

**SISTEM PENJADWALAN MAGIC MIRROR  
MENGUNAKAN RASPBERRY PI**



Disusun Oleh:

N a m a : Muhammad Ryan Apriyudan

NIM : 14523260

**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA**

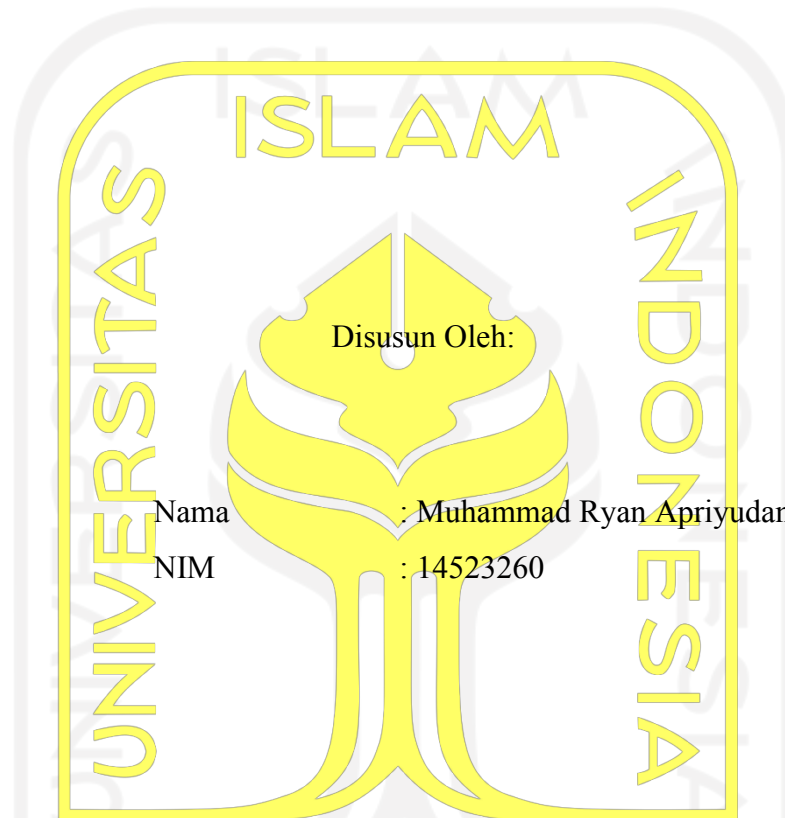
**2021**

**HALAMAN PENGESAHAN DOSEN PEMBIMBING**

**SISTEM PENJADWALAN MAGIC MIRROR**

**MENGGUNAKAN RASPBERRY PI**

**TUGAS AKHIR**



Nama : Muhammad Ryan Apriyudan  
NIM : 14523260

الجامعة الإسلامية  
Yogyakarta, 10 September 2021

Pembimbing,

(Muhammad Andri Setiawan, S.T. , M.S.c., Ph.D. )

**HALAMAN PENGESAHAN DOSEN PENGUJI**

**SISTEM PENJADWALAN MAGIC MIRROR  
MENGUNAKAN RASPBERRY PI**

**TUGAS AKHIR**

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

Yogyakarta, 10 September 2021

Tim Penguji

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

**Anggota 1**

Ari Sujarwo, S.Kom., MIT.

**Anggota 2**

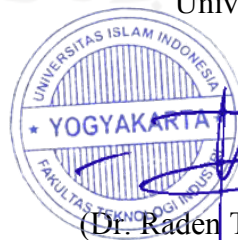
Sri Mulyati, S.Kom., M.Kom.

الجمعة الائمة الاندية  
Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



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

## HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Muhammad Ryan Apriyudan

NIM : 14523260

Tugas akhir dengan judul:

### **SISTEM PENJADWALAN MAGIC MIRROR MENGUNAKAN RASPBERRY PI**

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 saya sendiri, Tugas Akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apa pun.

Demikian surat pernyataan ini di buat, semoga dapat di pergunakan sebagaimana mestinya.

Yogyakarta, 10 September 2021



(Muhammad Ryan Apriyudan)

## HALAMAN PERSEMBAHAN

Dengan mengucapkan puji Syukur alhamdulillah, penulis mempersembahkan Tugas Akhir ini Kepada:

1. Terima kasih untuk Ayah dan Mama yang telah memberi semangat dan doa dalam mengerjakan Tugas Akhir ini sehingga dapat terselesaikan.
2. Terima Kasih untuk mbak Ayuk dan adik Githa yang selalu memberikan saran dan dukungan dalam mengerjakan Tugas Akhir ini.
3. Teman - teman belajar dunia teknologi Mbah, Titik, dan Paan yang telah memberi ilmu dan motivasi selama ini.
4. Teman - teman seperjuangan Dimas, Gilang, Bayu, Adzan, Unyil, dan Hilman yang sudah sangat membantu Tugas Akhir ini sampai selesai.
5. Rekan - rekan Iin manajemen yang selalu mendoakan dan memberi dukungan dalam mengerjakan Tugas Akhir ini.
6. Seluruh Keluarga besar Informatika UII yang sudah membantu selama masa perkuliahan hingga sampai pada saat ini.
7. Kepada semua pihak yang tidak dapat penulis sebutkan satu per satu yang telah membantu dalam pengerjaan Tugas Akhir ini. Terimakasih banyak atas dukungan dan doanya.
8. Terima kasih kepada Muhammad Ryan Apriyudan yang sudah berjuang dan melewati berbagai macam proses hingga akhirnya dapat menyelesaikan Tugas Akhir ini.

**HALAMAN MOTO**

“I know the sky is not the limit because there are footprints on the Moon –  
and I made some of them  
(Buzz Aldrin)”



## KATA PENGANTAR

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

1. Prof. Fathul Wahid, S.T., M.Sc., Ph. D. Selaku Rektor Universitas Islam Indonesia.
2. Prof. Dr. Ir. Hari Purnomo selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
3. Dr. Raden Teduh Dirgahayu, S. T., M.Sc. Selaku Ketua Program Studi Informatika – Program Sarjana Universitas Islam Indonesia.
4. Hendrik, S.T., M.Eng. Selaku Ketua Program Studi Informatika Universitas Islam Indonesia.
5. Muhammad Andri Setiawan, S.T, M.Sc., Ph.D., Selaku dosen pembimbing yang telah memberikan waktu, tenaga, pikiran, semangat dan selalu memberikan pembelajaran dalam menyelesaikan Tugas Akhir ini.
6. Kedua orang tua yang selalu mendukung, mendoakan dan selalu percaya kepada anaknya sehingga Tugas Akhir ini dapat terselesaikan.

Tentunya sebagai manusia yang tidak pernah lepas dari kesalahan, sehingga dalam penyusunan tugas akhir ini masih banyak terdapat kesalahan dan kekurangan. Oleh karena itu penulis memohon maaf atas segala kesalahan dan kekurangan yang ada. Semoga Tugas Akhir ini bermanfaat bagi kampus Universitas Islam Indonesia, semoga Allah SWT selalu meridai kita menjadi lebih baik lagi. Amiin ya rabbal alamin.

Yogyakarta, 10 September 2021



(Muhammad Ryan Apriyudan)

## SARI

Perkembangan teknologi informasi pada saat ini terus berkembang seiring dengan kebutuhan manusia yang menginginkan kemudahan, kecepatan dan keakuratan dalam memperoleh informasi. Salah satu konsep dari penerapan ke perangkat lain dengan memanfaatkan jaringan sehingga menghasilkan keluaran yang berupa informasi dikenal sebagai konsep *Internet of Things (IoT)*.

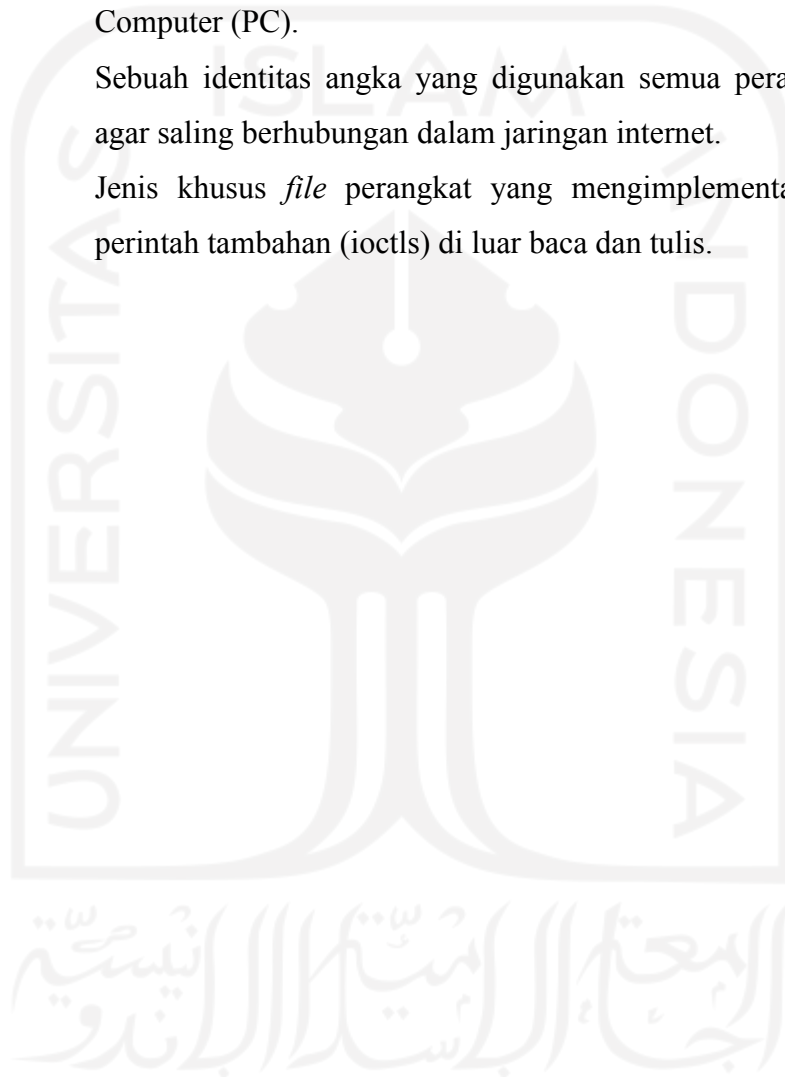
Dalam upaya untuk memenuhi kebutuhan akan informasi tersebut, maka diperlukan sebuah alat yang dapat mengatasi masalah ini dengan menggunakan konsep *Internet of Things*. *Magic Mirror* merupakan sebuah alat berbasis *Raspberry Pi* yang dapat menampilkan informasi penjadwalan meeting, dan dapat diintegrasikan dengan Google Calendar API dalam penerapannya. Pada penelitian ini menggunakan metode *Waterfall*, Adapun hasil yang diperoleh pada tahap pengujian terlihat bahwa sistem magic mirror sudah cukup diterima oleh pengguna. Hal ini dikarenakan hasil rata-rata yang didapat dari 10 responden menunjukkan persentase 88.2% di mana dalam variable skala likert jawaban SS (Sangat Setuju) diberi rentang nilai 81% - 100%.

Kata kunci: *Magic Mirror, Raspberry Pi, Google Calendar Api, Internet of Things*.



## GLOSARIUM

<i>Access Point</i>	Perangkat jaringan yang berisi penerima dan antena untuk mengirim sinyal dan menerima sinyal ke dan dari <i>client</i> atau pengguna.
<i>Software</i>	Merupakan suatu konstruksi atau bangunan dari sebuah program yang dirancang sedemikian rupa, serta diatur untuk suatu sistem kerja yang dijelankannya
<i>Hardware</i>	Komponen fisik yang membentuk satu kesatuan sistem Personal Computer (PC).
<i>IP Address</i>	Sebuah identitas angka yang digunakan semua perangkat <i>computer</i> agar saling berhubungan dalam jaringan internet.
Terminal	Jenis khusus <i>file</i> perangkat yang mengimplementasikan sejumlah perintah tambahan ( <i>ioctl</i> s) di luar baca dan tulis.



