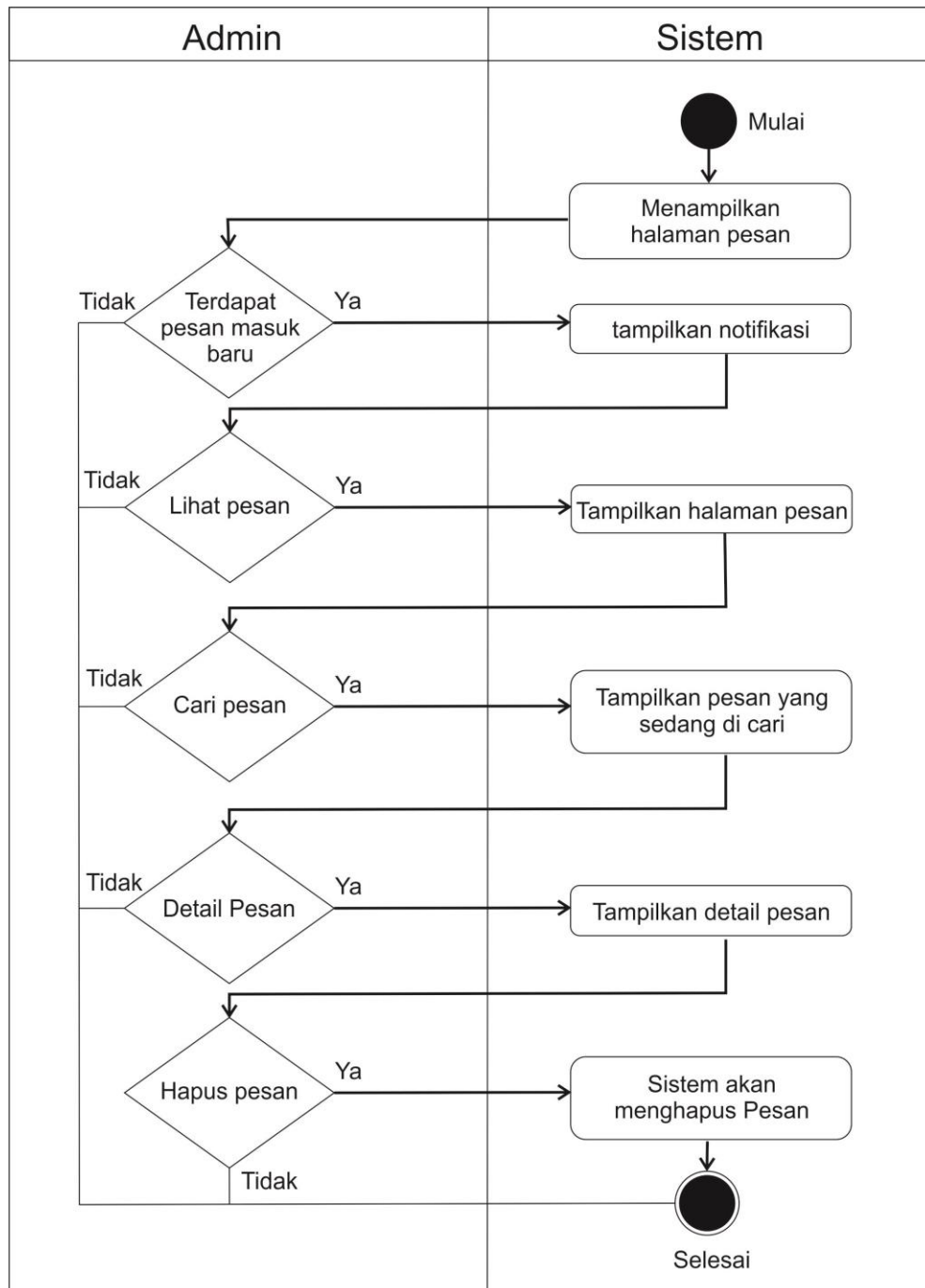
Gambar 3.10 Activity Diagram Melihat Data Rekam pada Sistem

Gambar 3.10 Merupakan gambaran aktifitas admin ketika melihat data rekaman pada sistem. Admin dapat melihat data rekaman dalam sistem lokal yang berupa video dengan format .avi dan dapat melihat data rekaman dalam bentuk gambar dalam format .jpg.



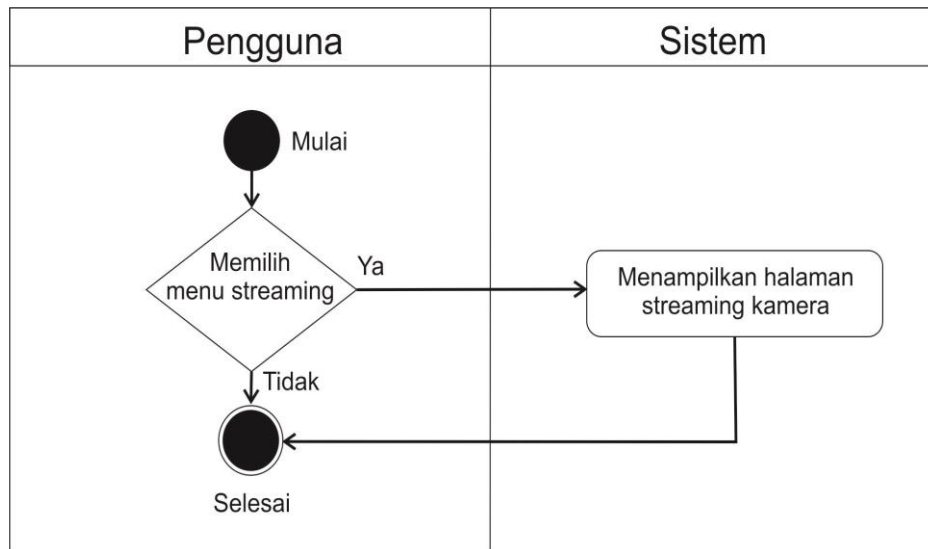
Gambar 3.11 Activity Diagram Menghapus Data Rekam pada Sistem

Gambar 3.11 Merupakan gambaran aktifitas admin ketika menghapus data, yang pertama admin dapat menghapus data melalui menu rekaman, dimana file yang akan dihapus berupa file video dengan format .avi. Kemudian admin juga dapat menghapus data gambar yang terdapat pada menu gambar, yang mana file yang terdapat pada menu gambar merupakan file rekaman gambar dengan format .jpg. Selain itu admin juga dapat menghapus secara keseluruhan data melalui menu pengaturan.



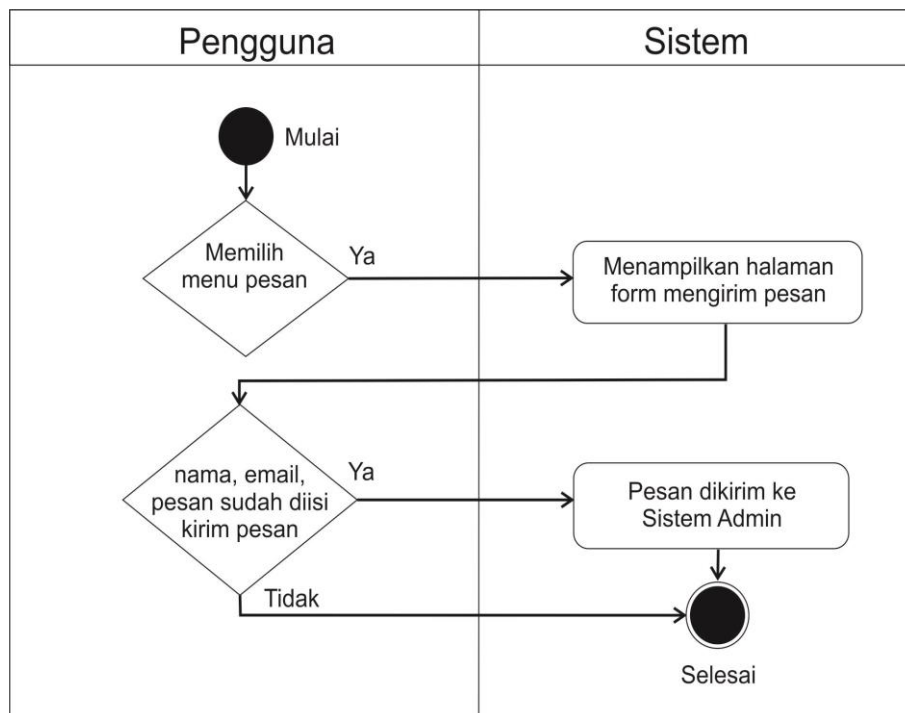
Gambar 3.12 Activity Diagram Melihat Pesan

Gambar 3.12 Merupakan gambaran aktifitas admin ketika melihat pesan masuk ke sistem. Ketika admin login ke dalam sistem, jika terdapat pesan masuk maka akan ada pesan notifikasi yang mana notifikasi ini jika dibuka akan merujuk kepada halaman pesan. Kemudian halaman pesan admin dapat melakukan beberapa aktifitas diantaranya admin dapat mencari pesan masuk, admin juga dapat melihat secara detail pesan yang masuk ke dalam sistem. Selain itu admin juga dapat menghapus pesan yang sudah tidak terpakai.



Gambar 3.13 Activity Diagram Pengguna Melihat Streaming Kamera

Gambar 3.13 Merupakan gambaran aktifitas pengguna ketika melihat streaming kamera secara langsung. Disini pengguna dapat melihat streaming di salah satu kamera, dan tidak dapat melihat beberapa kamera sekaligus.



Gambar 3.14 Activity Diagram Pengguna Mengirim pesan

Gambar 3.14 Merupakan gambaran aktifitas pengguna ketika mengirim pesan kepada admin. Pengguna diharuskan mengirim nama, email, dan pesan yang akan dikirimkan sebelum menekan tombol kirim pesan.