## DAFTAR ISI

SISTEM PENJADWALAN MAGIC MIRROR MENGGUNAKAN	
RASPBERRY PI .....	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING .....	ii
HALAMAN PENGESAHAN DOSEN PENGUJI .....	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR .....	iv
HALAMAN PERSEMBAHAN .....	v
HALAMAN MOTO .....	vi
KATA PENGANTAR .....	vii
SARI .....	viii
GLOSARIUM .....	ix
DAFTAR ISI .....	x
DAFTAR TABEL .....	xii
DAFTAR GAMBAR .....	xiii
BAB I PENDAHULUAN .....	
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Tujuan Penelitian .....	2
1.4 Manfaat Penelitian .....	2
1.5 Batasan Masalah .....	2
1.6 Langkah Penyelesaian .....	3
1.7 Sistematika Penulisan .....	3
BAB II LANDASAN TEORI .....	
2.1 Kerangka Berpikir .....	5
2.2 Tinjauan Pustaka .....	6
2.3 Dasar Teori .....	9
2.3.1 Penjadwalan .....	9
2.3.2 Internet of Things ( <i>IoT</i> ) .....	10
2.3.3 Rapsberry PI 3 .....	11
2.3.4 Google Application API Calendar .....	13
BAB III METODE PENELITIAN .....	
3.1 Pengumpulan Data .....	14
3.2 Metodologi Penelitian .....	14
3.3 Analisis Kebutuhan .....	15
3.3.1 Analisis Kebutuhan Input .....	15
3.3.2 Analisis Kebutuhan Proses .....	16
3.3.3 Analisis Kebutuhan Output .....	16
3.3.4 Analisis Kebutuhan Perangkat Lunak .....	16
3.3.5 Analisis Kebutuhan Perangkat Keras .....	16
3.4 Menentukan Desain .....	17
3.5 Metode Perancangan .....	17
3.5.1 Perancangan Perangkat Keras .....	17
3.5.2 Perancangan Perangkat Lunak .....	19
3.5.3 Perancangan Antarmuka Pengguna .....	20
3.5.4 Use Case Diagram .....	21
3.5.5 Activity Diagram .....	24
BAB IV IMPLEMENTASI DAN PEMBAHASAN .....	
4.1 Implementasi .....	29

4.1.1	Perangkat Keras yang Digunakan .....	29
4.1.2	Perangkat Lunak yang Digunakan .....	29
4.1.3	Implementasi Perancangan Perangkat Keras .....	30
4.1.4	Implementasi Perancangan Perangkat Lunak.....	33
4.1.5	Implementasi Perancangan Antarmuka Pengguna .....	40
4.2	Pengujian Penelitian.....	44
4.2.1	Black Box Testing .....	45
4.2.2	Pengujian UAT ( <i>User Acceptance Test</i> ) .....	45
4.3	Analisis penelitian.....	47
4.3.1	Kelebihan Sistem.....	47
4.3.2	Kekurangan Sistem.....	47
	BAB V KESIMPULAN DAN SARAN.....	49
5.1	Kesimpulan .....	49
5.2	Saran.....	49
	DAFTAR PUSTAKA .....	50
	LAMPIRAN .....	51



## DAFTAR TABEL

Tabel 2.1 Perbandingan Penelitian .....	7
Tabel 3.1 Tabel Hak Akses .....	22
Tabel 4.1 Perintah instalasi <i>Magic Mirror</i> .....	34
Tabel 4.2 Perintah menjalankan <i>Magic Mirror</i> .....	34
Tabel 4.3 Kode program Open Weather Map API .....	35
Tabel 4.4 Perintah mengunduh googleapis .....	37
Tabel 4.5 Kode program <i>credentials</i> .....	37
Tabel 4.6 Kode program <i>index.js</i> .....	37
Tabel 4.7 Perintah autentikasi .....	38
Tabel 4.8 Kode program <i>parsing</i> objek .....	39
Tabel 4.9 Perintah mengunduh modul MMM-MyCalendar .....	41
Tabel 4.10 Kode program MMM-MyCalendar .....	41
Tabel 4.11 Kode program <i>upcoming event</i> .....	42
Tabel 4.12 Kode program <i>current events</i> .....	43
Tabel 4.13 Hasil <i>Black box Test</i> .....	45
Tabel 4.14 Tabel Nilai .....	46
Tabel 4.15 Hasil Kuesioner .....	46

## DAFTAR GAMBAR

Gambar 2. 1 kerangka berpikir .....	5
Gambar 2.2 Ilustrasi dari <i>Internet Of Things</i> .....	11
Gambar 2.3 <i>Raspberry Pi 3</i> .....	12
Gambar 3.1 Skema Perangkat yang terhubung pada <i>Raspberry Pi</i> .....	18
Gambar 3.2 Blok Diagram perangkat lunak pada sistem .....	19
Gambar 3.3 Rancangan Halaman Tampil Data <i>Event</i> .....	20
Gambar 3.4 <i>Use Case Diagram</i> Pemesanan Ruangan .....	22
Gambar 3.5 <i>Use Case Diagram</i> Sistem <i>Magic Mirror</i> .....	23
Gambar 4.1 Proses membuat <i>bootable</i> disk .....	31
Gambar 4.2 Pemasangan <i>Raspberry Pi</i> dengan power adaptor .....	32
Gambar 4.3 Konfigurasi aktivasi VNC pada <i>Raspberry Pi</i> .....	32
Gambar 4.4 Koneksi <i>remote</i> menggunakan Vinagre .....	33
Gambar 4.5 Tampilan <i>Magic Mirror</i> .....	34
Gambar 4.6 <i>Generate</i> Open Weather Map API .....	35
Gambar 4.7 Tampilan cuaca pada sistem .....	36
Gambar 4.8 Kunci keamanan untuk akses Google Calendar API .....	36
Gambar 4.9 Mendapatkan <i>file</i> token .....	39
Gambar 4.10 <i>parsing</i> data .....	40
Gambar 4.11 Menambahkan Modul MMM-MyCalendar .....	42
Gambar 4.12 Antarmuka pengguna Tabel <i>upcoming event</i> .....	43
Gambar 4.13 Antarmuka pengguna Tabel <i>current events</i> .....	44

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Perkembangan teknologi informasi pada saat ini terus berkembang seiring dengan kebutuhan manusia yang menginginkan kemudahan, kecepatan dan keakuratan dalam memperoleh informasi. Perkembangan ini membawa pengaruh yang besar bagi kelangsungan hidup sehari-hari masyarakat sehingga menyebabkan meningkatnya kebutuhan akan informasi dan tentunya membutuhkan teknologi untuk memperoleh informasi tersebut. Contoh sumber informasi yang banyak digunakan oleh masyarakat saat ini adalah *Internet*. Salah satu konsep dari penerapan ke perangkat lain dengan memanfaatkan jaringan sehingga menghasilkan keluaran yang berupa informasi dikenal sebagai konsep *Internet of Things (IoT)*.

*Internet of Things (IoT)* adalah struktur di mana objek, orang disediakan dengan identitas eksklusif dan kemampuan untuk pindah data melalui jaringan tanpa memerlukan dua arah antara manusia ke manusia yaitu sumber ke tujuan atau interaksi manusia ke computer (Burange & Misalkar, 2015). *IoT* memiliki konsep di mana suatu objek memiliki kemampuan untuk mentransfer data melalui jaringan untuk menghubungkan suatu perangkat ke perangkat lain. Contoh masalah yang dapat diselesaikan dengan *IoT* adalah dengan menggunakan *Raspberry Pi*. Di kehidupan sehari-hari, cermin digunakan oleh manusia untuk dapat mematu diri dan mengevaluasi penampilan. Sama halnya dengan *Magic Mirror*, cermin tersebut akan dipadukan pada perangkat pintar dengan mengimplementasikan konsep *Internet of Things (IoT)*. *Magic Mirror* merupakan sebuah sistem yang dipadukan pada cermin serta dapat menampilkan data berupa informasi cuaca, penjadwalan, dan waktu yang dapat dilihat oleh pengguna saat sedang bercermin.

Masalah tingginya tuntutan akan kebutuhan informasi bagi *user* saat ini perlu dipecahkan. Informasi tersebut perlu ditampilkan secara ringkas dan menyesuaikan mobilitasnya dengan pengguna. Dalam upaya untuk memenuhi kebutuhan akan informasi tersebut, maka diperlukan sebuah alat yang dapat mengatasi masalah ini. *Magic Mirror* merupakan sebuah sistem yang dapat memberikan layanan informasi secara efisien.

Tujuan dari *Magic Mirror* adalah untuk menyediakan akses bagi seorang pengguna dalam menerima informasi waktu, cuaca, dan lain-lain secara *up to date*. Kelebihan lain dari sistem *Magic Mirror* ini adalah menyediakan fitur notifikasi pada calendar menggunakan Google

Calendar API. Ini akan sangat membantu pengguna untuk melihat informasi pada tanggal tertentu. Contohnya *meeting room*, pengguna akan mendapatkan informasi berupa *owner meeting*, tanggal *meeting*, jam *meeting* dan informasi bermanfaat lainnya.

Oleh karena itu, berdasarkan penjelasan paragraf diatas fokus pada penulisan tugas akhir ini adalah menerapkan *Magic Mirror* menggunakan konsep *Internet of Things (IoT)* menggunakan *Raspberry Pi*. Dengan begitu penerapan ini diharapkan dapat memberikan manfaat bagi pengguna.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang diuraikan di atas, maka peneliti merumuskan masalah sebagai berikut:

1. Bagaimana mengembangkan sebuah sistem yang dapat mengatur penjadwalan menggunakan *Raspberry Pi*?
2. Bagaimana menentukan lokasi peletakan sistem penjadwalan sesuai kebutuhan pengguna?
3. Siapa saja pengguna yang dapat menggunakan sistem penjadwalan?

## 1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini diantaranya:

1. Sistem dapat diintegrasikan dengan Google Calendar API yang bertujuan untuk mengatur jadwal *meeting*.
2. Sistem dapat menampilkan informasi penjadwalan secara *real time*.

## 1.4 Manfaat Penelitian

Manfaat dari penelitian ini adalah mengembangkan sistem yang dapat memberikan informasi mengenai penjadwalan yang meliputi nama tamu, nama ruangan, status ruangan, waktu mulai, dan waktu selesai. Sehingga dapat memudahkan pengguna dalam mengatur penjadwalan agar lebih efisien.

## 1.5 Batasan Masalah

Sistem ini mempunyai batasan masalah sebagai berikut:

1. Perangkat yang digunakan pada penelitian ini menggunakan *Raspberry Pi 3 Model B*.
2. Sistem hanya dapat diintegrasikan Google Calendar API.

3. Sistem hanya menampilkan informasi yang meliputi agenda *meeting*, waktu, dan cuaca.
4. Pengujian yang dilakukan pada sistem hanya menggunakan pengujian *black box* dan pengujian *user acceptance test*.
5. Sistem Penjadwalan *Magic Mirror* menggunakan *Raspberry Pi* hanya digunakan di lingkungan kampus terpadu Universitas Islam Indonesia.

## 1.6 Langkah Penyelesaian

Langkah penyelesaian untuk pengembangan aplikasi *Magic Mirror* akan dilakukan dengan beberapa tahapan berikut:

- a. Analisis masalah  
Pada tahap analisis masalah ini, akan dilakukan analisis terhadap proses kebutuhan bagi para pengguna *Magic Mirror* untuk kemudian digunakan sebagai bahan perancangan sistem.
- b. Perancangan Desain Sistem  
Hasil dari tahapan analisis kebutuhan dijadikan masukan untuk mendesain sistem yang akan dibuat sehingga diperoleh desain untuk alur kerja sistem *Magic Mirror*.
- c. Implementasi Desain  
Pada tahap ini mulai dilakukan pembuatan sistem, sistem *Magic Mirror* akan dikembangkan berdasarkan hasil dari tahap perancangan desain sistem.
- d. Pengujian Sistem  
Tahap pengujian ini bertujuan untuk mengetahui tingkat kesesuaian sistem yang dibangun dengan kebutuhan pengguna aplikasi.

## 1.7 Sistematika Penulisan

Untuk mempermudah dan memahami lebih jelas terkait materi yang akan dibahas, penulis memberikan gambaran umum tentang sistematika penulisan yang dibuat menjadi beberapa bagian bab, dapat dilihat lebih jelas seperti di bawah ini:

### **BAB I: PENDAHULUAN**

Bab ini merupakan bagian awal bab yang membahas tentang latar belakang masalah yang mengenai sistem penjadwalan *Magic Mirror* menggunakan *Raspberry Pi* kemudian dilanjutkan dengan menguraikan rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, penyelesaian masalah, langkah penyelesaian hingga sistematika penulisan.



## **BAB II: LANDASAN TEORI**

Bab ini berisikan penjelasan dasar teori yang berhubungan dengan Sistem *Magic Mirror* Menggunakan *Raspberry Pi* dan berfungsi sebagai sumber informasi dalam memahami permasalahan yang berkaitan dengan konsep sistem *Magic Mirror* menggunakan *Raspberry Pi* serta penjelasan lebih lanjut mengenai tugas akhir ini.

## **BAB III: METODOLOGI**

Bab ini memberikan penjelasan mengenai metode apa yang dibutuhkan atau digunakan dalam pembuatan sistem *Magic Mirror* menggunakan *Raspberry Pi* dengan Metode *Waterfall*, menjelaskan metode pengumpulan data, menganalisis kebutuhan dan penjelasan terhadap perancangan sistem.

## **BAB IV: IMPLEMENTASI DAN PENGUJIAN**

Bab ini memuat tentang pembahasan yang berisi implementasi dari perancangan antarmuka yang telah dibuat serta mendokumentasikan sistem yang telah dibuat termasuk didalamnya penjelasan hasil kinerja sistem, pengujian dan evaluasi terhadap kesesuaian sistem dengan kebutuhan dan tujuan yang diharapkan.

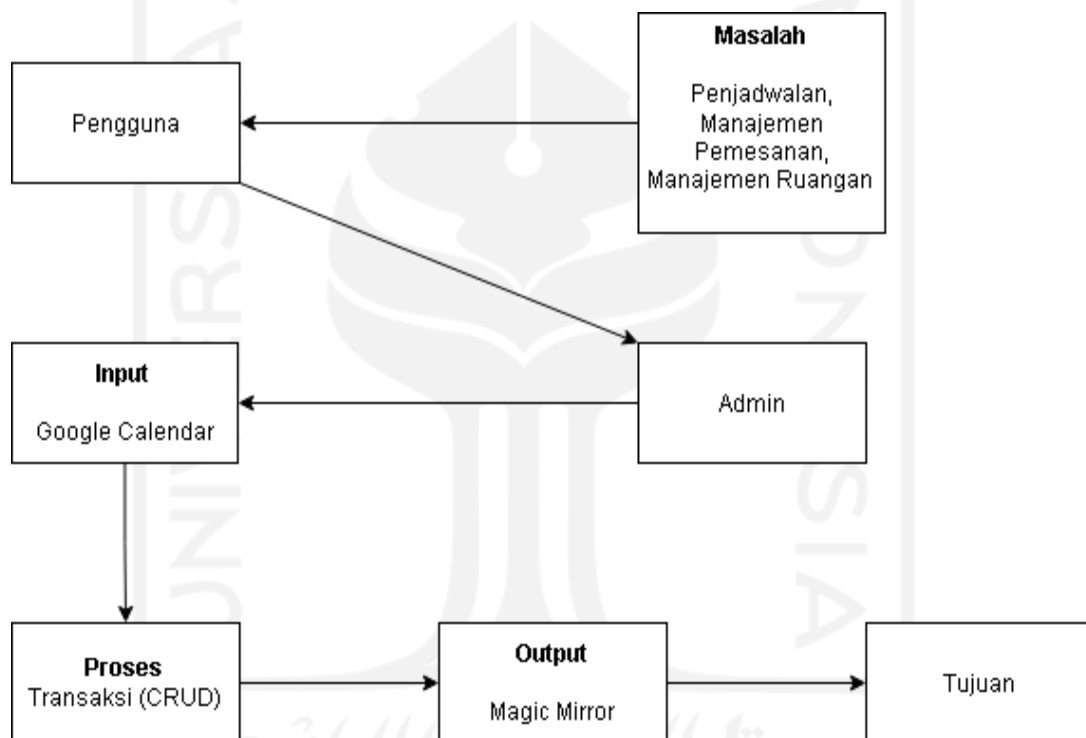
## **BAB V: KESIMPULAN DAN SARAN**

Bab ini membahas tentang kesimpulan yang merupakan ulasan keseluruhan mengenai hasil yang telah didapatkan pada bab sebelumnya berdasarkan temuan yang ada. Dan bab ini juga berisikan saran bagi pengembang sistem berdasarkan hasil pengujian yang telah dilakukan.

## BAB II LANDASAN TEORI

### 2.1 Kerangka Berpikir

Kerangka berpikir pada penulisan ini sebagai arahan dalam pelaksanaan penulisan, terutama untuk memahami alur pemikiran, sehingga analisis yang dilakukan lebih sistematis dan sesuai dengan tujuan penulisan. Kerangka berpikir dalam penulisan ini digambarkan dalam skema berikut:



Gambar 2. 1 kerangka berpikir

Berdasarkan kerangka berpikir diatas, dapat dijelaskan bahwa masalah yang dimiliki oleh pengguna diantaranya penjadwalan, manajemen pemesanan, dan manajemen ruangan. Untuk menyelesaikan masalah tersebut pengguna harus menghubungi *admin*, sehingga *admin* dapat melakukan *input* pada Google Calendar yang kemudian dilakukan proses *CRUD*. Proses *CRUD* menghasilkan *output* berupa informasi yang ditampilkan pada perangkat *Magic Mirror*, sehingga dapat mencapai tujuan.

## 2.2 Tinjauan Pustaka

- a. Penelitian yang dilakukan oleh Nugraha (2018) dengan judul *Pembangunan Smart Mirror menggunakan Raspberry Pi Berbasis IoT*. Bandung Lautan Photographer memiliki tiga Bagian utama yaitu *Photographer*, Model, dan *Makeup Artist*. Rata-rata seorang yang berada di bagian *makeup* artis memiliki lebih dari satu pekerjaan. Tujuan dari penelitian ini adalah membantu bagian *makeup artist* di Bandung Lautan *Photographer* untuk mengingatkan jadwal, data client serta catatan yang telah mereka buat agar dapat dilihat saat dalam keadaan sibuk atau saat sedang merias dan bercermin. *Smart Mirror* merupakan sebuah cermin yang dapat menampilkan data berupa data jadwal, data *client* dan catatan yang dapat dilihat oleh seorang *makeup artist* saat sedang bercermin. Sebelum *smart mirror* menampilkan data pengguna diharuskan melakukan *login* dengan menggunakan sidik jari. Selain itu cermin akan menampilkan notifikasi yang akan mengingatkan jadwal atau data *client* baik saat pengguna sudah *login* ataupun belum melakukan *login*. Data sendiri dapat ditambahkan oleh seorang *makeup artist* melalui aplikasi pada perangkat *mobile*. *Form* data pada aplikasi *mobile* sudah disesuaikan dengan kebutuhan seorang *makeup artist* berdasarkan hasil wawancara dengan koordinator bagian *makeup* artis di Bandung Lautan Photographer. Berdasarkan hasil pengujian dapat disimpulkan bahwa perangkat *smart mirror* sudah cukup memudahkan seorang *makeup* artis khususnya dalam mengingat jadwal dan data *client*. Sistem ini diharapkan dapat terus dikembangkan dan dapat membantu tidak hanya pada bagian *makeup* artis saja.
- b. Penelitian yang dilakukan oleh Assagaf dkk (2018) menyatakan bahwa penjadwalan praktikum merupakan proses penyusunan jadwal pelaksanaan yang menginformasikan sejumlah mata kuliah, dosen yang mengajar, ruang, serta waktu kegiatan perkuliahan di laboratorium. Perlu diperhatikan beberapa aspek untuk menyusun jadwal perkuliahan yang sesuai dengan kebutuhan. Aspek yang perlu diperhatikan antara lain adalah aspek dari dosen yang mengajar, mata kuliah yang diajar. Penyusunan jadwal secara manual cenderung membutuhkan waktu yang lebih lama dan ketelitian yang cukup bagi pembuat jadwal. Untuk dapat membuat jadwal yang *optional*, dibutuhkan metode *optimasi*. Pada penelitian ini, akan diuji coba metode *optimasi* dalam pembuatan jadwal praktikum yaitu Algoritma Genetika. Algoritma genetika merupakan pendekatan komputasional untuk menyelesaikan masalah yang dimodelkan dengan proses biologi dari evolusi. Parameter

parameter Algoritma Genetika yang mempengaruhi jadwal perkuliahan yang dihasilkan adalah jumlah individu, probabilitas *crossover*, probabilitas mutasi serta metode seleksi, *crossover* yang digunakan. Pengujian dilakukan dengan cara mencari nilai parameter algoritma genetika yang paling optimal dalam jadwal perkuliahan. Hasil penelitian menunjukkan bahwa dengan jumlah generasi, jumlah individu, probabilitas *crossover* dan probabilitas mutasi dapat menghasilkan jadwal yang paling optimal.

- c. Penelitian lainnya oleh Ramadhan (2019) mengatakan bahwa *meeting* menjadi bagian penting yang harus dilakukan oleh suatu organisasi karena dapat meningkatkan produktivitas, *performance* dan moral organisasi tersebut. Departemen Ilmu Komputer (Ilkom) sebagai salah satu departemen di Fakultas Matematika dan Ilmu Pengetahuan Alam (FMIPA) IPB memiliki kegiatan *meeting* yang dihadiri oleh banyak dosen dan staf pada ruangan tertentu. Tetapi ruangan untuk *meeting* tidak selalu tersedia sehingga menyulitkan para dosen dan staf dalam mencari ruangan. Hal ini membuat proses *meeting* menjadi kurang efisien. Oleh karena itu, untuk memudahkan pencarian dan pemesanan ruangan *meeting*, pada penelitian ini dikembangkan sebuah sistem informasi *meeting* dengan pendekatan *prototyping* model yang diimplementasikan dalam framework PHP, yaitu CodeIgniter. Sistem yang dikembangkan sudah bisa melakukan pemesanan ruangan, membatalkan pesanan, mengubah pesanan, dan pengelola ruangan dapat melakukan konfirmasi. Pengujian *black box* telah dilakukan menggunakan tujuh skenario dan hasilnya sesuai dengan yang diharapkan. Pada iterasi kedua ditambahkan mengeksport daftar reservasi menjadi excel dan PDF dan *dashboard* yang berfungsi untuk melihat persentase pemakaian ruangan dan frekuensi *booking*.

Tabel 2.1 Perbandingan Penelitian

NO	Judul	Penulis	Pencapaian	Fitur
1	Pembangunan <i>smart mirror</i> menggunakan <i>Raspberry Pi</i> Berbasis <i>IoT</i>	Nugraha	Sistem dapat <i>login</i> menggunakan <i>fingerprint</i> , serta <i>admin</i> bisa menambahkan data dengan menggunakan aplikasi mobile	<ul style="list-style-type: none"> <li>• Fitur <i>Login</i> menggunakan <i>fingerprint</i>.</li> <li>• Fitur menambahkan data <i>client</i>, edit, dan hapus.</li> </ul>

			dan menampilkan informasi kebutuhan pengguna pada cermin.	<ul style="list-style-type: none"> <li>• Fitur <i>authentication</i> agar dapat menampilkan data <i>client</i> pada cermin.</li> </ul>
2	Membangun Sistem Informasi Penjadwalan Dengan Metode Algoritma Genetika Pada Laboratorium Teknik Informatika Universitas Muhammadiyah Maluku Utara	Agil Assagaf, Adelina Ibrahim, Catur Suranto	Sistem menggunakan metode seleksi dan <i>crossover</i> sehingga dapat menghasilkan penjadwalan yang lebih optimal.	<ul style="list-style-type: none"> <li>• Fitur <i>login</i> berdasarkan peran yaitu <i>admin</i> dan dosen.</li> <li>• Fitur peran dosen meliputi <i>mastering profile</i>, dan menampilkan hasil jadwal perkuliahan.</li> <li>• Fitur peran <i>admin</i> <i>mastering data waktu</i>, kelas, dosen, generate penjadwalan, dan perhitungan algoritma genetika.</li> </ul>
3	Pengembangan Sistem Informasi <i>Meeting</i> Berbasis <i>Web</i> Menggunakan	Ramadhan, Dean Apriana	Sistem dibangun menggunakan metode <i>prototyping</i> dan informasi <i>meeting</i>	<ul style="list-style-type: none"> <li>• Fitur <i>login admin</i>.</li> <li>• Fitur pencarian ruangan.</li> </ul>

	<i>Framework</i> CodeIgniter		pada sistem dapat diunduh dengan keluaran format pdf atau excel.	<ul style="list-style-type: none"> <li>• Fitur pemesanan ruangan yang tersedia.</li> <li>• Fitur cetak data reservasi ruangan berupa excel dan pdf.</li> </ul>
--	---------------------------------	--	------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------

Berdasarkan hasil *review* penelitian diatas terdapat perbedaan dengan penelitian yang akan dilakukan. Pada penelitian yang akan dilakukan, sistem *Magic Mirror* ini dapat diintegrasikan dengan Google Calendar API yang bertujuan untuk mengatur jadwal *meeting* serta informasi tersebut nantinya akan ditampilkan pada halaman *Magic Mirror*.

## 2.3 Dasar Teori

Dasar Teori pada Sistem Penjadwalan *Magic Mirror* menggunakan *Raspberry Pi* menjadi dasar dalam penelitian yang akan dilakukan. Dasar teori yang digunakan dalam penjelasan di bawah sebagai berikut:

### 2.3.1 Penjadwalan

Penjadwalan terkait pada aktivitas dalam hal untuk membuat sebuah jadwal. Sebuah jadwal adalah sebuah tabel dari kegiatan-kegiatan yang disusun berdasarkan waktu kapan aktivitas tersebut dilaksanakan. Kegiatan ini biasanya pertemuan antar beberapa komponen pada sebuah waktu dan tempat yang sama. Jadwal harus memenuhi beberapa persyaratan dan memenuhi keinginan semua orang yang terlibat sebaik mungkin. Waktu dari aktivitas harus disusun sedemikian rupa sehingga tidak ada salah satu komponen pun memiliki lebih dari satu kegiatan pada waktu yang sama (Pinedo, 2012).

Dalam masalah penjadwalan memiliki beberapa macam batasan yang dapat menyebabkan *output* yang dihasilkan menjadi salah. Dalam menerapkan batasan dalam suatu masalah, biasanya tidak terlalu sama untuk setiap masalah. Batasan tersebut terdiri dari :

- Edge constraint*

*Edge constraint* adalah batasan yang mengatur dua kejadian tidak boleh menempati satu slot pada waktu yang sama. Contohnya pada hari Senin jam 07.30 sampai 08.10 tidak mungkin dosen A mengajar di ruang kelas 1-A dan mengajar di ruang kelas 5-A.

b. *Ordering constraint*

*Ordering constraint* adalah batasan yang menjaga urutan kejadian dalam *time table*. Contoh mata pelajaran A harus dilakukan sebelum mata pelajaran B. Biasanya masalah batasan ini jarang digunakan karena dapat menyebabkan penjadwalan menjadi lebih rumit.

c. *Capacity constraint*

*Capacity constraint* adalah batasan yang berhubungan dengan kapasitas ruangan. Untuk masing-masing ruangan hanya boleh diisi sebanyak 40 orang.

Pada penelitian tugas akhir ini penjadwalan dan informasi yang ditampilkan pada sistem akan dikembangkan menggunakan konsep *Internet of Things*.

### 2.3.2 Internet of Things (IoT)

*Internet of Things (IoT)* adalah struktur di mana objek, orang disediakan dengan identitas eksklusif dan kemampuan untuk pindah data melalui jaringan tanpa memerlukan dua arah antara manusia ke manusia yaitu sumber ke tujuan atau interaksi manusia ke komputer. *IoT* memiliki konsep di mana suatu objek memiliki kemampuan untuk mentransfer data melalui jaringan untuk menghubungkan suatu perangkat ke perangkat lain (Burange & Misalkar, 2015).

Adapun kemampuan seperti berbagi data, *remote control*, dan sebagainya, termasuk juga pada benda di dunia nyata. Contohnya bahan pangan, elektronik, koleksi, peralatan apa saja, termasuk benda hidup yang semuanya tersambung ke jaringan lokal dan global melalui sensor yang tertanam dan selalu aktif. Pada dasarnya, *Internet of Things* mengacu pada benda yang dapat diidentifikasi secara unik sebagai representasi virtual dalam struktur berbasis Internet. Istilah *Internet of Things* awalnya disarankan oleh Kevin Ashton pada tahun 1999 dan mulai terkenal melalui *Auto-ID Center* di MIT. Dalam implementasi *Internet of Things (IoT)* sangat erat kaitannya dengan sebuah perangkat seperti *Raspberry Pi*.



Gambar 2.2 Ilustrasi dari *Internet Of Things*

### 2.3.3 Rapsberry PI 3

*Raspberry Pi* adalah sebuah komputer papan tunggal (*single board computer*) atau SBC berukuran kartu kredit. *Raspberry Pi* telah dilengkapi dengan semua fungsi layaknya sebuah komputer lengkap, menggunakan SoC (*System-on-a-chip*) ARM yang dikemas dan diintegrasikan diatas PCB. Perangkat ini menggunakan kartu SD untuk *booting* dan penyimpanan jangka panjang.