3.4.6. Perancangan Basis Data

Pada pembuatan sistem kamera pengawas ini dibutuhkan basis data untuk menampung data yang dibutuhkan oleh sistem. Dalam sistem ini basis data dibutuhkan untuk proses autentifikasi pada login. Selain itu basis data juga digunakan untuk menyimpan judul, tanggal pada data rekaman, juga untuk menyimpan perpesanan yang masuk antara admin dengan pengguna. Oleh karenanya basis data harus dirancang dengan baik untuk menciptakan sistem yang dapat berjalan dengan sempurna.

a. Struktur Tabel

Setiap entitas yang terdapat pada diagram relasi dirubah ke dalam bentuk tabel. Sedangkan atribut-atribut yang melekat pada entitas tersebut di konversi ke dalam isian kolom pada tabel.



Gambar 3.15 Struktur Tabel Basis Data pada Sistem Kamera Pengawas

3.4.7. Perancangan Antarmuka

a. Halaman Login

Halaman login merupakan halaman awal untuk admin dapat masuk ke dalam dashboard atau halaman admin.

X
Logo

username

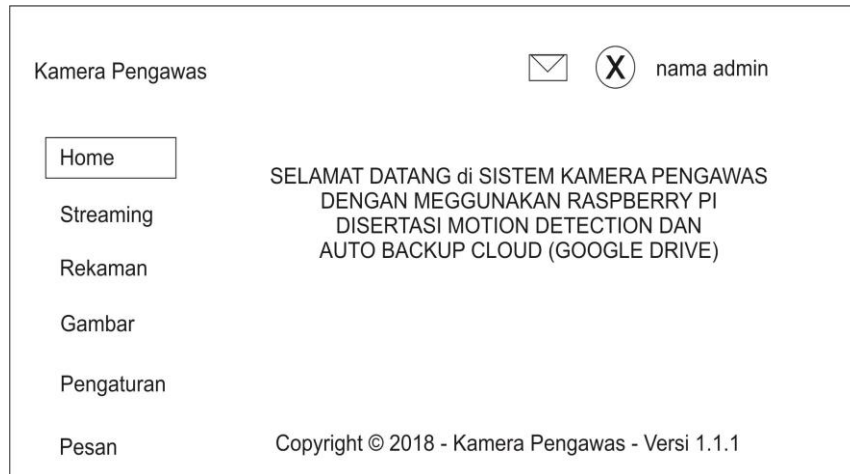
password

login

Gambar 3.16 Rancangan Halaman Login

b. Halaman Utama Admin

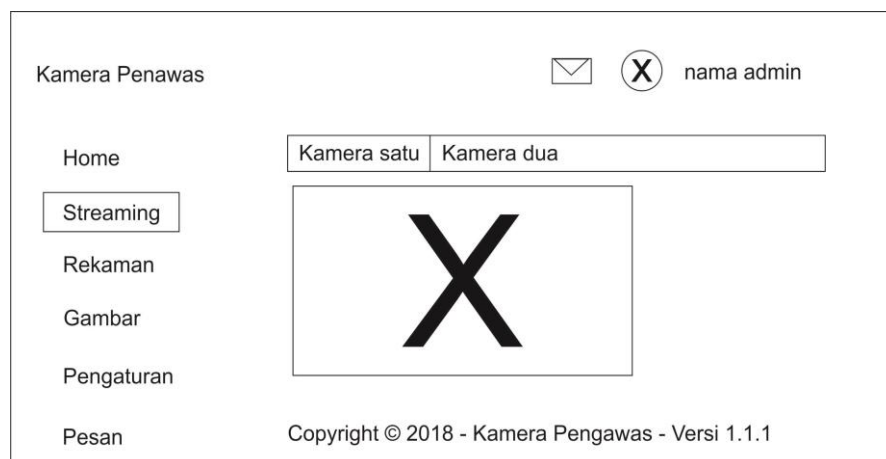
Setelah Admin memasukkan username dan password dengan benar di halaman login, maka admin akan di arahkan pada halaman utama admin. Pada halaman ini terdapat tampilan notifikasi jika terdapat pesan masuk, selain itu juga di halaman utama admin terdapat ucapan selamat datang.



Gambar 3.17 Rancangan Halaman Utama Admin

c. Halaman Streaming

Pada halaman streaming terdapat pilihan kamera satu atau kamera dua yang dapat di tampilkan pada halaman ini. Jadi hanya dapat menampilkan tayangan langsung dari salah satu kamera, hal ini dikarenakan untuk menghemat *resource* dari perangkat raspberry pi.

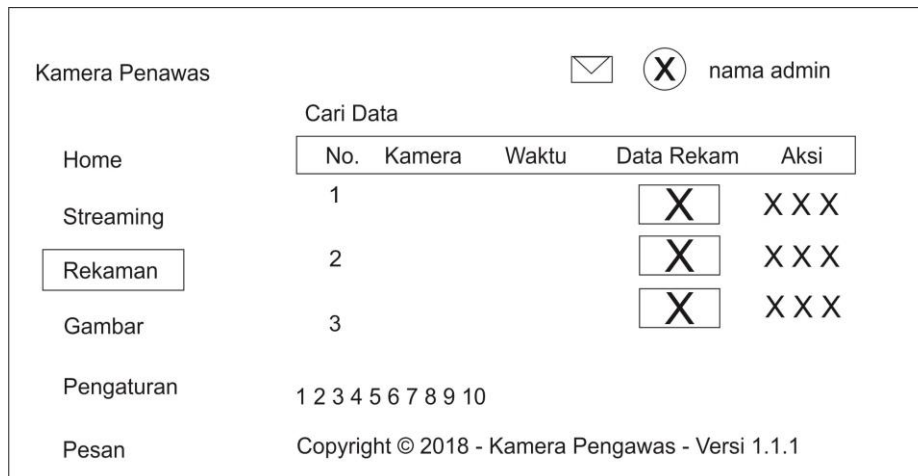


Gambar 3.18 Rancangan Halaman Streaming Admin

d. Halaman Rekaman

Halaman rekaman merupakan halaman yang berfungsi untuk menampilkan hasil rekaman yang berupa video dengan format rekaman *.avi*. pada halaman ini juga terdapat

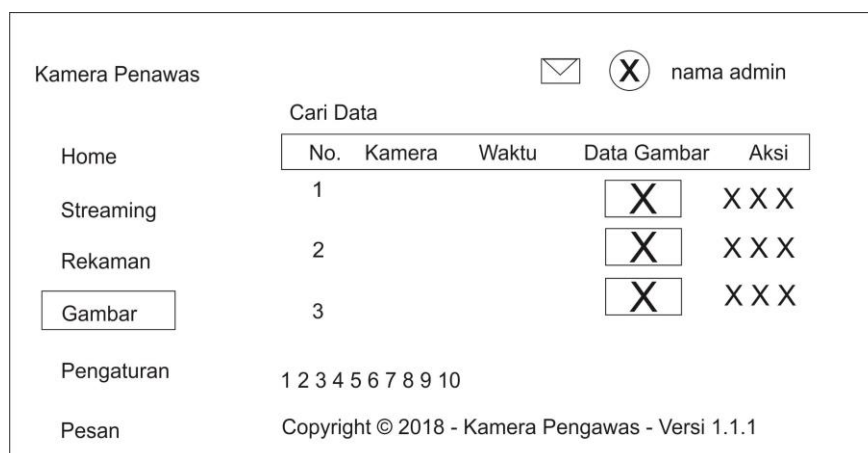
pilihan untuk mencari data, menghapus data, melihat secara detail, dan mengunduh data rekaman.



Gambar 3.19 Rancangan Halaman Rekaman Admin

e. Halaman Gambar

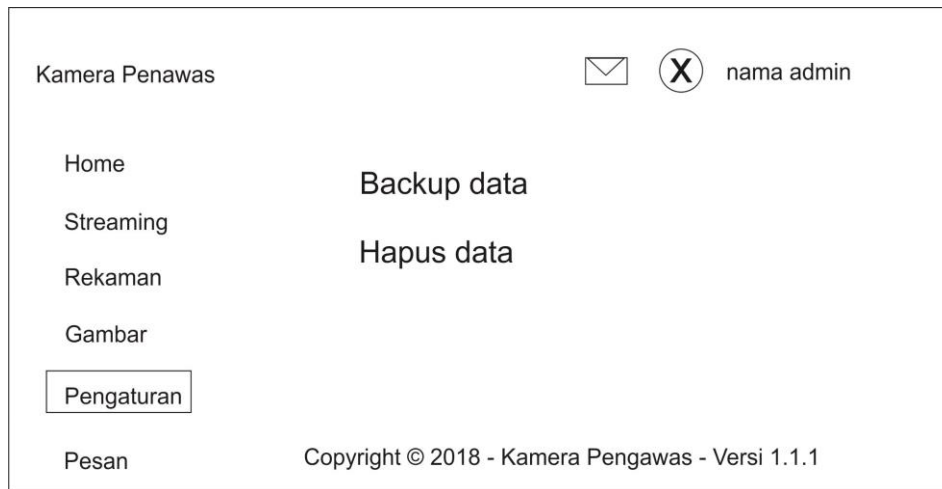
Halaman ini tidak jauh berbeda dengan halaman rekaman, hanya saja data gambar disini menampilkan data hasil tangkapan gambar dengan format .jpg ketika pada lokasi terdapat pergerakan, dan data gambar ini yang di upload ke cloud Google Drive secara otomatis.



Gambar 3.20 Rancangan Halaman Gambar Admin

f. Halaman Pengaturan

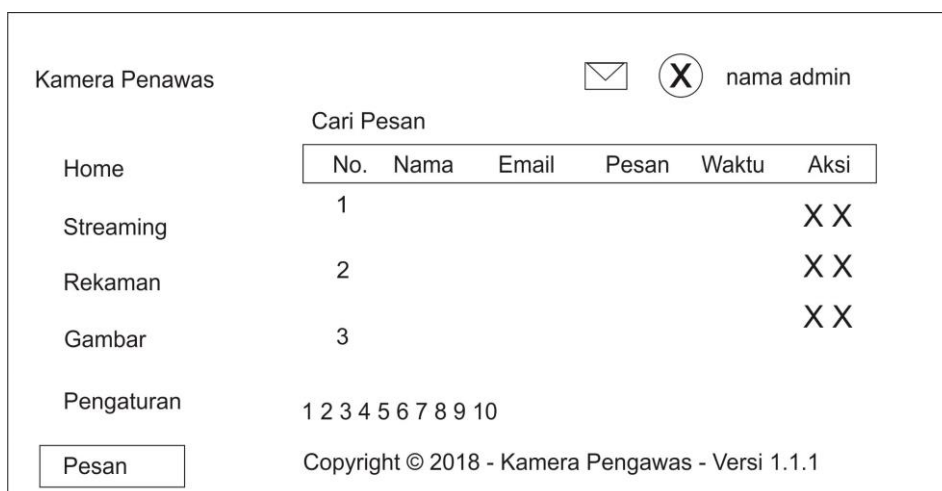
Halaman pengaturan ini terdapat pilihan untuk membackup data secara keseluruhan dengan hasil backup .zip, dan juga terdapat pilihan untuk menghapus data secara keseluruhan. Sehingga dapat membantu admin untuk manajemen data secara berkala.