Raspberry Pi memiliki dua model yaitu model A dan model B. Secara umum *Raspberry Pi* Model B, 512 MB RAM. Perbedaan model A dan B terletak pada *memory* yang digunakan, Model A menggunakan *memory* 256 MB dan model B 512 MB. Selain itu model B juga sudah dilengkapi dengan *ethernet port* (kartu jaringan) yang tidak terdapat di model A. Desain *Raspberry Pi* didasarkan seputar SoC (*System on a chip*) Broadcom BCM2835, yang telah menanamkan prosesor ARM1176JZF-S dengan 700 MHz, VideoCore IV GPU, dan 256 Megabyte RAM (model B). Penyimpanan data didesain tidak untuk menggunakan *hard disk* atau *solid-state drive*, melainkan mengandalkan kartu SD (*memory card*) untuk *booting* dan penyimpanan jangka panjang. *Hardware Raspberry Pi* tidak memiliki *real time clock*, sehingga OS harus memanfaatkan *timer* jaringan server sebagai pengganti. Namun komputer yang mudah dikembangkan ini dapat ditambahkan dengan fungsi *real time* (seperti DS1307) dan banyak lainnya, melalui saluran GPIO (*General purpose input/output*) via antarmuka IC (*Inter Integrated Circuit*). *Raspberry Pi* yang bersifat *open source* bisa dimodifikasi sesuai dengan kebutuhan penggunaannya. Sistem operasi utama *Raspberry Pi* menggunakan Debian



GNU/Linux dan mendukung Bahasa pemrograman apa pun. Salah satu pengembang OS untuk Raspberry Pi telah meluncurkan sistem operasi yang dinamai Raspbian, Raspbian di klaim mampu memaksimalkan perangkat *Raspberry Pi*. Sistem operasi tersebut dibuat berbasis Debian yang merupakan salah satu distribusi Linux OS (Yuwono, Nugroho, & Heriyanto, 2015).

*Raspberry Pi 3* merupakan generasi ketiga dari keluarga *Raspberry Pi*. *Raspberry Pi 3* memiliki RAM 1GB dan grafis Broadcom Video Core IV pada frekuensi *clock* yang lebih tinggi dari sebelumnya yang berjalan pada 250MHz. *Raspberry Pi 3* menggantikan *Raspberry Pi 2* model B pada bulan Februari 2016. Kelebihannya dibandingkan dengan *Raspberry Pi 2* adalah:

- a. A 1.2GHz 64-bit quad-core ARMv8 CPU
- b. 802.11n Wireless LAN
- c. Bluetooth 4.1
- d. Bluetooth Low Energy (BLE)

Sama seperti Pi 2, *Raspberry Pi 3* juga memiliki 4 USB port, 40 pin GPIO, *Full HDMI port*, *Port Ethernet*, *Combined 3.5mm audio jack and composite video*, *Camera interface (CSI)*, *Display interface (DSI)*, slot kartu Micro SD (Sistem tekan dan tarik, berbeda dari yang sebelumnya ditekan-tekan), dan VideoCore IV 3D graphics core. *Raspberry Pi* merupakan perangkat yang sangat *compatible* apabila dipadukan dengan sebuah teknologi lainnya seperti Google Calendar API.



Gambar 2.3 Rapsberry Pi 3

Sumber: (Ramli, Mamahit, & Wuwung, 2018)

### 2.3.4 Google Application API Calendar

Google Application API memungkinkan pengembang aplikasi untuk menulis aplikasi yang mengakses aplikasi Google seperti Gmail, *Calendar* dan lainnya menggunakan Google Data API, Gadget, dan Google Apps Script. Google menyediakan protokol sederhana Google Data yang dapat digunakan para pemrogram untuk membuat aplikasi yang membaca dan menulis data diinternet yang disediakan oleh Google. Beberapa layanan Google menyediakan akses data eksternal melalui API yang memanfaatkan protokol ini (Suyadi, 2016).

Google Calendar API digunakan untuk mengembangkan aplikasi yang dapat membuat kegiatan baru, menelusuri, mengubah dan menghapus kegiatan di Google Calendar menggunakan berbagai bahasa pemrograman. API tersedia dalam bentuk library yang dapat diunduh pada situs Google Developer. Query menggunakan Google Calendar API dibatasi 1.000.000 kali, untuk menambah jumlah query pengguna harus membayar dengan mengaktifkan billing terlebih dahulu untuk proyek yang bersangkutan di alamat <https://developers.google.com/console/help/console/> .

Berikut konsep dasar Google Calendar:

- a. Event  
Event (kegiatan) tunggal dalam kalender memuat informasi seperti judul kegiatan, awal dan akhir kegiatan, dan peserta.
- b. Calendar  
Sekumpulan metadata, seperti deskripsi untuk kalender tunggal.
- c. Calendar List  
Daftar semua kalender pengguna dalam antarmuka Calendar.
- d. Setting  
Preferensi pengguna dari antarmuka Calendar, seperti zona waktu.
- e. ACL  
Sebuah aturan kontrol akses yang berisi informasi seperti jenis dan ruang lingkup aturan.
- f. Free/busy  
Satu set waktu, untuk satu set kalender, di mana kalender tidak memiliki acara.

Berdasarkan dasar teori diatas, tugas akhir ini akan mengembangkan sistem penjadwalan *Magic Mirror* menggunakan *Raspberry Pi* dengan mengadopsi konsep *Internet of Things (IoT)*, serta memanfaatkan Google Calendar dalam penerapannya. Sehingga didapatkan sebuah informasi penjadwalan.

## BAB III METODE PENELITIAN

### 3.1 Pengumpulan Data

Pada tahap pengumpulan data dilakukan untuk mendapatkan suatu informasi yang berhubungan dengan masalah yang akan diteliti hingga mencapai tujuan penelitian. Adapun pengumpulan data yang penulis gunakan untuk mencapai tujuan penelitian adalah sebagai berikut:

a. Observasi

Proses pengumpulan data dengan cara mengadakan pengamatan langsung pada lingkungan kampus Universitas Islam Indonesia untuk memperoleh informasi tambahan. Dalam hal ini ditemukan ruangan kelas yang tidak digunakan ketika tidak ada jam mata kuliah, dan dapat ditemukan juga mahasiswa masih melakukan *meeting* diluar ruangan, sehingga menyebabkan *meeting* tersebut menjadi kurang efisien.

Dari hasil observasi yang didapat bahwa lingkungan kampus Universitas Islam Indonesia belum mempunyai sistem yang dapat mengelola setiap ruangan kelas untuk memberikan kemudahan bagi mahasiswa, staf, dan dosen dalam melakukan *meeting*.

b. Kepustakaan

Teknik pengumpulan data yang dikumpulkan bersumber dari literatur (buku-buku yang mendukung) untuk mendapatkan konsep teori mengenai masalah yang diteliti seperti dengan mencari data dari internet maupun penelitian sebelumnya. Data yang menjadi landasan perbandingan ini diperoleh melalui *browsing* diinternet dengan menggunakan mesin pencari Google dan membaca berbagai jenis literatur dan hasil penelitian terdahulu.

### 3.2 Metodologi Penelitian

Metode yang akan digunakan pada penelitian ini adalah metode *waterfall*. Pada teknik *waterfall*, sistem berkembang secara sistematis dari satu tahap ke tahap yang lainnya. Secara umum, pada penelitian ini terdiri dari 5 tahapan yaitu:

a. Tahap identifikasi dan analisis. Tahap ini terdiri dari identifikasi dan analisis:

1. Proses - proses evaluasi hasil analisis kebutuhan aplikasi *Magic Mirror*.
2. Keinginan dan kebutuhan pengguna terhadap sistem yang akan dikembangkan.

- b. Tahap perancangan aplikasi dengan memperhatikan prinsip-prinsip desain yang akan diterapkan dalam keseluruhan proses desain.
- c. Tahapan implementasi.
- d. Tahapan uji coba fungsionalitas sistem dan kesesuaiannya dengan kebutuhan pengguna sistem.
- e. Pemeliharaan (*operation & maintenance*) sistem yang telah digunakan oleh pengguna akan mengalami kesalahan atau membutuhkan perkembangan fungsional menyesuaikan dengan lingkungan.

### 3.3 Analisis Kebutuhan

Analisis kebutuhan merupakan suatu proses dalam mencari informasi akan kebutuhan sistem yang dibangun. Informasi yang dikumpulkan berupa spesifikasi dari perangkat keras yang akan digunakan, kebutuhan perangkat lunak yang digunakan dan bentuk sistem pada saat berjalan. Informasi ini berguna untuk menunjang pengembangan pada sistem agar sesuai dengan apa yang diinginkan. Pada tahap ini penulis mengumpulkan informasi tentang sistem yang akan dirancang pada tugas akhir ini yaitu sistem penjadwalan *Magic Mirror* menggunakan *Raspberry Pi*. Informasi yang dikumpulkan mengenai sistem yang akan dibuat adalah kebutuhan perangkat keras dan kebutuhan perangkat lunak yang sesuai dengan sistem yang akan digunakan dalam perancangan sistem. Pengumpulan informasi pada sistem ini dilakukan guna mendapatkan gambaran akan kebutuhan dari sistem itu sendiri agar dapat berjalan dan berfungsi sesuai yang diinginkan.

#### 3.3.1 Analisis Kebutuhan Input

Pada tahapan kebutuhan *input* dilakukan analisis terhadap kebutuhan *input* apa saja yang dibutuhkan dalam melakukan pemesanan ruangan menggunakan layanan Google Calendar. Dalam menerapkan kebutuhan *input* diperlukan batasan agar tidak menimbulkan masalah pada *output* yaitu *edge constraint*, *ordering constraint*, dan *capacity constraint*. *Input* yang dibutuhkan dalam pengembangan sistem ini adalah sebagai berikut:

- a. Dosen dapat melakukan pemesanan ruangan kepada *admin*.
- b. Staf dapat melakukan pemesanan ruangan kepada *admin*.
- c. Mahasiswa dapat melakukan pemesanan ruangan kepada *admin*.

### 3.3.2 Analisis Kebutuhan Proses

Pada tahapan kebutuhan proses ini dilakukan analisis terhadap kebutuhan proses apa saja yang dibutuhkan oleh *admin*. Proses yang dibutuhkan dalam sistem ini adalah sebagai berikut:

- a. Proses *login admin* pada layanan Google Calendar.
- b. Proses tata kelola data *event* oleh *admin* pada layanan Google Calendar.

### 3.3.3 Analisis Kebutuhan Output

Pada tahapan kebutuhan output ini dilakukan analisis kebutuhan *output* informasi apa saja yang akan diberikan kepada pengguna. *Output* informasi yang diberikan kepada pengguna yang terdiri dari dosen, staf, dan mahasiswa dalam sistem ini berupa informasi data pemesanan ruangan meliputi nama *event*, nama pemesan, nama ruangan, status ruangan, deskripsi, waktu mulai, dan waktu selesai pada sistem *Magic Mirror*.

### 3.3.4 Analisis Kebutuhan Perangkat Lunak

Kebutuhan yang diperuntukkan bagi perangkat lunak merupakan sekumpulan perangkat lunak yang dibutuhkan dalam tahapan pengembangan pada sistem agar dapat berjalan sesuai yang diinginkan. Analisis kebutuhan perangkat lunak pada sistem ini adalah sebagai berikut:

- a. Raspbian OS
- b. Terminal
- c. Vinagre
- d. Open Weather Map
- e. Google Calendar
- f. *Magic Mirror*
- g. Postman

### 3.3.5 Analisis Kebutuhan Perangkat Keras

Dalam tahapan pengembangan sistem digunakan beberapa perangkat keras sebagai komponen pendukung agar sistem dapat berjalan dengan sesuai keinginan. Perangkat keras yang digunakan pada sistem ini antara lain sebagai berikut:

- a. *Raspberry Pi 3 Model B*
- b. MicroSD card 16 GB
- c. Kabel USB
- d. Adaptor power 5V

- e. VGA konverter
- f. Monitor 18 inch
- g. Cermin
- h. Alumunium foil

### 3.4 Menentukan Desain

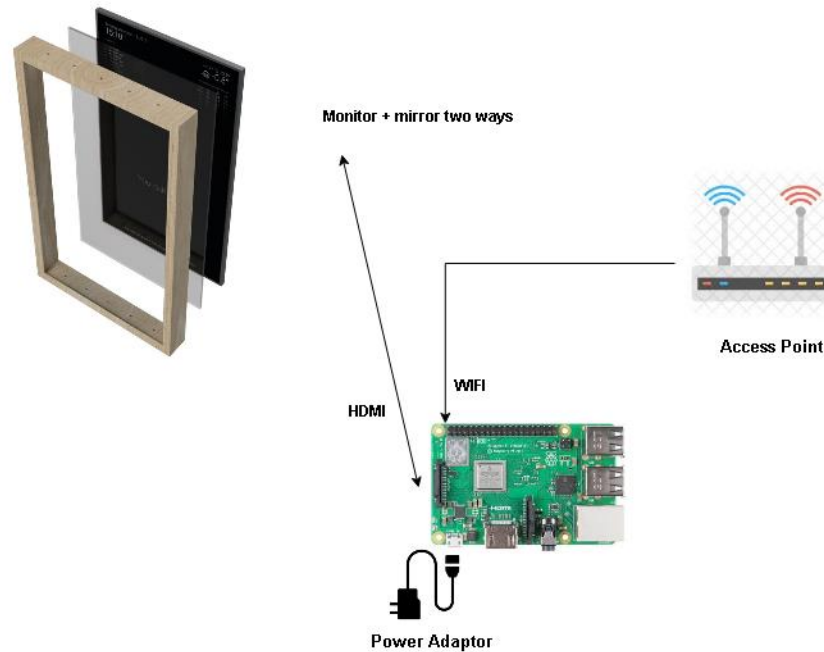
Menentukan desain dalam perancangan sistem yang dibangun bukan hanya terpaku dalam perangkat lunak, namun desain ini juga berlaku terhadap perangkat keras seperti halnya mengatur tata letak masing-masing komponen dari perangkat keras agar tidak mengganggu estetika pada sistem, dan untuk lokasi penempatan sistem *Magic Mirror* akan ditempatkan pada tiap-tiap ruangan kelas. Pada perangkat lunak juga didesain agar sangat mudah digunakan oleh pengguna baik yang sudah profesional maupun yang masih awam. Menentukan desain ini sama halnya menentukan bagaimana antarmuka tampilan, alur perangkat lunak, dan lain sebagainya.

### 3.5 Metode Perancangan

Pada tahap perancangan ini penulis membuat rencana mengenai aplikasi yang akan dibuat berdasarkan hasil analisis kebutuhan yang telah dilakukan. Dalam hal ini yaitu struktur perancangan dari perangkat keras, perancangan dari perangkat lunak, perancangan antarmuka pengguna, *use case diagram*, dan *activity diagram*. Tujuan utama dari perancangan ini adalah untuk memenuhi kebutuhan akan pengguna dan dalam memberikan gambaran yang jelas dari rancang bangun yang lengkap kepada penulis dan desain sistem yang terinci. Adapun penjelasan mengenai tahapan tersebut dapat dilihat pada penjelasan sebagai berikut:

#### 3.5.1 Perancangan Perangkat Keras

Dalam tahap merancang perangkat keras dilakukan perancangan terhadap perangkat keras yang digunakan dalam pengembangan sistem *Magic Mirror*. Perangkat keras yang digunakan dalam pengembangan sistem ini telah disebutkan pada bagian analisis kebutuhan perangkat keras. Relasi antar perangkat keras pada sistem ini dapat dilihat pada Gambar 3.1.



Gambar 3.1 Skema Perangkat yang terhubung pada *Raspberry Pi*

Pada Gambar 3.1 dapat dilihat hubungan antar perangkat keras yang sudah disebutkan pada analisis kebutuhan yang terhubung satu sama lain pada perancangan *Magic Mirror* menggunakan *Raspberry Pi*. Alur kerja antar perangkat keras yang terhubung pada sistem dapat dijelaskan sebagai berikut:

*Raspberry Pi* sebagai sistem *Magic Mirror* di mana semua proses dan aktivitas sistem dilakukan pada perangkat ini. Model *Raspberry Pi* yang digunakan adalah *Raspberry Pi 3 Model B* dengan CPU Quad Core 1.2 GHz Broadcom Bcm2837 64bit dan RAM 1GB, perangkat ini juga memiliki modul HDMI, Bluetooth dan WIFI yang sudah tertanam pada perangkat, sehingga dapat memungkinkan perangkat bisa terkoneksi pada jaringan internet melalui WIFI dari *access point*.

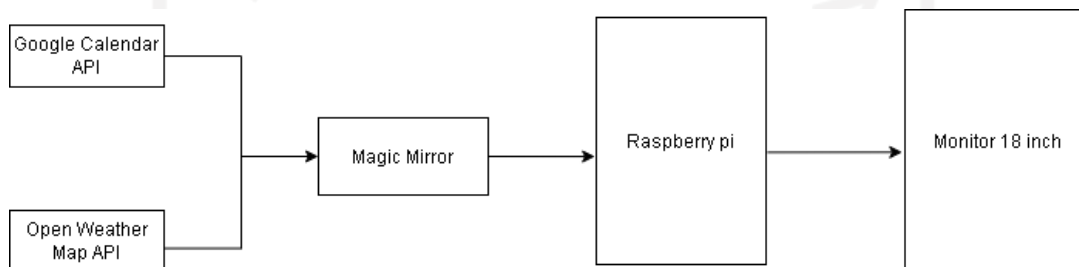
*Access point* di sini berfungsi sebagai penyedia internet bagi perangkat *Raspberry Pi*. Perangkat ini memberikan akses internet dari ISP (*Internet Service Provider*) ke *Raspberry Pi* melalui koneksi WIFI. Perangkat ini juga sudah memiliki DHCP (*Dynamic Host Configuration Protocol*) server yang bertugas memberi alamat internet protocol (IP) pada perangkat yang terhubung secara acak.

Monitor di sini berfungsi sebagai media informasi bagi perangkat *Raspberry Pi*, dan perangkat ini akan menampilkan sejumlah informasi mengenai cuaca, waktu, *event* hari ini dan *event* yang akan datang kepada pengguna. Perangkat ini akan dihubungkan dengan *Raspberry*

*Pi* menggunakan modul HDMI yang sudah ada, serta perangkat ini juga akan dipadukan pada cermin dua arah sehingga dalam penerapannya seperti selayaknya cermin.

### 3.5.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak didasarkan pada rancangan diagram blok pada Gambar 3.2. Tahap awal perancangan pada perangkat lunak yang perlu dilakukan adalah mendefinisikan *service* apa saja yang berjalan pada sistem, bagaimana setiap *service* berhubungan satu sama lain pada sistem dan di mana setiap *service* berjalan. Untuk uraian perancangan perangkat lunak pada sistem ini dapat dilihat pada Gambar 3.2



Gambar 3.2 Blok Diagram perangkat lunak pada sistem

Penjelasan pada Gambar 3.2 merupakan diagram blok dari perangkat lunak yang diimplementasikan pada sistem *Magic Mirror* dengan konfigurasi Google Calendar API dan Open Weather Map API menggunakan *Raspberry Pi*. Berikut ini merupakan penjelasan dari alur kerja pada *service* yang berjalan:

Google Calendar yang di konfigurasi dengan memanfaatkan *Application Programming Interface* (API) sebagai penghubung pada sistem *Magic Mirror* dalam penerapannya. Sehingga pada saat melakukan proses pemesanan ruangan oleh *admin* menggunakan layanan Google Calendar, semua data hasil dari pemesanan ruangan dapat diolah dan ditampilkan sebagai informasi pada sistem *Magic Mirror*. Pada sistem ini juga terdapat konfigurasi Open Weather Map di mana layanan ini merupakan layanan cuaca terkini, yang memanfaatkan API sebagai penghubung antara layanan cuaca dengan sistem *Magic Mirror*.

*Magic Mirror* merupakan sebuah perangkat lunak yang menggunakan bahasa pemrograman *javascript*, dan menggunakan *Electron framework* dalam penggunaannya sehingga tidak membutuhkan *web server* untuk menjalankan sistem. Pada tahapan pengembangan, *Magic Mirror* merupakan pusat dari semua konfigurasi untuk setiap layanan yang akan digunakan.



*Raspberry Pi* sebagai media pendukung dalam mengelola data dan menginstalasi *package* penunjang pada sistem, sehingga sistem dapat dikonfigurasi dan berjalan sesuai keinginan. Monitor hanya akan menampilkan hasil dari data *event* yang sudah diolah dan diuraikan setelah berhasil melakukan pemesanan ruangan, dan data ini juga merupakan sebuah informasi yang akan dapat dilihat oleh pengguna pada sistem.

### 3.5.3 Perancangan Antarmuka Pengguna

Rancangan halaman antarmuka pengguna merupakan sebuah rancangan halaman yang dirancang sedemikian rupa yang hanya dikhususkan untuk pengguna pada sistem *Magic Mirror*. Halaman ini akan digunakan untuk melihat detail *event* yang sudah berhasil melakukan pemesanan ruangan oleh pengguna. Rancangan halaman antarmuka pengguna dilihat pada Gambar 3.3.

The screenshot shows a user interface with two weather widgets at the top. The left widget displays 'SPECIAL REGION YOGYAKARTA', '29.1°', and 'Feels like 33°'. The right widget displays 'Tuesday, December 15, 2020' and '12:28'. Below these is a table titled 'EVENTS' with two columns: 'CURRENT EVENTS' and 'UPCOMING EVENTS'. Each column contains one event entry with details like 'Penanggung Jawab', 'Ruang', 'Waktu', 'Deskripsi', and 'Status'.

EVENTS	
CURRENT EVENTS	UPCOMING EVENTS
Penanggung Jawab : Fahri Ruang : Gedung FIAI-lantai-1 Waktu : 02:30 PM - 03:00 PM Deskripsi : deskripsi Status : Ruangan sudah dibooking	Penanggung Jawab : Reza Ruang : Gedung FIAI-lantai-2 Waktu : 01:30 PM - 04:00 PM Deskripsi : deskripsi Status : Ruangan sudah dibooking

Gambar 3.3 Rancangan Halaman Tampil Data *Event*

Adapun penjelasan dari fitur halaman antarmuka pengguna dari sistem *Magic Mirror* adalah sebagai berikut:

1. Fitur cuaca

Fitur cuaca pada sistem berguna untuk memberikan informasi terkait cuaca terkini untuk daerah Yogyakarta.

2. Fitur waktu

Fitur waktu pada sistem berguna untuk memberikan informasi hari, tanggal, dan waktu secara *real time* menyesuaikan waktu di Indonesia. Dan fitur ini juga sangat berguna untuk mengingatkan pengguna akan *event* yang sedang berjalan pada hari ini.

3. Fitur *current events*

Fitur tabel *current events* pada sistem berguna untuk menampilkan detail informasi kepada pengguna terkait *event* yang sedang berjalan hanya untuk hari ini.

4. Fitur *upcoming events*

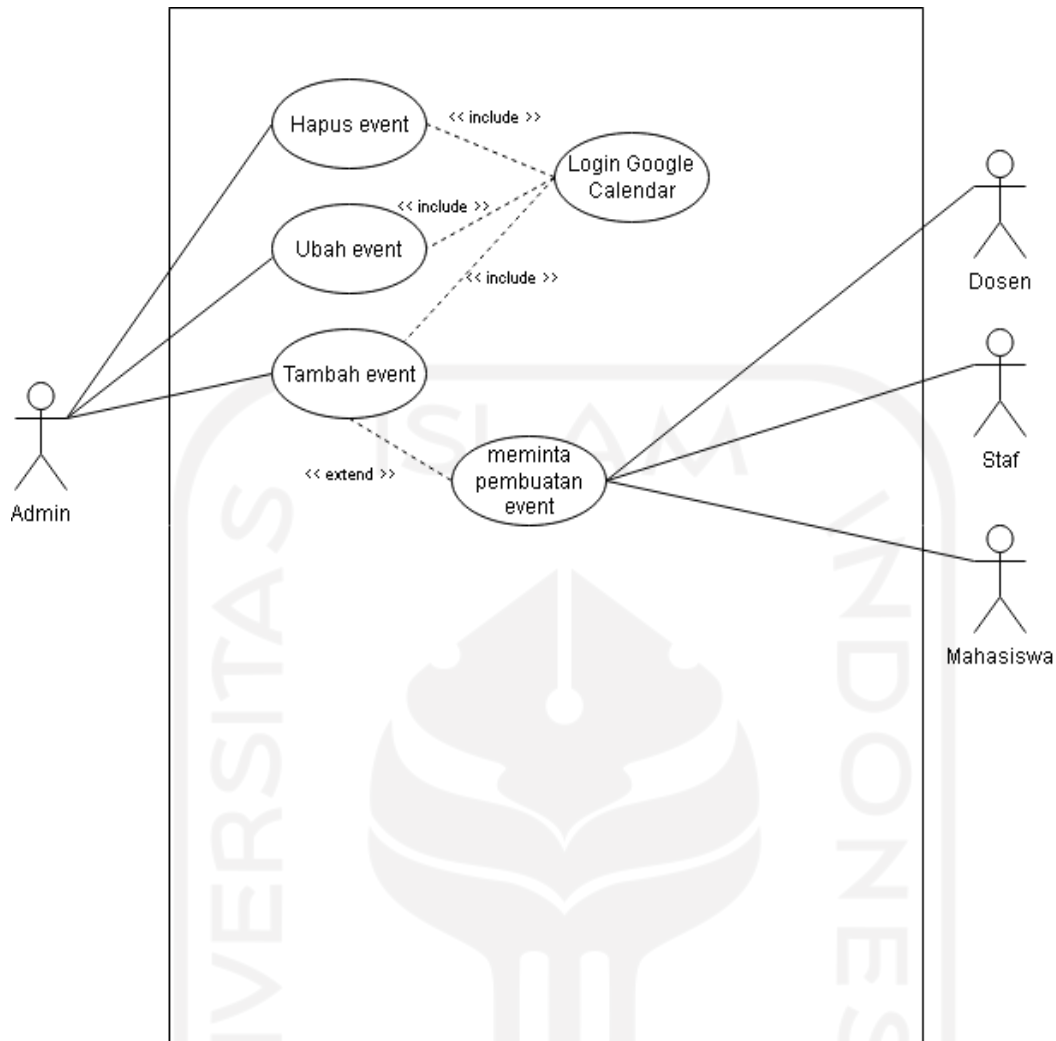
Fitur tabel *upcoming events* pada sistem berguna untuk menampilkan detail informasi kepada pengguna terkait *event* yang akan datang, sehingga pengguna dapat melihat perbedaan antara *Current event* dan *upcoming event*.

### 3.5.4 Use Case Diagram

*Use case diagram* merupakan sebuah representasi *behavior* atau interaksi antara sistem dan pengguna. Manfaat penggunaan *use case diagram* yaitu dapat melihat siapa saja yang akan berinteraksi dengan sistem, serta dapat melihat ruang lingkup dengan sistem. Dalam pembuatan sistem *Magic Mirror* menggunakan *Raspberry Pi*, sebelumnya sangat diperlukan membuat sebuah rancangan untuk menunjang kegunaan dari sebuah sistem. Pada penelitian ini terdapat dua buah *use case* diantaranya sebagai berikut:

a. Use Case Pemesanan Ruangan

*Use case diagram* dibawah menjelaskan mengenai interaksi aktor pada saat melakukan pemesanan ruangan pada layanan Google Calendar dan juga menjelaskan mengenai aktivitas yang dapat dilakukan oleh aktor. Alur pada saat melakukan pemesanan ruangan hanya memiliki empat aktor yang terlibat yaitu *admin*, dosen, staf, dan mahasiswa. Rancangan sistem pemesanan ruangan dilihat pada Gambar 3.4.



Gambar 3.4 Use Case Diagram Pemesanan Ruang

Adapun penjelasan keterangan hak akses *use case diagram* pemesanan ruangan yang akan dilihat pada Tabel 3.1.