Gambar 3.21 Rancangan Halaman Pengaturan Admin

g. Halaman Login

Halaman perpesanan ini terdapat tabel yang berisi pesan masuk yang dikirim oleh pengguna. Dalam tabel terdapat fungsi pencarian pesan, melihat detail pesan dan hapus pesan. Selain itu dalam halaman ini jika pesan belum terbaca, maka warna pesan akan berbeda dengan pesan yang sudah terbaca.

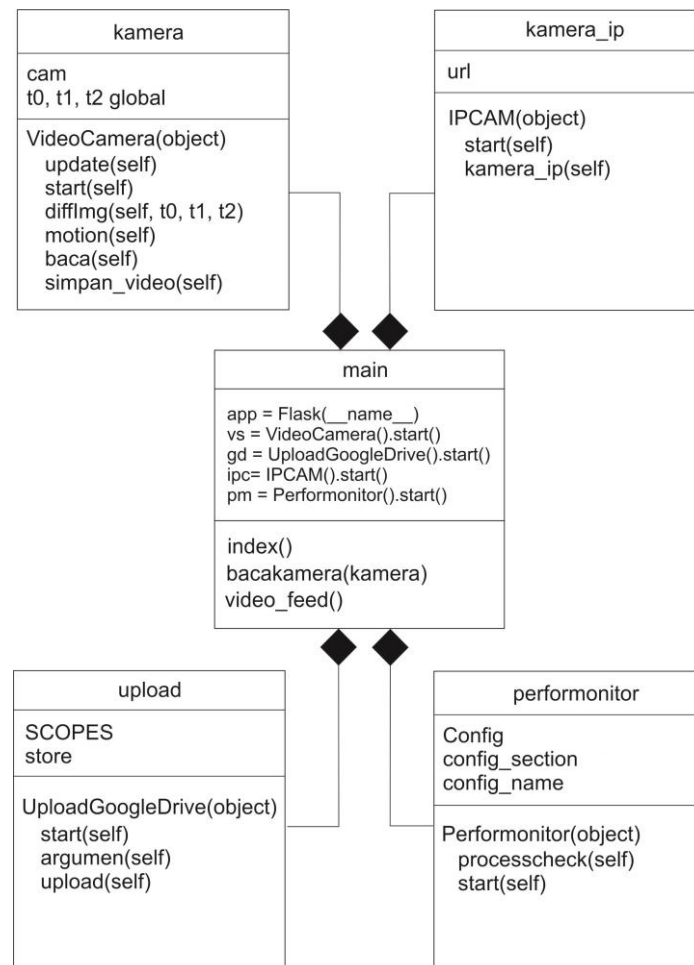


Gambar 3.22 Rancangan Halaman Pesan Admin

3.4.8. Class Diagram

Class merupakan sebuah spesifikasi yang apabila diinstansiasi akan menghasilkan sebuah objek yang merupakan inti dari pengembangan dan desain pemrograman berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi). *Class diagram* menggambarkan struktur dan deskripsi class, package, dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain. Sehingga adanya *class diagram*

ini dapat membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling ditemui dalam pemodelan sistem berorientasi *object*.



Gambar 3.23 Class Diagram

Gambar 3.23 Class Diagram diatas merupakan gambaran daripada system kamera pengawas ini. *Main class* sebagai inti dari program yang digunakan sebagai pemanggilan kelas – kelas yang lain. Relasi seperti diatas ini merupakan relasi komposit (*Composite*) yaitu relasi yang punya tingkat ketergantungan yang kuat, artinya *object* yang menjadi bagian (*part*) tergantung sangat kuat dengan object yang jadi whole classnya.

3.4.9. Pseudocode

Pseudocode merupakan salah satu cara penulisan algoritma yang hampir menyerupai bahasa pemrograman, namun ditulis lebih sederhana dengan bahasa baku yang lebih mudah untuk dipahami oleh manusia. Dengan menggunakan pseudocode akan lebih memudahkan dalam membuat sebuah sistem, karena dalam pseudocode telah dideskripsikan kerangka awal (*ide*) untuk menggambarkan logika pada suatu program dengan jelas tanpa memandang bahasa pemrograman yang akan digunakan.

<p>Judul:</p> <p>kamera.py</p>
<p>Deklarasi:</p> <pre> self.cam = cv2.VideoCapture(0) self.cam.set(3, 960) self.cam.set(4, 720) self.batas = 219000 self.cek_waktu = datetime.now().strftime('%Ss') t_minus = cv2.cvtColor(self.cam.read()[1], cv2.COLOR_RGB2GRAY) t = cv2.cvtColor(self.cam.read()[1], cv2.COLOR_RGB2GRAY) t_plus = cv2.cvtColor(self.cam.read()[1], cv2.COLOR_RGB2GRAY) jam = datetime.now().strftime("%d-%m-%Y %H-%M-%S") totalDiff = cv2.countNonZero(self.diffImg(t_minus, t, t_plus)) </pre>
<p>Implementasi:</p> <pre> While True: if totalDiff > batas and cek_waktu != datetime.now().strftime('%Ss'): dimg = self.cam.read()[1] cv2.imwrite('../penyimpanan/gambar/' + str(jam) + '.jpg'), dimg) </pre>

Gambar 3.24 Pseudocode kamera.py

Gambar 3.24 Pseudocode kamera.py di atas merupakan logika program untuk mengakses kamera usb yang mana pada implementasi program terdapat perulangan untuk mengakses kamera, kemudian di dalam perulangan terdapat logika percabangan dimana ketika nilai rata-rata yang diperoleh dari perbedaan warna telah melebihi batas yang sudah di deklarasikan dan waktu tidak sama dengan waktu sekarang maka akan melakukan pengambilan gambar dan kemudian di simpan dalam direktori './penyimpanan/gambar/' dengan nama waktu pada jam tersebut dan format gambar jpg. Dengan pengambilan gambar dengan format jpg maka akan dapat menghemat memori penyimpanan. Dikarenakan rata-rata gambar yang di ambil akan berukuran ± 1 megabyte tergantung kerapatan gambar dan warna. Proses ini juga dapat menghemat jalur akses internet yang nantinya gambar jpg ini akan di upload secara otomatis dengan koneksi internet ke *cloud* (*Google Drive*) pada proses upload.py. Selain itu dengan adanya pencadangan gambar secara otomatis ke *cloud* ini diharapkan akan lebih mengamankan barang bukti berupa gambar apabila perangkat kamera pengawas rusak ataupun hilang.

<p>Judul:</p> <p>kamera_ip.py</p>
<p>Deklarasi:</p> <pre> self.url = 'http://192.168.1.16:80/webcapture.jpg?command=snap&channel=1' self.cam = cv2.VideoCapture(self.url) (self.grabbed, self.frame) = self.cam.read() self.waktu = datetime.now().strftime("%d-%m-%Y %H:%M:%S") width = int(self.cam.get(cv2.CAP_PROP_FRAME_WIDTH) + 0.5) height = int(self.cam.get(cv2.CAP_PROP_FRAME_HEIGHT) + 0.5) fourcc = cv2.VideoWriter_fourcc(*'XVID') out = cv2.VideoWriter('../recording/' + str(self.waktu)+'.avi', fourcc, 20.0, (width, height)) video = (str(self.waktu)+'.avi') jam = datetime.now().strftime("%d-%m-%Y %H:%M:%S") </pre>
<p>Implementasi:</p> <pre> while True: imgResp = urllib.urlopen(self.url) imgNp = np.array(bytearray(imgResp.read()), dtype=np.uint8) img = cv2.imdecode(imgNp,-1) while(self.cam.isOpened()): if self.grabbed == True: out.write(img) if (cv2.waitKey(1) & 0xFF) == ord('q'): break else: break </pre>

Gambar 3.25 Pseudocode kamera_ip.py

Gambar 3.25 Pseudocode kamera_ip.py di atas merupakan logika program untuk mengakses kamera ip yang berbeda dengan logika untuk mengakses kamera *usb*. Dalam implementasi program terdapat perulangan untuk melakukan pengecekan koneksi melalui alamat url yang telah di deklarasikan sebelumnya jika masih terhubung maka logika program akan berlanjut pada logika percabangan untuk simpan gambar dengan perintah *out.write(img)* yang mana *out* telah di deklarasikan sebelumnya.