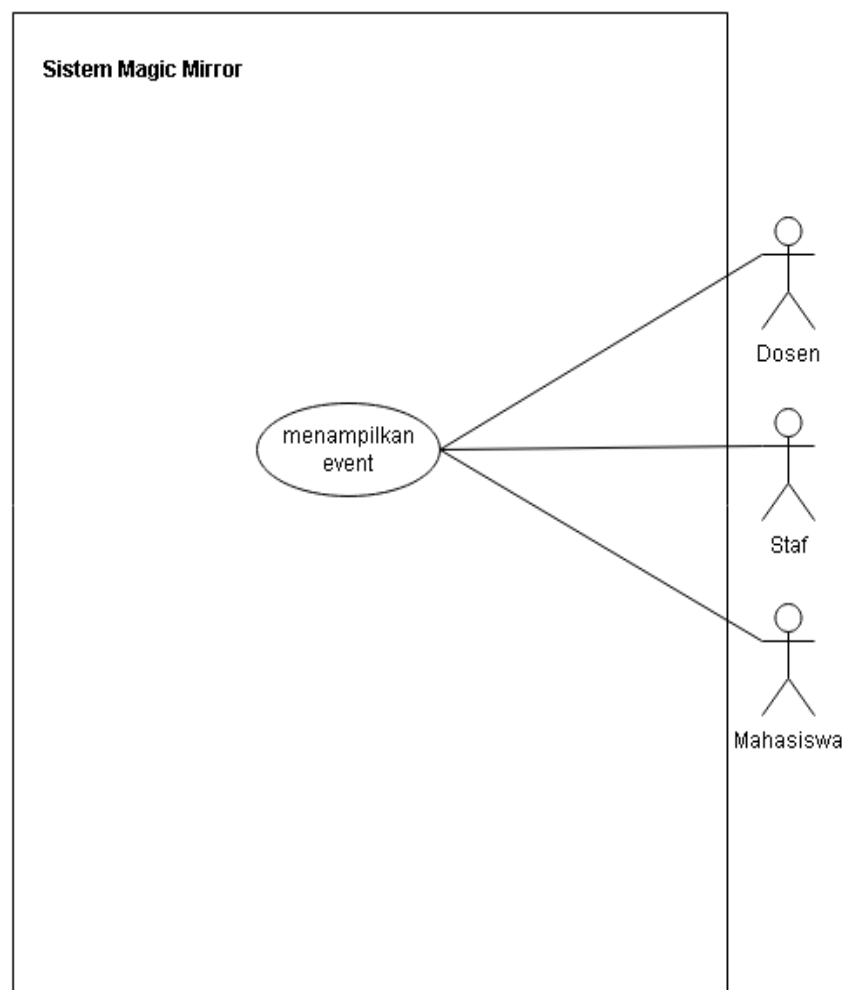
Tabel 3.1 Tabel Hak Akses

No	Nama Aktor	Proses	Hak Akses
1	<i>Admin</i>	<i>Login pada Google Calendar</i>	<i>Input email, dan Password</i>
2	<i>Admin</i>	<i>Menambah data event</i>	<i>Input nama event, nama tamu, nama ruangan, deskripsi, waktu mulai dan waktu selesai</i>
3	<i>Admin</i>	<i>Mengubah data event</i>	<i>Mengubah nama event, nama tamu, nama ruangan, deskripsi, waktu mulai dan waktu selesai</i>
4	<i>Admin</i>	<i>Menghapus data event</i>	<i>Hapus data event</i>
5	<i>Dosen</i>	<i>meminta pembuatan event</i>	<i>Dapat meminta pembuatan event meliputi nama event, hari,waktu mulai, dan waktu selesai</i>

6	Staf	meminta pembuatan <i>event</i>	Dapat meminta pembuatan event meliputi nama <i>event</i> , hari,waktu mulai, dan waktu selesai
5	Mahasiswa	Meminta pembuatan <i>event</i>	Dapat meminta pembuatan <i>event</i> meliputi nama <i>event</i> , hari,waktu mulai, dan waktu selesai

b. Use Case Magic Mirror

Dalam sistem *Magic Mirror* ini hanya memiliki tiga aktor yang terdiri dari dosen, staf, dan mahasiswa. Di mana ketiga aktor ini dapat melihat detail *event* pada sistem *Magic Mirror* setelah berhasil melakukan proses pemesanan ruangan sebelumnya. Rancangan sistem *Magic Mirror* dilihat pada Gambar 3.5.



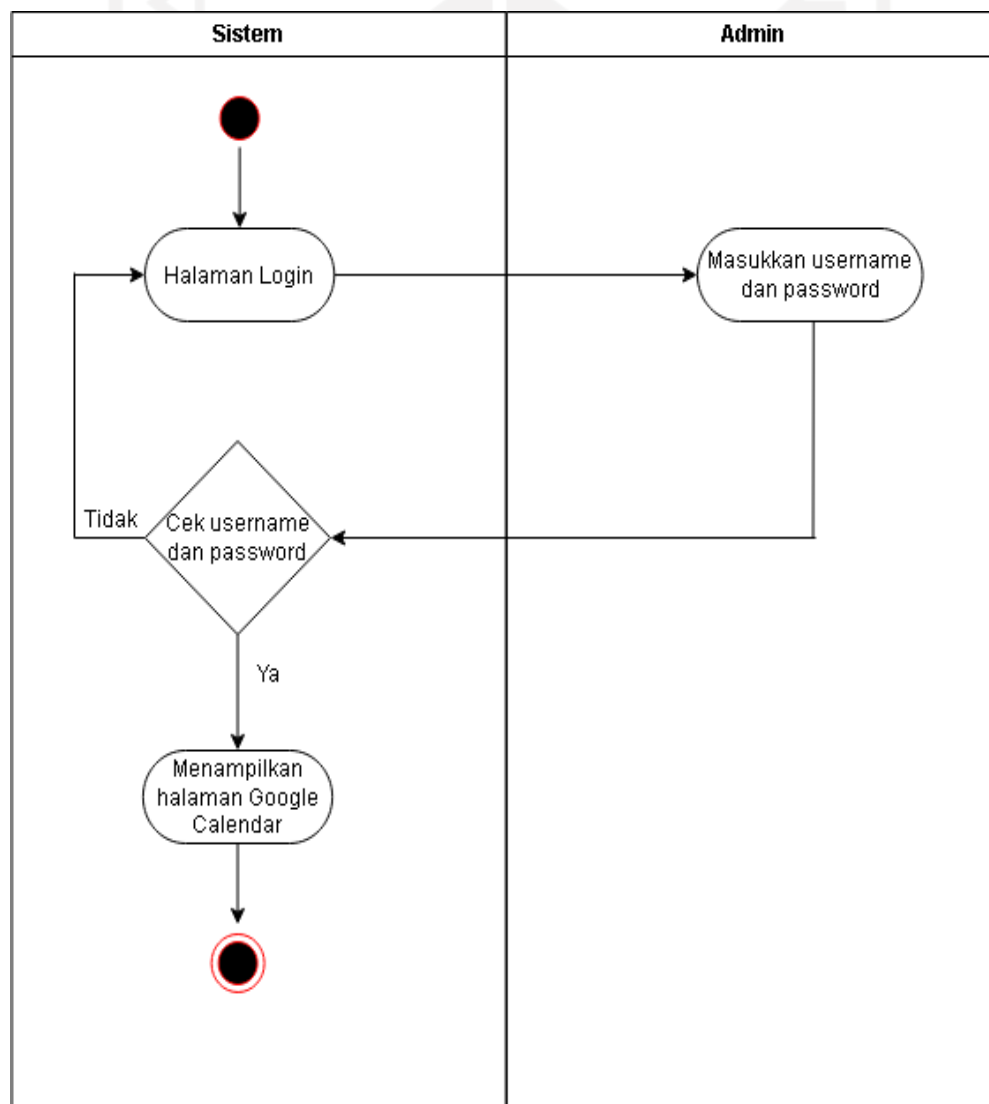
Gambar 3.5 Use Case Diagram Sistem Magic Mirror

### 3.5.5 Activity Diagram

*Activity Diagram* menggambarkan suatu aktivitas, berbagai alur aktivitas dalam sistem yang sedang dirancang, Bagaimana masing-masing alur berawal dari yang mungkin terjadi dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* untuk *admin* adalah alur yang dilakukan oleh *admin* saat mengakses Google Calendar.

#### 1. Activity Diagram Login

Pada *activity diagram* ini, *admin* dapat melakukan proses *login admin*. *Activity diagram login admin* dapat dilihat pada Gambar 3.6.

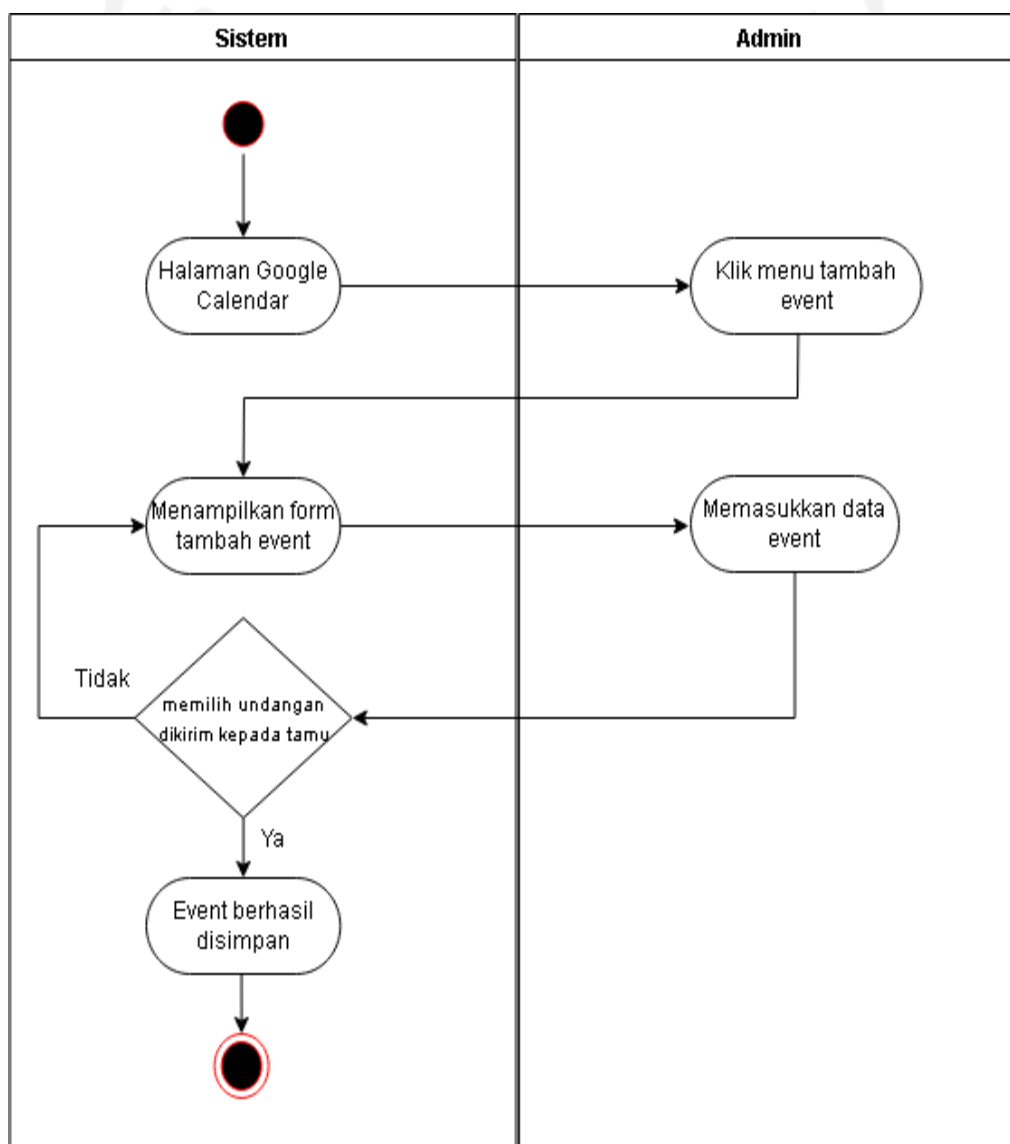


Gambar 3.6 Activity Diagram Login Admin

Pada *activity diagram login* seorang *admin* diminta untuk memasukkan *username* dan *password* terlebih dahulu. Kemudian *username* dan *password* tersebut akan melalui proses validasi oleh sistem. Jika *username* dan *password* benar maka akan diarahkan ke halaman beranda, tetapi jika *username* dan *password* salah maka akan kembali ke *form login*.

## 2. Activity Diagram Tambah Data Event

Pada *activity diagram* ini, *admin* dapat melakukan aksi tambah data *event*. *Activity diagram* tambah data *event* dapat dilihat pada Gambar 3.7.