<p>Judul:</p> <p>upload.py</p>
<p>Deklarasi:</p> <pre> self.SCOPEES = 'https://www.googleapis.com/auth/drive.file' self.store = file.Storage('storage.json') self.creds = self.store.get() </pre>

Implementasi:

```

if not self.creds or self.creds.invalid:
    print("buat penyimpanan file data baru")
    flow = client.flow_from_clientsecrets('client_secret.json',
self.SCOPEES)
    self.creds = tools.run_flow(flow, store, self.flags) \
    if self.flags else tools.run(flow, self.store)
DRIVE = build('drive', 'v3', http=self.creds.authorize(Http()))
tmp=0
while True:
    time.sleep(.1)
    db = MySQLdb.connect("localhost","root","","nama-db")
    cursor = db.cursor()
    sql = "select gambar from gambars order by gambar_id desc
        limit 1"
    cursor.execute(sql)
    results = cursor.fetchall()

    for row in results:
        data = row[0]
        if tmp != data:
            tmp = data
            cek = True
        if cek == True:
            FILES = (('../pictures/'+data),)
            for file_title in FILES:
                file_name = file_title
                metadata = {'name': file_name,
                    'mimeType': None
                }
            res = DRIVE.files().create(body=metadata,
                media_body=file_name).execute()

            if res:
                file_name = file_title
                print('Uploaded "%s" (%s)' % (file_name,
                    res['mimeType']))
                #print('\n Upload berhasil')
                cek = False
                key = cv2.waitKey(10)
                if key == 27:
                    cv2.destroyWindow(winName)
                    break
                db.close()

```

Gambar 3.26 Pseudocode upload.py

Gambar 3.26 Pseudocode `upload.py` di atas merupakan logika program untuk *upload* gambar rekaman ke dalam akun *Google Drive* melalui layanan *Api* pada *Google Drive* (Awim, 2018). Sebelum menggunakan layanan ini meminta pengguna untuk melakukan verifikasi kode keamanan *Google Drive* yang bisa di dapat pada halaman *dashboard console Google Drive Api*. Oleh karenanya pada pendeklarasian di atas terdapat laman untuk melakukan autentifikasi yaitu proses yang berfungsi untuk melakukan pembuktian (*validasi*) terhadap identitas dari pengguna agar dapat memasuki atau mengakses sistem *Google Drive*. Langkah berikutnya yaitu pada logika perulangan untuk melakukan pengecekan data. Data ini didapat dari database yang telah tersimpan pada sistem kamera pengawas. Dengan menggunakan perulangan *for* akan di dapatkan data pada *array* ke 0. Ketika data baru masuk maka akan dilakukan pengecekan dan sistem akan melakukan pengambilan data pada direktori *Raspberry Pi* untuk di *upload* ke dalam akun *Google Drive* yang sudah disetujui sebelumnya. Hal tersebut akan dilakukan berulang terus menerus hingga sudah tidak menemukan data baru dan sistem dihentikan.

<p>Judul:</p> <p>main.py</p>
<p>Deklarasi:</p> <pre>app = Flask(__name__) vs = VideoCamera().start() gd = UploadGoogleDrive().start() ipc= IPCAM().start() pm = Performonitor().start()</pre>
<p>Implementasi:</p> <pre>@app.route('/') def index(): return render_template('index.html') def bacakamera(kamera): while True: frame = vs.baca() time.sleep(1) yield (b'--frame\r\n' b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n\r\n') @app.route('/video_feed') def video_feed(): return Response(bacakamera(VideoCamera()), mimetype='multipart/x-mixed-replace;</pre>

```

        boundary=frame')
if __name__ == '__main__':
    logging.basicConfig(format='%(asctime)s %(message)s',
        filename='kamerapengawas.log', level=logging.INFO)
    app.run(host='0.0.0.0', port=8081)

```

Gambar 3.27 Pseudocode main.py

Gambar 3.27 Pseudocode main.py di atas merupakan logika program untuk menjalankan program secara keseluruhan. Oleh karenanya pada bagian deklarasi program main merupakan pemanggilan fungsi yang sudah dibuat sebelumnya. Kemudian di sini karena menggunakan modul flask, maka perlu adanya perulangan agar video rekaman dapat ditayangkan secara langsung atau *streaming video*. Dengan menggunakan *flask server* maka pengguna ataupun admin dapat melihat video secara langsung dengan mengakses alamat dan *port* yang sudah di *setting* sebelumnya.

3.4.10. Pengujian

Penelitian ini harus melalui tahap pengujian, untuk memastikan apakah semua fungsi pada sistem berkerja dengan baik dan benar, maka perlu dilakukan sebuah pengujian. Pengujian ini menggunakan metode *black box* atau pengujian dilakukan berdasarkan fungsionalitas sistem. Sehingga uraian pertanyaan yang diujikan pada pihak-pihak yang dijadikan sebagai sampel penelitian (*responden*) berdasarkan apa yang dilihat, hanya berfokus kepada fungsionalitas dan output. Kemudian untuk perhitungan skor dengan menggunakan skala likert yaitu dengan menggunakan beberapa butir pertanyaan untuk mengukur individu dengan merespon 5 titik pilihan pada setiap butir pertanyaan Sangat Setuju/SS (Skor 5), Setuju/S (Skor 4), Cukup Setuju/CS (Skor 3), Tidak Setuju/TS (Skor 2), dan Sangat Tidak Setuju/STS (Skor 1) (Choizes, 2018).

Tabel 3.2 Pengujian

NO	URAIAN	SKOR				
		1	2	3	4	5
A.	MANFAAT					
1.	Sistem kamera pengawas ini bermanfaat untuk masyarakat umum di Indonesia?					
2	jika sistem ini diterapkan di lingkungan anda maka dapat membantu mengurangi tindak kejahatan di sekitar anda					
3	Fitur auto backup cloud (Google Drive) dapat membantu mengamankan barang bukti jika terdapat kelalaian baik sengaja ataupun tidak sengaja					

NO	URAIAN	SKOR				
		1	2	3	4	5
A.	MANFAAT					
4	Fitur hapus semua data dapat bermanfaat dan menghemat waktu admin untuk manajemen data rekaman baik berupa video maupun gambar					
5	Fitur pesan dari sisi pengguna dapat membantu pengguna untuk memberikan saran ataupun permintaan kepada admin					
6	Fitur backup dengan format .zip dapat membantu admin untuk mengarsipkan data rekaman					
B.	TAMPILAN	1	2	3	4	5
1.	Dari sisi pengguna tampilan menarik					
2.	Video player pada halaman admin tampilan menarik					
3.	Dari sisi Dashboard Admin tampilan menarik					
C.	FUNGSIONALITAS					
1.	Fitur pendeteksi pergerakan (motion detection) dapat berfungsi sebagaimana mestinya					
2.	Fitur auto backup data gambar .jpg ke akun <i>Google Drive</i> dapat berjalan secara otomatis					
3.	Fungsi pengarsipan berkas dengan format .zip dapat dijalankan					
4.	Perpesanan antara pengguna dengan admin dapat berjalan normal					
5.	Info pengecekan penyimpanan lokal dapat bekerja dengan baik					
6.	Fungsi untuk melihat video rekaman secara langsung (<i>streaming</i>) dapat berjalan normal					
7.	Fungsi hak akses admin dapat beroperasi normal, jika pengguna benar-benar memiliki akses maka akan disetujui, namun jika tidak maka sebaliknya.					
Total						

Kritik dan Saran:

BAB IV

HASIL DAN PEMBAHASAN

Bab ini membahas tentang hasil sistem kamera pengawas dan pembahasan implementasi penelitian yang dilakukan untuk dapat mengetahui manfaat daripada penggunaan sistem ini. Bab ini dibagi menjadi beberapa subbab sesuai dengan parameter – parameter yang telah ditentukan.

4.1. Hasil

4.1.1. Hasil Pembuatan Perangkat Keras (Hardware)

a. Rangkaian *Raspberry Pi*

Penelitian kamera pengawas ini menggunakan raspberry pi sebagai media pengendali utama pada sistem ini. Dengan menggunakan bantuan raspberry pi 2 yang dilengkapi dengan 4 *port usb*, *port LAN*, *WLAN*, dengan power adaptor yang kecil menjadikan perangkat ini lebih dapat diandalkan daripada harus menggunakan perangkat komputer yang lebih besar menghabiskan daya listrik (Upton, 2013). Berikut merupakan tampilan perangkat raspberry pi seperti pada gambar 4.1.



Gambar 4.1 Perangkat Raspberry Pi

b. Rangkaian kamera usb dan ip kamera

Rangkaian kamera ini merupakan salah satu rangkaian terpenting dalam penelitian ini. Oleh karena itu butuh pengetahuan umum mengenai komunikasi dan akses antar kamera yang berbeda jalur komunikasinya. Agar rangkaian raspberry pi dapat terkoneksi dengan

ip kamera, dan maka dibutuhkan perangkat tambahan untuk mendukung koneksi antar perangkat tersebut yaitu berupa *switch*. Tampilan rangkaian dari kamera usb dan ip kamera dapat dilihat pada gambar 4.2



Gambar 4.2 Rangkaian komunikasi kamera usb dan ip kamera

c. Rangkaian Perangkat Keseluruhan

Agar sistem kamera pengawas dapat berjalan sebagaimana mestinya, maka dibutuhkan sebuah rangkaian yang saling terhubung. Media jaringan yang digunakan yaitu menggunakan LAN dan WIFI. Hal ini diterapkan sesuai dengan kejadian di lapangan, di mana tidak semua rumah dapat terhubung dengan koneksi internet kabel, maka disini dibutuhkan sebuah perangkat tambahan yaitu router yang bertindak sebagai antena *reciver* agar perangkat raspberry pi mendapatkan koneksi internet. Dalam router ini juga terdapat fitur *port forwarding* yang mana perangkat raspberry pi ini dapat dikendalikan jarak jauh secara *remote* melalui *ip public* yang didapatkan oleh perangkat *router* tersebut. Berikut merupakan rangkaian perangkat secara keseluruhan pada gambar 4.3.



Gambar 4.3 Rangkaian perangkat keseluruhan

4.1.2. Hasil Pembuatan Perangkat Lunak (Software)

Hasil implementasi pada aplikasi web dashboard admin ini merupakan antarmuka web yang telah dirancang sebelumnya dengan menggunakan bantuan software *visual studio code*. Berikut ini merupakan hasil implementasi penerapan sistem dashboard admin dengan menggunakan web browser mozilla firefox 60.0.2



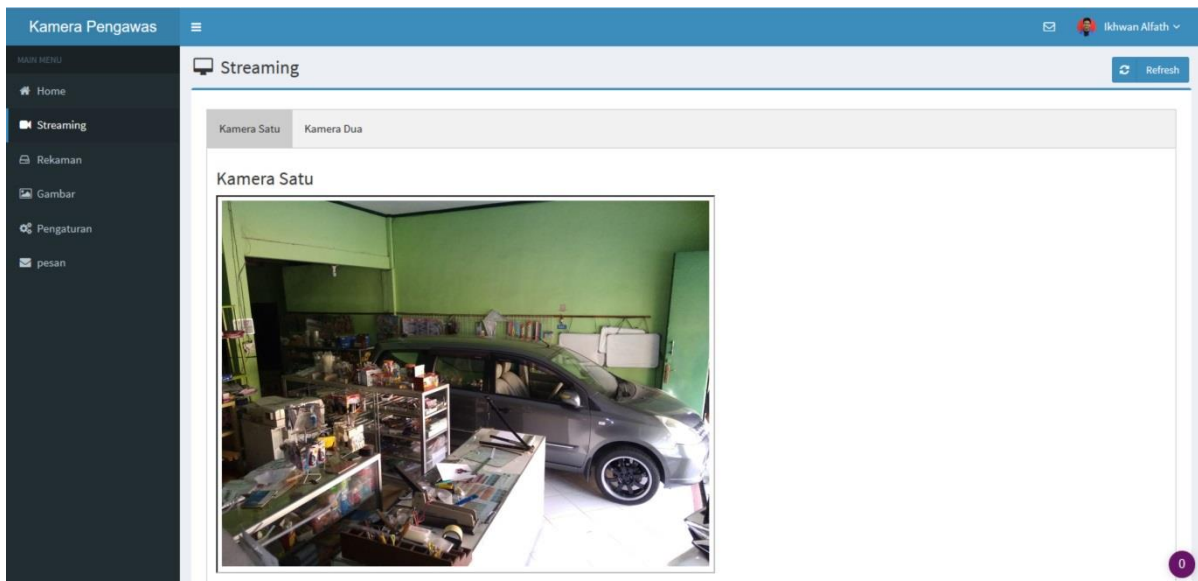
Gambar 4.4 Halaman Login Admin

Halaman login merupakan gerbang akses menuju halaman dashboard admin. Dalam alaman ini user atau yang bertindak sebagai admin harus memasukkan *username* dan *password* yang sesuai dengan data yang terdapat pada database.



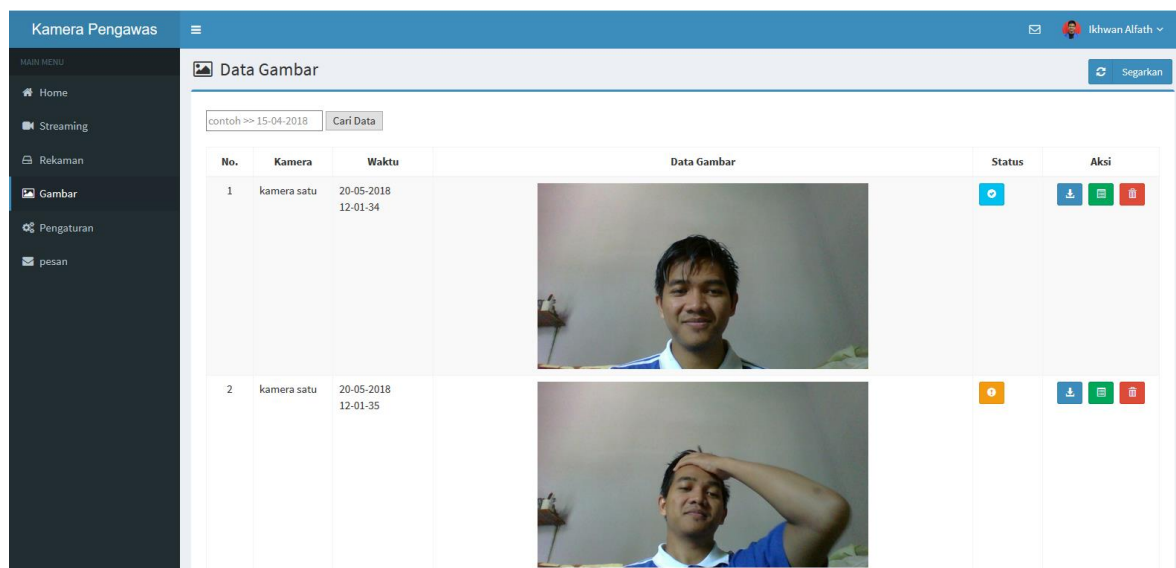
Gambar 4.5 Dashboard Halaman Admin

Halaman dashboard admin merupakan tampilan utama bagi admin untuk dapat mengakses seluruh data rekaman yang ada pada sistem. Pada halaman utama, admin dapat mengetahui informasi status pesan, informasi penggunaan media penyimpanan pada raspberry pi, dan admin juga dapat melihat informasi waktu.



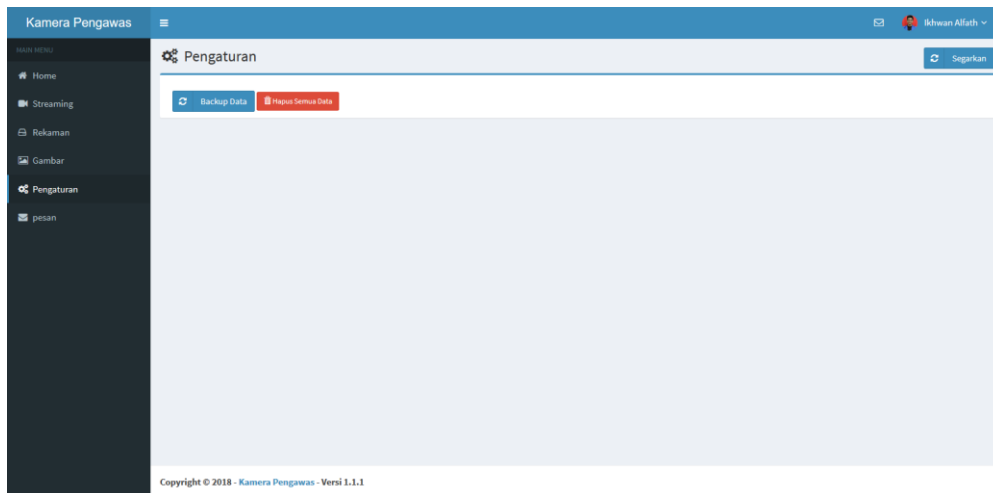
Gambar 4.6 *Dashboard Halaman Streaming*

Pada halaman *streaming* terdapat 2 pilihan yaitu kamera satu atau kamera dua, dengan pilihan tersebut maka admin dapat melihat rekaman secara langsung dari kamera satu atau kamera dua. Admin hanya dapat melihat di salah satu kamera. Hal ini diterapkan dengan tujuan untuk menghemat sumber daya daripada *raspberry pi*.



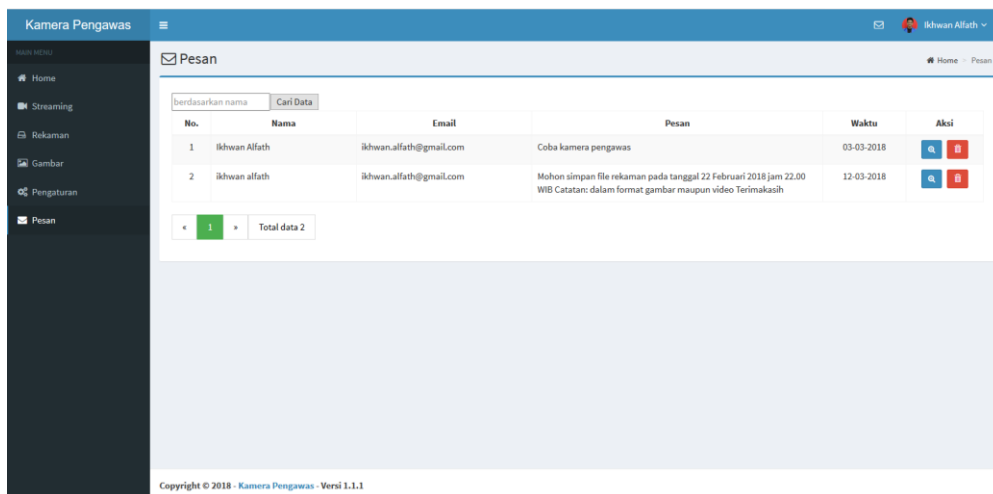
Gambar 4.7 *Dashboard Halaman Gambar*

Halaman gambar digunakan untuk menampilkan hasil tangkapan gambar berdasarkan pergerakan di sekitar lokasi kamera. Pada halaman ini terdapat tampilan status yang berarti keterangan gambar tersebut sudah ter-*backup* dalam *Google Drive*, telah ter-*upload*, ataupun gagal ter-*upload*. Kemudian dalam halaman ini juga terdapat beberapa aksi yang pertama admin dapat mendownload gambar, admin dapat melihat detail gambar, ataupun admin juga dapat menghapus gambar satu per satu.



Gambar 4.8 *Dashboard Halaman Pengaturan*

Halaman pengaturan ini terdapat 2 pilihan yaitu yang pertama untuk membuat salinan data rekaman gambar ataupun data rekaman video rekaman yang kemudian dapat di unduh dalam format kompresi .zip. Setelah yakin semua data gambar dan data rekaman terunduh, maka admin juga dapat melakukan penghapusan seluruh data rekaman gambar ataupun data rekaman video melalui menu hapus semua data.



Gambar 4.9 *Dashboard Halaman Pesan*

Halaman pesan ini untuk menampilkan pesan atau permintaan daripada pengguna lain yang tidak memiliki akses ke halaman dsashboard admin. Jadi pengguna lain dapat mengirimkan pesan permintaan kepada admin yang mana isinya tidak ditentukan, sehingga pesan ini bersifat umum. Ketika terdapat pesan masuk maka akan muncul notifikasi pada halaman dashboard admin. Sehingga admin dapat segera mengetahui ketika terdapat pesan masuk ke dalam sistem. Selain itu fitur pesan ini akan membantu admin dalam memilah dan memilih data rekaman yang perlu dibuat salinannya ataupun data yang dapat langsung dihapus.

4.2. Pembahasan Implementasi Penelitian

Implementasi merupakan tahap lanjutan untuk menjelaskan tentang apa yang telah dirancang pada bab sebelumnya. Implementasi ini akan menunjukkan apakah perancangan yang telah dilakukan sebelumnya akan dapat berjalan dan dapat digunakan dengan baik. Tahapan implementasi ini terbagi atas dua bagian utama yaitu implementasi perangkat keras (*hardware*) dan implementasi perangkat lunak (*software*).