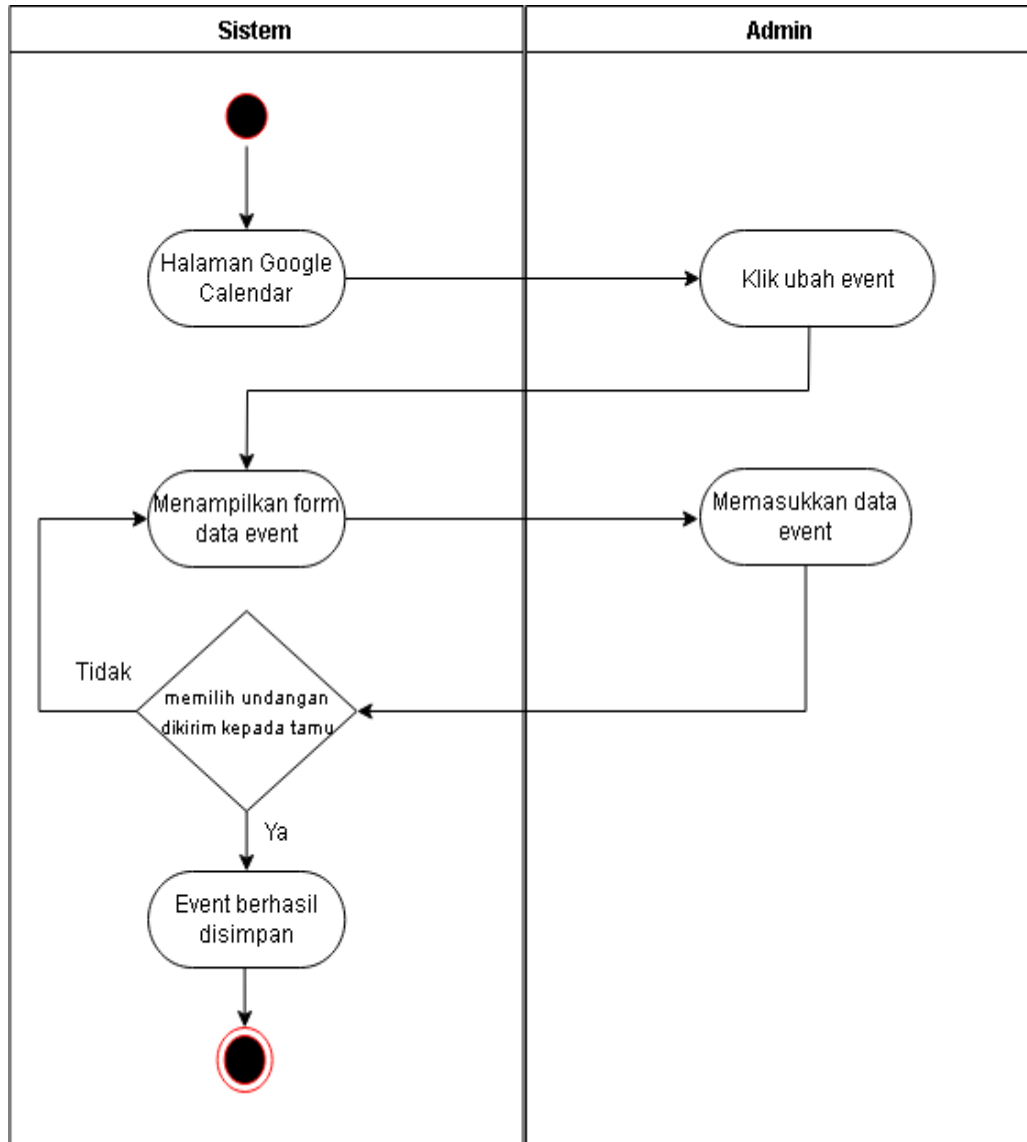
Gambar 3.7 Activity Diagram Tambah Data Event

*Activity diagram* yang dilihat pada Gambar 3.6 menggambarkan proses tambah data *event*. Proses ini diawali dengan memilih menu tambah data *event*. Kemudian ditampilkan *form* tambah data *event* yang berisi *form* nama *event*, nama tamu, nama ruangan, deskripsi, waktu mulai dan waktu selesai. Kemudian penambahan *event* akan melalui proses validasi oleh sistem. Jika memilih ya maka data *event* akan disimpan, tetapi jika memilih tidak maka akan kembali pada *form* penambahan *event*.

### 3. Activity Diagram Ubah Data Event

Pada *activity diagram* ini, *admin* dapat melakukan aksi ubah data *event*. *Activity diagram* ubah data *event* dapat dilihat pada Gambar 3.8.





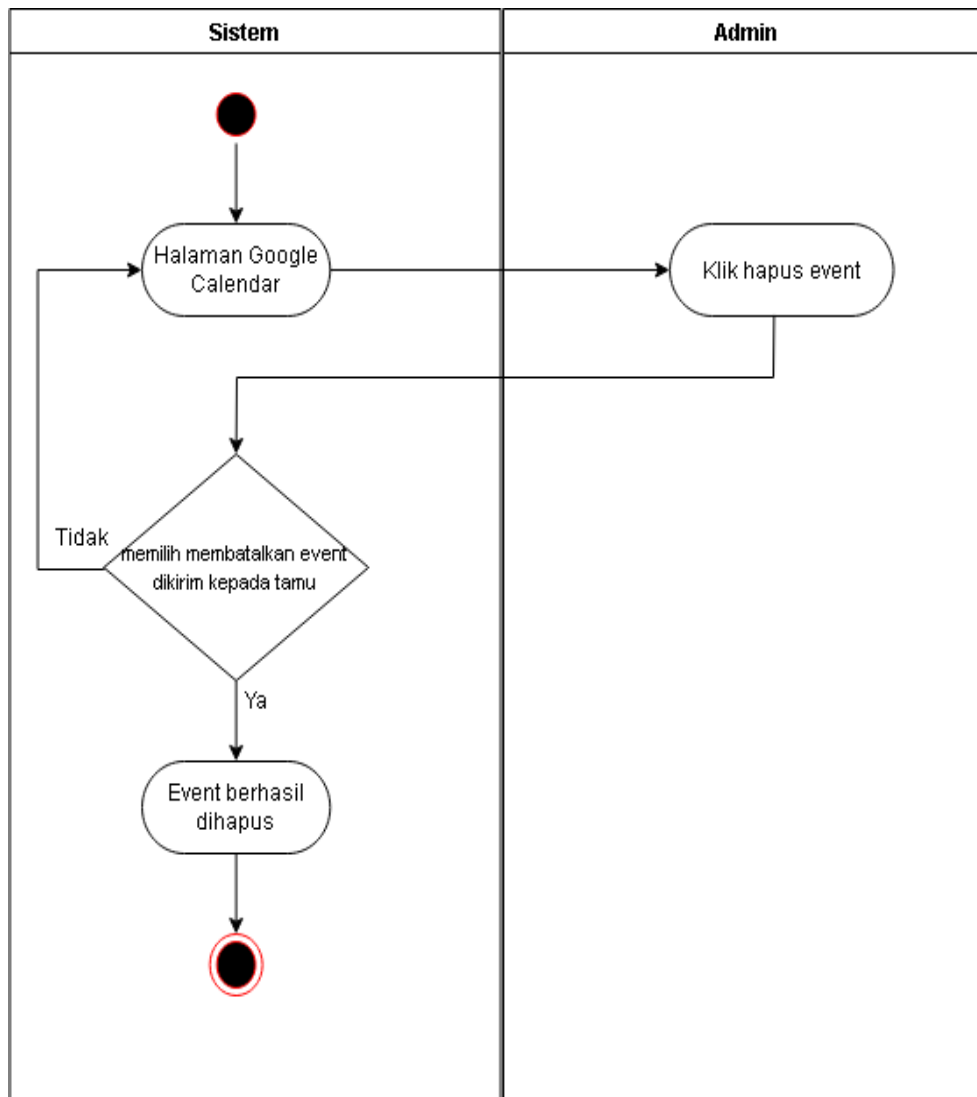
Gambar 3.8 Activity Diagram Ubah Data Event

Activity diagram yang dilihat pada Gambar 3.8 menggambarkan proses ubah data event. Proses ubah data event hampir sama dengan proses tambah data event. Proses ini diawali dengan memilih menu ubah data event yang diinginkan. Kemudian ditampilkan form ubah data event yang berisi form nama event, nama tamu, nama ruangan, deskripsi, waktu mulai dan waktu selesai. Kemudian perubahan data event akan melalui proses validasi oleh sistem. Jika memilih ya maka data event akan disimpan, tetapi jika memilih tidak maka akan kembali pada form data event.



#### 4. Activity Diagram Hapus Data Event

Pada *activity diagram* ini, *admin* dapat melakukan aksi hapus data *event*. *Activity diagram* hapus data *event* dapat dilihat pada Gambar 3.9.



Gambar 3.9 Activity Diagram Hapus Data Event

*Activity diagram* yang dilihat pada Gambar 3.9 menggambarkan proses hapus data *event*. Proses ini diawali dengan memilih menu hapus data *event* yang diinginkan. Sebelum proses hapus data di eksekusi maka akan muncul peringatan apakah anda yakin untuk menghapus *event* ini terlebih dahulu. Jika memilih ya maka data *event* akan terhapus, tetapi jika memilih tidak maka data *event* tidak akan dihapus.

## BAB IV

### IMPLEMENTASI DAN PEMBAHASAN

Implementasi dan pembahasan akan membahas tahapan perancangan sistem *Magic Mirror* yang memanfaatkan Google Calendar API menggunakan *Raspberry Pi*. Pada tahapan implementasi ini akan dibahas mengenai hasil dan analisis dari perancangan sistem yang sudah berhasil dikembangkan.

#### 4.1 Implementasi

Implementasi merupakan tahapan yang dilakukan oleh penulis dalam mengimplementasikan perancangan yang sebelumnya sudah dilakukan. Pada tahapan ini implementasi akan memberikan hasil dari sistem *Magic Mirror* dengan memanfaatkan Google Calendar API menggunakan *Raspberry Pi* yang akan berjalan sesuai dengan fungsinya. Pada tahap implementasi penjelasan akan dibagi menjadi tiga, yang pertama adalah implementasi perangkat keras, yang kedua adalah implementasi perangkat lunak, dan yang ketiga adalah implementasi antarmuka pengguna.

##### 4.1.1 Perangkat Keras yang Digunakan

Perangkat keras yang digunakan sebagai perangkat pendukung dalam melakukan perancangan sistem *Magic Mirror* menggunakan *Raspberry Pi*. Perangkat keras yang digunakan oleh penulis dalam menunjang rancangan sistem adalah sebagai berikut:

- a. Laptop MSI GL62 Core i7-7700HQ 24 GB RAM
- b. *Raspberry Pi 3* Model B CPU Quad Core 1.2GHz Broadcom BCM2837 64bit dan 1GBRAM
- c. Monitor 18 *inch*

##### 4.1.2 Perangkat Lunak yang Digunakan

Dalam implementasinya sistem *Magic Mirror* menggunakan *Raspberry Pi* dibutuhkan perangkat lunak yang digunakan untuk menunjang pengembangan pada sistem. Perangkat lunak yang digunakan sebagai penunjang pada pengembangan sistem adalah sebagai berikut:

- a. Sistem Operasi

Sistem Operasi merupakan perangkat lunak yang digunakan penulis untuk mengatur sumber daya dari perangkat keras dan perangkat lunak yang ada pada komputer. Dalam perancangan sistem *Magic Mirror* ini digunakan sistem operasi Ubuntu 18.04.5 LTS.

b. Linux Terminal

Linux Terminal adalah perangkat lunak yang digunakan untuk menjalankan perintah dasar dalam komputer. Pada penelitian ini digunakan Linux Terminal untuk melakukan konfigurasi pada *Raspberry Pi*.

d. Vinagre

Vinagre adalah klien VNC, SSH, RDP dan SPICE untuk lingkungan desktop GNOME pada sistem operasi Linux. Vinagre memiliki beberapa fitur, seperti kemampuan untuk terhubung ke beberapa *server* secara bersamaan. Pada penelitian ini digunakan Vinagre untuk melakukan koneksi ke perangkat keras yang terhubung internet melalui protokol VNC dengan menggunakan *port* 5900.

e. Postman

Postman merupakan *tools* yang digunakan untuk *developer* yang bertujuan untuk menguji API. Pada penelitian ini digunakan postman untuk melakukan pengujian Google Calendar API.

#### 4.1.3 Implementasi Perancangan Perangkat Keras

Pada tahap implementasi perangkat keras dibutuhkan beberapa perangkat keras yang sudah ditetapkan sebelumnya pada tahap analisis perangkat keras dalam Bab 3. Perangkat keras yang diperlukan adalah sebagai berikut:

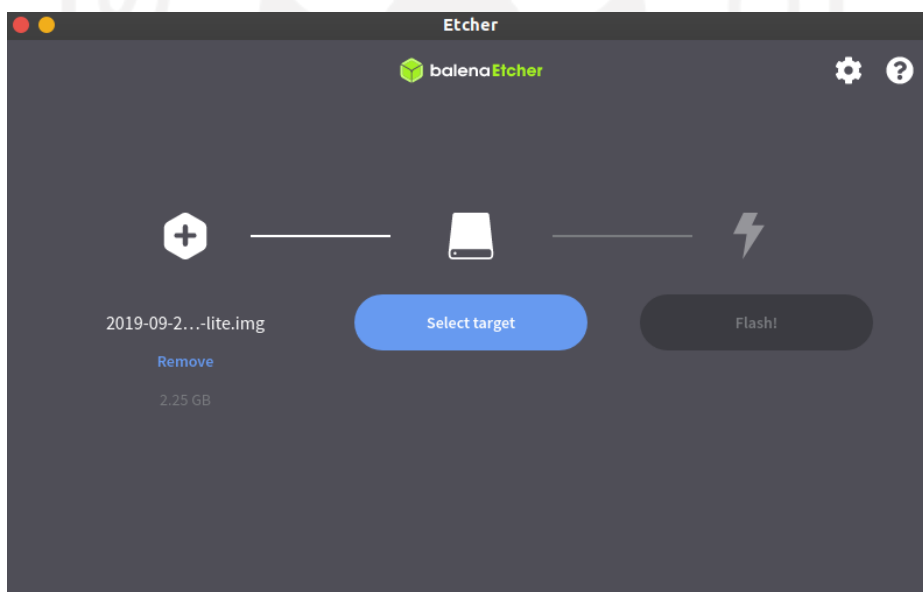
- a. *Raspberry Pi 3 Model B*
- b. MicroSD card 16 GB
- c. Kabel USB
- d. *Adaptor power 5V*
- e. Cermin 2 arah
- f. Monitor 18 inch
- g. Kabel HDMI

Kemudian semua perangkat keras yang diperlukan telah tersedia, selanjutnya yang harus dilakukan adalah melakukan instalasi pada perangkat keras. Langkah-langkah dalam instalasi pada perangkat keras yang dijabarkan sebagai berikut:

#### a. Instalasi dan konfigurasi *Raspberry Pi*

Langkah pertama yang dilakukan dalam instalasi dan konfigurasi *Raspberry Pi* adalah mengunduh sistem operasi *Raspberry Pi OS* yang akan digunakan sebagai sistem operasi yang berjalan. Untuk mengunduh sistem operasi *Raspberry Pi OS* dapat dilakukan pada halaman resmi dari *Raspberry Pi*. Terdapat beberapa versi dalam sistem operasi *Raspberry Pi OS*, namun dalam penelitian ini hanya digunakan *Raspberry Pi OS Desktop*.