4.2.1. Pembahasan Implementasi Perangkat Keras (Hardware)

a. Raspberry pi dengan memory penyimpanan internal 32 Gb

Dengan menggunakan raspberry pi maka akan ada beberapa kelebihan dan ada beberapa kekurangan. Diantara kelebihan itu yaitu dengan perangkat ini maka akan lebih menghemat konsumsi daya listrik. Bukan hanya itu, dengan menggunakan raspberry pi, pengguna juga tidak akan kesulitan meletakkan alat ini pada tempat tempat tersembunyi. Kemudian, dengan memory penyimpanan yang hanya 32 Gb, penulis harus memajemen penggunaan data seminimal mungkin. Oleh karenanya maka salah satu cara yang dapat dilakukan yaitu dengan menginstall sistem operasi linux raspbian stretch tanpa menggunakan Graphical User Interface (GUI) atau tanpa menggunakan tampilan antarmuka pengguna yang menggunakan metode ineraksi elektronik secara grafis. Sehingga hal ini akan lebih meringankan beban sumber daya perangkat dan akan lebih menghemat penggunaan penyimpanan data. Namun untuk melakukan perintah perintah eksekusi harus dengan menggunakan terminal pada sistem operasi linux tersebut.

b. Perangkat antena TP-Link outdoor (TL-WA7210N)

Perangkat ini dapat digunakan ketika rumah yang akan menggunakan sistem ini masih belum ter-*cover* oleh internet kabel. Oleh karenanya perlu menggunakan perangkat ini untuk menunjang kebutuhan internet untuk keperluan akses pencadangan data rekam gambar ke dalam akun Google Drive secara otomatis.

c. Switch hub

Perangkat switch dibutuhkan untuk keperluan koneksi kabel antar perangkat pendukung, yaitu perangkat antena outdoor yang berguna untuk menangkap sinyal internet lalu disambungkan dengan perangkat switch hub. Kemudian perangkat *Raspberry Pi*, kamera *IP*, dan laptop admin dapat dihubungkan menggunakan kebel *lan* dengan port *Rj45*. Dengan demikian, maka antar perangkat dapat saling terhubung satu sama lain. Sehingga sistem kamera pengawas dapat berjalan dengan semestinya.

d. Kamera *webcam*

Dengan menggunakan kamera webcam maka pengguna dapat dengan mudah langsung mengakses kamera ini dengan menggunakan port usb.

e. Kamera *ipcam*

Dengan menggunakan kamera ip, pengguna tetap dapat memantau dengan jelas tangkapan gambar ketika malam hari atau minim cahaya. Karena pada kamera ini terdapat lampu infrared yang berfungsi untuk memberikan efek cahaya tambahan sehingga tangkapan gambar tetap terlihat jelas.

4.2.2. Pembahasan Implementasi Perangkat Lunak

Implementasi perangkat lunak yang dilakukan di antaranya akses kamera webcam, akses ip kamera, akses *Google Drive Api*, komunikasi *streamming* video rekaman.

a. Akses Kamera Webcam disertai *Motion Detection*

Dalam mengakses kamera webcam menggunakan program *python* perlu memperhatikan perangkat yang digunakan, karena untuk mengakses perangkat kamera ini tidak sama dengan semua perangkat, yang mana bergantung pada koneksi atau konektor yang digunakan. Sub bab ini terdapat kode program untuk mengakses kamera webcam dengan menggunakan pemrograman *python 2.7* disertai dengan logika *motion detection* seperti pada gambar 4.10 di bawah ini:

```
class VideoCamera(object):
    def __init__(self):
        self.cam = cv2.VideoCapture(0) # akses kamera 0
        self.cam.set(3, 960) //tinggi
        self.cam.set(4, 720) //lebar
        self.waktu = datetime.now().strftime("%d-%m-%Y %H-%M-%S")

    def start(self):
        p2 = Thread(target=self.motion, args=())
        p2.daemon = True
        p2.start()
        return self

    def motion(self):
        batas = 239000
        cek_waktu = datetime.now().strftime('%Ss')
        t_minus = cv2.cvtColor(self.cam.read()[1], cv2.COLOR_RGB2GRAY)
        t = cv2.cvtColor(self.cam.read()[1], cv2.COLOR_RGB2GRAY)
        t_plus = cv2.cvtColor(self.cam.read()[1], cv2.COLOR_RGB2GRAY)
        while True:
            jam = datetime.now().strftime("%d-%m-%Y %H-%M-%S")
            # total perbedaan nilai warna
```

```

totalDiff = cv2.countNonZero(self.diffImg(t_minus, t, t_plus))
text = "threshold: " + str(totalDiff)
cv2.putText(self.frame, text, (20,40), cv2.FONT_HERSHEY_SIMPLEX, 1,
(0,0,0), 2)
#print(text)
if totalDiff > batas and cek_waktu !=
    datetime.now().strftime('%Ss'):
    dimg = self.cam.read()[1]
    cv2.imwrite('../penyimpanan/gambar/' +
str(jam) + '.jpg'), dimg)
gambar = (str(jam)+'.jpg')
print(gambar)
dataKiriman = [('kamera','kamera
satu'),('tanggal',jam),('gambar',gambar)]
dataKiriman = urllib.urlencode(dataKiriman)
path='http://localhost/kamera-pengawas/python-
alfath/penerima.php'
req = urllib2.Request(path, dataKiriman)
page= urllib2.urlopen(req).read()
cek_waktu = datetime.now().strftime('%Ss')
t_minus = t
t = t_plus
t_plus = cv2.cvtColor(self.cam.read()[1],
cv2.COLOR_RGB2GRAY)
key = cv2.waitKey(10)
if key == ord('q'):
    cv2.destroyAllWindows()
    # cv2.destroyWindow(winName)
    break
return self

```

Gambar 4.10 Kode Program akses kamera webcam disertai Motion Detection

b. Akses Ip Kamera

Dalam mengakses ip kamera sedikit berbeda dengan cara mengakses kamera webcam.

Untuk mengakses kamera ini dapat dilihat pada gambar 4.11 dibawah ini:

```

class IPCAM(object):
    def __init__(self):
        self.url =
'http://192.168.1.16:80/webcapture.jpg?command=snap&channel=1'
        self.cam = cv2.VideoCapture(self.url)
        (self.grabbed, self.frame) = self.cam.read()
        self.waktu = datetime.now().strftime("%d-%m-%Y %H:%M:%S")

    def start(self):
        p6 = Thread(target=self.kamera_ip, args=())
        p6.daemon = True
        p6.start()
        return self

```

```

def kamera_ip(self):
    width = int(self.cam.get(cv2.CAP_PROP_FRAME_WIDTH) + 0.5)
    height = int(self.cam.get(cv2.CAP_PROP_FRAME_HEIGHT) + 0.5)
    fourcc = cv2.VideoWriter_fourcc(*'XVID')
    out = cv2.VideoWriter('../recording/' +
        str(self.waktu)+'.avi', fourcc, 20.0, (width, height))
    video = (str(self.waktu)+'.avi')
    print(video)
    jam = datetime.now().strftime("%d-%m-%Y %H:%M:%S")
    dataKiriman = [('kamera', 'Kamera Dua -
        IPCAM'), ('tanggal', jam), ('video', video)]
    dataKiriman = urllib.urlencode(dataKiriman)
    path='http://localhost/kamera-pengawas/python-
        alfath/penerima-video.php'
    req = urllib2.Request(path, dataKiriman)
    page= urllib2.urlopen(req).read()

    while True:
        imgResp = urllib.urlopen(self.url)
        imgNp = np.array(bytearray(imgResp.read()), dtype=np.uint8)
        img = cv2.imdecode(imgNp, -1)

        while(self.cam.isOpened()):
            if self.grabbed == True:
                out.write(img)
                if (cv2.waitKey(1) & 0xFF) == ord('q'):
                    break
            else:
                break

        out.release()
        self.cam.release()
        cv2.destroyAllWindows()

```

Gambar 4.11 Kode Program akses ip kamera dengan hasil format .avi

c. Komunikasi *Google Drive API* (Application Programming Interface)

Salah satu layanan yang disediakan oleh *Google* ini merupakan sebuah layanan yang dapat digunakan untuk menyimpan berkas ke dalam google drive melalui sistem yang kita buat. Untuk dapat mengakses google drive dari akun pribadi, maka diperlukan kunci atau *key* yang bisa didapat melalui dashboard *Google Drive Api* atau halaman *console developers* (<https://console.developers.google.com/apis/credentials>) yang nantinya dapat di *download* dengan format kunci *.json*. Sehingga layanan *google* ini dapat berinteraksi langsung dengan sistem yang sedang di buat dengan standar keamanan seperti yang diberikan oleh layanan seperti pada gambar 4.12 di bawah:

The screenshot shows the Google APIs console interface. At the top, there is a navigation bar with the Google APIs logo and the project name 'ikhwan-uploader'. Below this, there is a header for the 'Client ID for Other' section, with options to 'DOWNLOAD JSON', 'RESET SECRET', and 'DELETE'. The main content area displays the following information:

Client ID	136766217263-74od8b23bsm7e8o1co6giqg48of7vp4p.apps.googleusercontent.com
Client secret	wF5WQfK02ShetUeH9B80cWxv
Creation date	Jan 11, 2018, 8:37:54 AM

Below the table, there is a 'Name' field with a help icon, containing the text 'ikhwan-skripsi'. At the bottom, there are 'Save' and 'Cancel' buttons.

Gambar 4.12 Kunci rahasia untuk akses Google Drive Api

```

class UploadGoogleDrive(object):
    def __init__(self):
        self.SCOPEES = 'https://www.googleapis.com/auth/drive.file'
        self.store = file.Storage('storage.json')
        self.creds = self.store.get()

    def start(self):
        p5 = Thread(target=self.upload, args=())
        p5.daemon=True
        p5.start()
        return self

    def argumen(self):
        try :
            import argparse
            self.flags =

argparse.ArgumentParser(parents=[tools.argparser]).parse_args()
        except ImportError:
            self.flags = None

    def upload(self):
        if not self.creds or self.creds.invalid:
            print("buat penyimpanan file data baru")
            flow = client.flow_from_clientsecrets
                ('client_secret.json', self.SCOPEES)

            self.creds = tools.run_flow(flow, store, self.flags) \
                if self.flags else tools.run(flow,
self.store)
            DRIVE = build('drive', 'v3',
http=self.creds.authorize(Http()))
            tmp=0

```



```

while True:
    time.sleep(.1)
    db = MySQLdb.connect("localhost","root","PASS",
        "kamera-pengawas")
    cursor = db.cursor()
    sql = "select gambar from gambars order by gambar_id
        desc limit 1"

    cursor.execute(sql)
    results = cursor.fetchall()
    for row in results:
        data = row[0]
        if tmp != data:
            tmp = data
            cek = True
        if cek == True:
            FILES = (
                ('../pictures/'+data),
            )
            for file_title in FILES:
                file_name = file_title
                metadata = {'name': file_name,
                    'mimeType': None
                }
                res = DRIVE.files().create(body=metadata,
                    media_body=file_name).execute()
                if res:
                    file_name = file_title
                    print('Uploaded "%s" (%s)' % (file_name,
                        res['mimeType']))
                    #print('\n Upload berhasil')
                cek = False
            key = cv2.waitKey(10)

            if key == 27:
                cv2.destroyWindow(winName)
                break
    db.close()
return self

```

Gambar 4.13 Kode Program akses Google Drive Api

d. Komunikasi streaming video rekaman

Dalam sub bab ini terdapat kode program untuk mengakses video rekaman secara *realtime (streaming)*. Dengan menggunakan bantuan library flask server, sehingga video realtime dapat di akses dari halaman dashboard. Seperti tampilan pada gambar 4.4 dibawah ini:

```

app = Flask(__name__)
@app.route('/')
def index():
    return render_template('index.html')

def bacakamera(kamera):
    while True:
        frame = vs.baca()
        time.sleep(1)
        yield (b'--frame\r\n'
              b'Content-Type: image/jpeg\r\n\r\n' + frame +
              b'\r\n\r\n')

@app.route('/video_feed')
def video_feed():
    return Response(bacakamera(VideoCamera()),
                    mimetype='multipart/x-mixed-replace;
                    boundary=frame')

if __name__ == '__main__':
    logging.basicConfig(format='%(asctime)s %(message)s',
                        filename='kamerapengawas.log', level=logging.INFO)
    app.run(host='0.0.0.0', port=8081)

```

```

index.html
<html>
<!-- <head>
    <title>Video Streaming</title>
</head> -->
<body>
    
</body>
</html>

```

Gambar 4.14 Kode Program akses streaming video rekaman

4.2.3. Pembahasan Modul Pada Perangkat Lunak

a. Modul *python* untuk akses kamera dan *motion detection*

Untuk keperluan akses kamera dengan bahasa pemrograman python 2.7 maka perlu mengimport modul python yang telah dikembangkan oleh programmer *python* di seluruh dunia. Oleh karenanya dalam pembuatan sistem ini penulis perlu meng-*import* beberapa modul di antaranya:

1. Import cv2

Modul cv2 merupakan salah satu modul dari openCV (Open Source Computer Vision) yang dapat digunakan oleh beberapa kode pemrograman salah satunya pemrograman python. Modul cv2 sendiri berfungsi untuk mengolah gambar baik video maupun gambar. Oleh karenanya untuk dapat mengakses kamera pada pemrograman python, salah satunya yaitu dapat menggunakan modul cv2 ini.

2. Import Numpy

Numpy adalah modul untuk komputasi ilmiah dalam pemrograman python. Dengan menggunakan modul ini, maka dapat mendukung perhitungan, di sini konteksnya yaitu mengenai perubahan warna yang didapat dari hasil tangkapan gambar melalui kamera.

3. Import sys

Sys merupakan modul yang dapat digunakan untuk manajemen proses, manajemen memori hingga manajemen interupsi perangkat keras atau untuk melakukan penghentian proses untuk melepas perangkat keras.

4. Import urllib2

Urllib 2 pada sistem ini digunakan untuk melakukan komunikasi dan pengiriman data yang berasal dari proses – proses pemrograman python yang kemudian menghasilkan data untuk disimpan ke dalam web ataupun basisdata (Begoli, 2005).

5. Import time

Modul time di sini juga penting untuk memberikan judul pada setiap data yang akan disimpan.

6. Import Threading

Dengan adanya modul *threading* maka penulis dapat melakukan konkurensi atau melakukan proses secara bersamaan dalam mengeksekusi sebuah operasi. Dengan adanya *threading* ini, maka penulis dapat manajemen proses yang berjalan hampir beriringan. Dengan bantuan *threading* ini juga penulis cukup terbantu untuk dapat manajemen sumber daya daripada perangkat keras.

- b. Modul *python* untuk akses *Google Drive* dengan *Api*

1. Modul *googleapiclient.discovery*

Agar dapat menggunakan layanan *Api Google*, maka perlu menginstall *Google Api client* yang mana nantinya dapat digunakan untuk menghubungkan antara sistem

atau aplikasi yang kita buat dengan layanan yang diberikan oleh *Google*. Dalam kasus penelitian ini yaitu untuk mengakses layanan *Google Drive*.

2. Modul *httplib2*

Modul *httplib2* merupakan salah satu *library* yang dapat digunakan untuk melakukan *http request*. Dalam kasus penelitian ini yaitu untuk meminta *verifikasi* pada layanan *Google*, agar dapat menggunakan layanan *Google Drive*.

3. Modul *Oauth2client*

Modul *oauth2client* ini digunakan untuk melakukan proses standar keamanan *Google* yang mana untuk memastikan bahwa yang sedang menggunakan layanan tersebut merupakan pengguna yang sah berdasarkan kode kunci yang sudah di berikan sebelumnya.

4. Modul *Client*

Modul *client* di sini berfungsi untuk memastikan bahwa aplikasi yang dibangun memang sebagai *client* sehingga nantinya dapat memanfaatkan layanan yang ditawarkan *server* melalui *socket* yang sudah dibangun. *Socket* sendiri merupakan sebuah endpoint untuk melakukan komunikasi dalam jaringan. Sepasang proses atau thread berkomunikasi dengan membangun sepasang *socket* yang masing masing proses memilikinya. *Socket* dibuat dengan menyambungkan dua buah alamat ip melalui port tertentu. Secara umum *socket* digunakan dalam *client/server* yang menghubungi *server* maka *server* akan mentetujui komunikasi dengan *client* melalui *socket* yang dibangun (Alwand, 2013).

c. Modul python untuk akses *video* langsung (*streaming video*)

1. Modul *logging*

Modul *logging* ini dapat berguna untuk melakukan pencatatan mengenai apa saja yang dilakukan oleh proses di bawahnya. Dalam penelitian ini digunakan untuk melakukan pencatatan apabila terdapat *error* pada komunikasi *video streaming*.

2. Modul *flask*

Modul *flask* merupakan salah satu *micro web framework* yang ditulis dalam bahasa pemrograman *python*. Dalam penelitian ini digunakan untuk menampilkan *video* secara langsung atau *streaming video*. Dengan menggunakan *flask server* maka proses siaran langsung ini dapat diakses melalui alamat *ip* ataupun *port* yang sudah ditentukan sebelumnya.

3. Modul *render_template*

Modul *render_template* dapat digunakan untuk menampilkan hasil ke dalam file .html dengan menggunakan *flask*.

4.3. Pengujian Penelitian

Tahap pengujian penelitian dilakukan untuk mengetahui apakah sistem yang dibuat dapat berjalan dengan baik dan lancar. Dengan dilakukan pengujian ini maka akan dapat ditemukan kekurangan serta kelebihan daripada sistem kamera pengawas ini. Pengujian dilakukan dengan berbagai tahap di antaranya:

- a. Melakukan uji koneksi antara kamera dengan raspberry pi
- b. Melakukan uji *motion detection* kamera *usb (webcam)*
- c. Melakukan uji *auto backup cloud (Google Drive)*
- d. Melakukan uji *streaming* video sistem
- e. Melakukan uji pengiriman pesan user
- f. Melakukan uji backup data rekaman
- g. Melakukan uji penghapusan data rekaman

4.4. Hasil Pengujian Sistem

Pengujian sistem telah dilakukan yaitu dengan menyebarkan kuesioner dengan melibatkan 20 orang responden. Berdasarkan data yang di dapat, kemudian untuk menghitung hasil dengan menggunakan metode skala Likert.

4.4.1. Perhitungan Skala Likert

Skala Likert atau Likert skale merupakan salah satu skala penelitian yang digunakan untuk mengukur sikap dan pendapat. Dengan skala likert ini, responden diminta untuk melengkapi kuisisioner yang mengharuskan mereka untuk menunjukkan tingkat persetujuan terhadap serangkaian pertanyaan. Pertanyaan atau pernyataan yang digunakan dalam penelitian ini biasanya disebut dengan variabel penelitian dan ditetapkan secara spesifik oleh peneliti.

Untuk mendapatkan hasil interpretasi, harus diketahui dulu skor tertinggi (X) dan angka terendah (Y) untuk item penilaian dengan rumus sebagai berikut :

Y = Skor tertinggi likert x jumlah responden (Angka Tertinggi 5) "Perhatikan Bobot Nilai"

X = Skor terendah likert x jumlah responden (Angka Terendah 1) "Perhatikan Bobot Nilai"

Rumus Indeks (%) = (Total Skor / Skor Maksimum) x 100

Tabel 4.1 Tabel Prosentase

Jawaban	Keterangan
0% - 19.99%	Sangat Tidak Setuju
20% - 39.99%	Tidak Setuju
40% - 59.99%	Cukup Setuju atau Netral
60% - 79.99%	Setuju
80% - 100%	Sangat Setuju

4.4.2. Hasil Total Skala Likert

Pada tahapan ini telah diketahui data yang didapat dari kuesioner yang telah dibagikan dengan skor sebagai berikut : Sangat Setuju/SS (Skor 5), Setuju/S (Skor 4), Cukup Setuju/CS (Skor 3), Tidak Setuju/TS (Skor 2), dan Sangat Tidak Setuju/STS (Skor 1). Berdasarkan skor yang ada maka akan hasil perhitungan seperti pada tabel 4.2

Tabel 4.2 Tabel Pengujian Skala Likert

NO	URAIAN	SKOR					Total Skor	
		1	2	3	4	5	Total	%
A.	MANFAAT							
1.	Sistem kamera pengawas ini bermanfaat untuk masyarakat umum di Indonesia?	0	0	1	5	14	93	93 %
2	jika sistem ini diterapkan di lingkungan anda maka dapat membantu mengurangi tindak kejahatan di sekitar anda	0	0	2	3	15	93	93 %
3	Fitur auto backup cloud (<i>Google Drive</i>) dapat membantu mengamankan barang bukti jika terdapat kelalaian baik sengaja ataupun tidak sengaja	0	0	1	3	16	95	95 %
4	Fitur hapus semua data dapat bermanfaat dan menghemat waktu admin untuk manajemen data rekaman baik berupa video maupun gambar	0	0	2	9	9	87	87 %
5	Fitur pesan dari sisi pengguna dapat membantu pengguna untuk memberikan saran ataupun permintaan kepada admin	0	0	1	9	10	89	89 %

NO	URAIAN	SKOR					Total Skor	
		1	2	3	4	5	Total	%
A.	MANFAAT							
6	Fitur backup dengan format .zip dapat membantu admin untuk mengarsipkan data rekaman	0	0	1	9	10	89	89 %
B.	TAMPILAN							
1.	Dari sisi pengguna tampilan menarik	0	0	3	7	10	87	87 %
2.	Video player pada halaman admin tampilan menarik	0	0	4	9	7	83	83 %
3.	Dari sisi Dashboard Admin tampilan menarik	0	0	4	7	9	85	85 %
C.	FUNGSIONALITAS							
1.	Fitur pendeteksi pergerakan (motion detection) dapat berfungsi sebagaimana mestinya		0	1	13	6	85	85 %
2.	Fitur auto backup data gambar .jpg ke akun <i>Google Drive</i> dapat berjalan secara otomatis	0	0	1	12	7	86	86 %
3.	Fungsi pengarsipan berkas dengan format .zip dapat di jalankan	0	0	1	15	4	83	83 %
4.	Perpesanan antara pengguna dengan admin dapat berjalan normal	0	0	2	7	11	89	89 %
5.	Info pengecekan penyimpanan lokal dapat bekerja dengan baik	0	0	1	9	10	89	89 %
6.	Fungsi untuk melihat video rekaman secara langsung (streaming) dapat berjalan normal	0	0	3	4	13	90	90 %
7.	Fungsi hak akses admin dapat beroperasi normal, jika pengguna benar-benar memiliki akses maka akan disetujui, namun jika tidak maka sebaliknya.	0	0	3	4	13	90	90 %
Total							88,3	88,3%

Berdasarkan perhitungan di atas tersebut, maka penulis dapat menyimpulkan bahwasannya tanggapan responden mayoritas mendukung untuk pengembangan lebih lanjut pada sistem aplikasi kamera pengawas ini, dapat dilihat berdasarkan hasil perhitungan rata rata pada tabel di atas yaitu sebesar 88,3125 yang mana berdasarkan pada table prosentase yaitu sangat setuju.

4.5. Analisis Penelitian

Pada tahap analisis penelitian ini merupakan tahapan di mana sebuah penelitian dinilai kelebihan serta kekurangannya setelah dilakukan tahap pengujian. Setiap kelebihan dan kekurangan dapat menjadi bahan pertimbangan bagi penulis dan pembaca untuk menilai keberhasilan dari penelitian ini. Kekurangan dari penelitian ini juga diharapkan akan menjadi pertimbangan untuk pengembangan lebih lanjut demi tercapainya sistem yang sempurna di masa mendatang. Oleh karenanya maka di bawah ini merupakan kelebihan serta kekurangan pada penelitian ini antara lain:

a. Kelebihan Sistem

1. Sistem dapat melakukan kontrol data rekaman yang tersimpan pada perangkat raspberry pi
2. Sistem dapat memajemen pesan masuk dari user
3. Sistem dapat menjadi perantara *interface* atau tampilan utama bagi pengguna
4. Sistem tetap dapat berjalan walau tidak terdapat koneksi internet
5. Sistem dapat diakses jarak jauh atau *remote* dari *internet ip public forwarding*
6. Sistem ini dapat dijalankan pada perangkat *raspberry pi* yang hemat listrik

b. Kekurangan Sistem

1. Tampilan pada halaman admin untuk menampilkan gambar masih sangat sederhana, perlu ditambahkan menu untuk melihat detail gambar untuk lebih menarik lagi.
2. Sistem kamera pengawas ini hanya dapat melakukan *backup* secara otomatis dalam *Google Drive* saja.
3. Akses kamera hanya dapat mengakses dua kamera yang sudah diatur sebelumnya.
4. Akses admin masih belum menggunakan multi akses atau *multi level admin*
5. Batasan mode *motion detection* masih statis sesuai yang sudah ditentukan sebelumnya.
6. Untuk melakukan pencadangan dan penghapusan data masih dengan mode manual dilakukan oleh admin.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penelitian di atas, maka dapat diambil kesimpulan bahwasannya:

- a. Sistem kamera pengawas ini dapat mengakses kamera usb (webcam) dan kamera internet protocol (*IP*) dengan menggunakan *Raspberry Pi*.
- b. Sistem ini dapat membuat file gambar dengan format .jpg berdasarkan pergerakan yang terdeteksi oleh kamera (*motion detection*).
- c. hasil gambar yang didapatkan dari (*motion detection*) kemudian dapat dicadangkan secara otomatis ke dalam akun *Google Drive* selama terdapat koneksi internet pada perangkat *raspberry pi*.

5.2 Saran

Setelah dilakukan penelitian ini, penulis menyadari masih terdapat beberapa kekurangan daripada sistem ini sebagaimana sudah dijelaskan di atas, maka diharapkan pada penelitian selanjutnya dapat menghilangkan atau menutupi kekurangan yang ada pada sistem ini. Bahkan sangat diharapkan penelitian selanjutnya dapat mengembangkan sistem ini untuk dapat dimanfaatkan masyarakat secara luas. Saran untuk penelitian selanjutnya adalah sebagai berikut:

- a. Sistem kamera pengawas ini sudah terdapat fitur yang dapat di jalankan pada *Raspberry Pi* namun dengan terbatasnya penyimpanan akan menjadi hambatan pada sisi perekaman video, sehingga lebih baik di tambahkan hardisk eksternal untuk menambah kapasitas penyimpanan yang lebih besar.
- b. Saat ini penulis menggunakan kamera *usb* yang dapat digunakan di siang hari saja, mungkin akan lebih baik jika pengguna menggunakan modul kamera *usb* yang dilengkapi dengan fitur *night vision*, sehingga kamera tetap dapat kelihatan dengan jelas ketika di malam hari.
- c. Pada sistem ini sudah menggunakan modul *video player standart* untuk pemutaran video, alangkah baiknya jika terdapat fitur *crop* video yang langsung bisa di unduh.
- d. Mode penghapusan dan pencadangan data pada penyimpanan lokal sudah dapat dilakukan secara manual oleh admin, penulis berharap kedepannya dapat dilakukan secara otomatis dengan mode penjadwalan.

DAFTAR PUSTAKA

- Awim. (2018). *Client-side web APIs*. Dipetik Juli 10, 2018, dari https://developer.mozilla.org/id/docs/Learn/JavaScript/Client-side_web_APIs
- Begoli, E. (2005). *Connecting to Mysql From Python*. Dipetik Januari 28, 2018, dari <https://it.toolbox.com/blogs/edmonbegoli/connecting-to-mysql-from-python-060805>
- Beizer, B. (1995). *Black – Box Testing: Techniques for Functional Testing of Software and System*. John Willey & Sons, Inc.
- Choizes. (2018). *Pengertian Skala Likert dan Contoh Cara Hitung Kuesionernya*. Dipetik Juli 10, 2018, dari <https://www.diedit.com/skala-likert/>
- Falahah, A. K. (2012). *Pengukuran Usability Sistem Menggunakan User Questionnaire*. Yogyakarta: Seminar Nasional Aplikasi Teknologi Informasi (SNATI).
- Fowler, M. (2005). *UML Distilled Edisi 3*. Yogyakarta: Andi.
- Grossman, T. F., George, Attar, & Ramtin. (2009). *A Survey of Software Learnability: Metrics, Methodologies and Guidelines*. Boston: Autodesk Research.
- Jay. (2015). *Membuat Web App dari Awal Menggunakan Python Flask dan Mysql*. Dipetik Juli 10, 2018, dari <https://code.tutsplus.com/id/tutorials/creating-a-web-app-from-scratch-using-python-flask-and-mysql--cms-22972>
- Khosravi, K. G. (2004). *A Quality Model for Design Patterns*. University of Montreal.
- Nielsen, J. (1993). *Usability Engineering*. Dipetik Pebruari 1, 2018, dari http://www.academia.edu/3770217/EVALUASI_USABILITY
- Ottensooser, A. B. (1999). *Determination of Software Functionality*. US: Bunker Trust Australia Limited.
- P Botella, X. B. (2004). *ISO / IEC 9126 in Practice: What Do We Need to Know*. In *Proceedings of The 1st Software Measurement European Forum*.
- Pressman, R. (2002). *Rekayasa Perangkat Lunak Pendekatan Praktisi*. Yogyakarta: Andi.
- Somba, R. (2018). *Polda Bali Dinilai Sukses Berantas Premanisme*. Dipetik Juli 10, 2018, dari <http://www.validnews.co/Polda-Bali-Dinilai-Sukses-Berantas-Premanisme-EnI>

Tullis, Tom, & Albert, B. (2008). *Measuring The User Experience*. Morgan Kauffman.

Upton, L. (2013). *Turn your Pi into a low-cost HD surveillance cam*. Dipetik Januari 8, 2018, dari <https://www.raspberrypi.org/blog/turn-your-pi-into-a-low-cost-hd-surveillance-cam/>

Warsito, G. (2015). *Memanfaatkan ip camera di rumah jakarta*. Dipetik Pebruari 5, 2018, dari https://www.kompasiana.com/gunklaten/memanfaatkan-ip-camera-di-rumah-jakarta_54f3d259745513a12b6c7f80

Zrymiak, D. (2001). *Software Quality Function Deployment*. Dipetik Pebruari 12, 2018, dari http://www.isixsigma.com/index.php?option=com_k2&view=item&id=1384:software-quality-function-deployment&Itemid=213

LAMPIRAN