Langkah selanjutnya adalah membuat *bootable disk Raspberry Pi OS* dengan sistem operasi yang sudah diunduh sebelumnya ke dalam MicroSD. Pada tahap ini digunakan perangkat lunak Balena Etcher sebagai media pembuatan *bootable disk*. Cara membuatnya cukup sederhana hanya dengan memilih *file* sistem operasi yang sudah diunduh dan media targetnya yaitu MicroSD seperti yang terlihat pada Gambar 4.1. Setelah itu sistem operasi *Raspberry Pi OS* akan melakukan instalasi secara otomatis saat dihidupkan dengan menghubungkan perangkat *Raspberry Pi* pada *power adaptor* seperti pada Gambar 4.2.



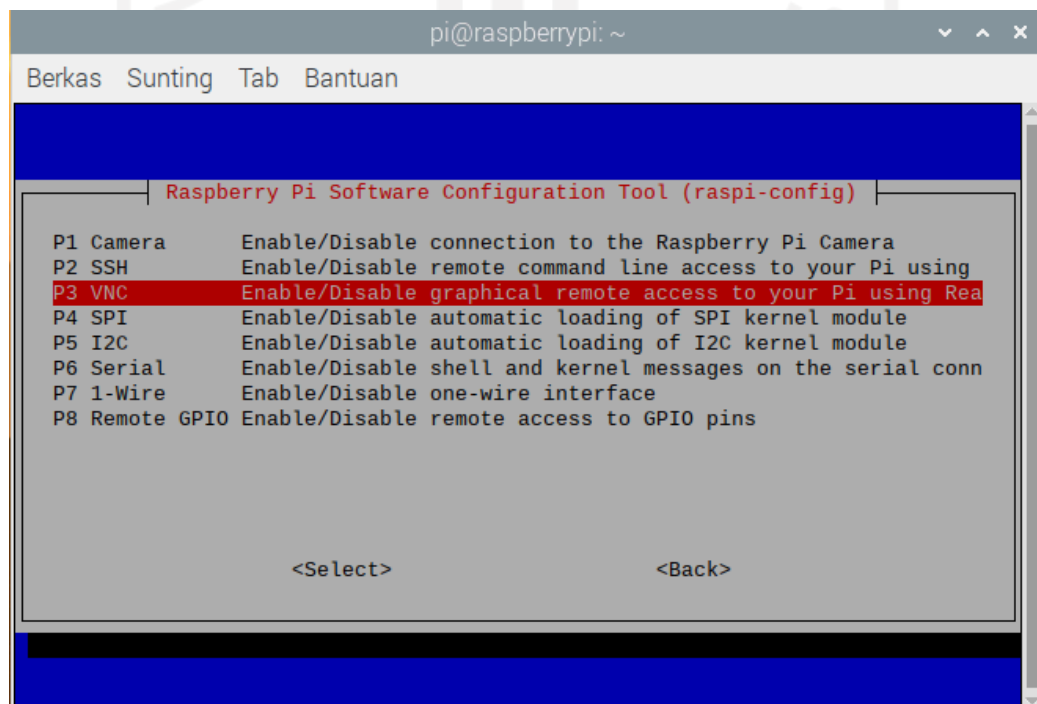
Gambar 4.1 Proses membuat *bootable disk*

Setelah perangkat dihubungkan dengan *power adaptor* menggunakan kabel USB, perangkat bisa dihubungkan dengan perangkat *keyboard* dan layar monitor untuk melihat progres instalasi sistem operasi dan melakukan konfigurasi awal.



Gambar 4.2 Pemasangan *Raspberry Pi* dengan power adaptor

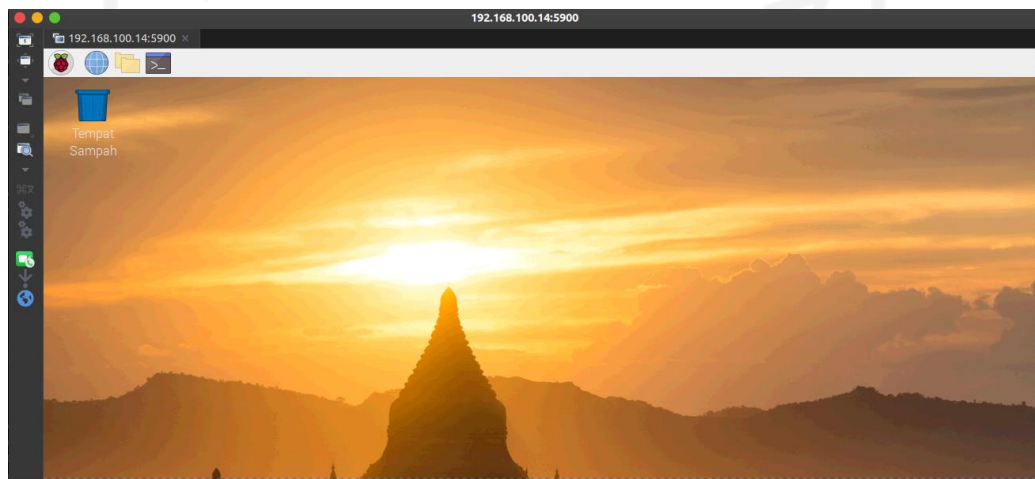
Konfigurasi awal yang dilakukan adalah melakukan aktivasi protocol VNC pada perangkat. VNC berguna untuk mengakses perangkat secara *remote* melalui internet. Untuk mengaktifkan VNC dilakukan menggunakan perintah “*raspi-config*” pada terminal dan memilih “*enable VNC*” pada bagian interface seperti yang terlihat pada Gambar 4.3.



Gambar 4.3 Konfigurasi aktivasi VNC pada *Raspberry Pi*

#### 4.1.4 Implementasi Perancangan Perangkat Lunak

Implementasi perancangan perangkat lunak ini adalah penerapan dari perancangan beberapa perangkat lunak yang sudah disebutkan pada Bab 3. Pada implementasinya perancangan perangkat lunak dilakukan dengan menggunakan akses *remote* pada perangkat *Raspberry Pi* untuk pengembangan sistem. Akses *remote* dilakukan menggunakan protokol VNC menggunakan *port* 5900 dan dilakukan menggunakan perangkat lunak Vinagre seperti pada Gambar 4.4. Untuk melakukan akses *remote* diperlukan data *user* dan *password* dari *Raspberry Pi*. Data *user* yang otomatis diberikan oleh sistem saat instalasi adalah “pi” dengan password “raspberrry”.



Gambar 4.4 Koneksi *remote* menggunakan Vinagre

Pada tahap implementasi perancangan perangkat lunak, penulis membagi implementasi menjadi tiga bagian yaitu implementasi konfigurasi *Magic Mirror*, *open weather map*, Google Calendar. Implementasi perancangan perangkat lunak pada sistem *Magic Mirror* menggunakan *Raspberry Pi* adalah sebagai berikut:

##### a. Instalasi *Magic Mirror*

Langkah pertama yang dilakukan dalam instalasi *Magic Mirror* pada *Raspberry Pi* dapat dilihat pada halaman dokumentasi resmi *Magic Mirror*. Adapun langkah yang diperlukan untuk instalasi *Magic Mirror* pada *Raspberry Pi* yaitu dengan mengeksekusi perintah pada Tabel 4.1.

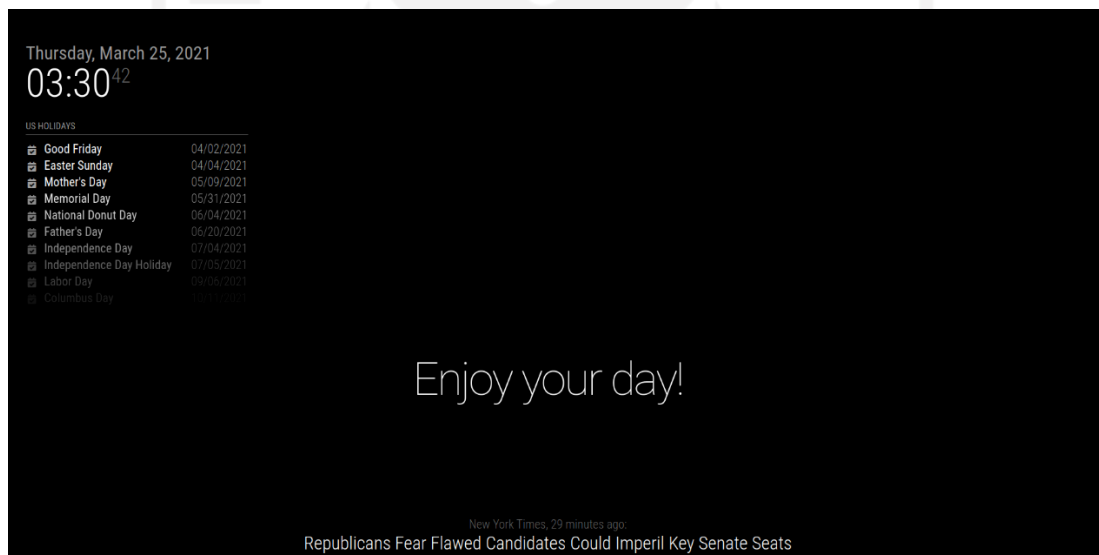
Tabel 4.1 Perintah instalasi *Magic Mirror*

```
$ bash -c "$(curl -sL
https://raw.githubusercontent.com/sdetweil/MagicMirror_scripts/master/raspberry.sh)"
```

Perintah yang dilakukan pada Tabel 4.1 adalah perintah yang melakukan instalasi perangkat lunak *Magic Mirror* pada *Raspberry Pi*. Setelah melakukan instalasi kemudian langkah selanjutnya adalah menjalankan sistem dengan mengeksekusi perintah pada Tabel 4.2. Terlampir pada Gambar 4.5 merupakan halaman *interface* pada sistem yang sudah berhasil di *install*.

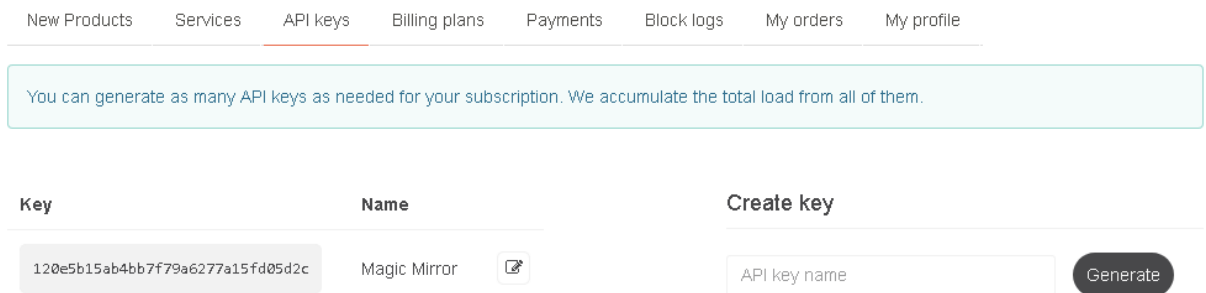
Tabel 4.2 Perintah menjalankan *Magic Mirror*

```
$ npm run start
```

Gambar 4.5 Tampilan *Magic Mirror*

#### b. Implementasi Konfigurasi Open Weather Map

Pada penelitian ini digunakan Open Weather Map untuk melihat cuaca terkini yang digunakan pada sistem *Magic Mirror*. Untuk dapat terhubung pada layanan Open Weather Map dapat dilakukan dengan memanfaatkan API (*Application Programming Interface*) seperti pada Gambar 4.6.



Gambar 4.6 *Generate Open Weather Map API*

Setelah berhasil membuat API pada gambar 4.6 perlu dilakukan konfigurasi pada *file* `config.js` yang berada pada folder konfigurasi. *File* tersebut berguna untuk mendefinisikan *module* yang akan digunakan, mulai dari *location*, *location ID*, *API*, dan konfigurasi lainnya. Isi konfigurasi *file* `config.js` pada penelitian ini dapat dilihat pada Tabel 4.3. Terlampir pada Gambar 4.7 merupakan halaman *interface* cuaca yang sudah berhasil di konfigurasi.

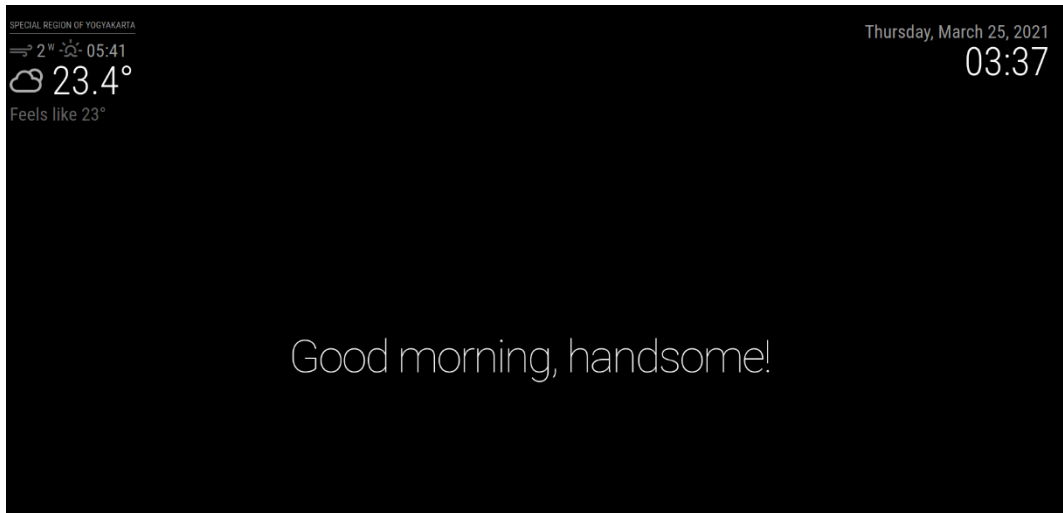
Tabel 4.3 Kode program Open Weather Map API

```

...
{
  module: "currentweather",
  position: "top_left",
  config: {
    showSun: true,
    location: "Yogyakarta",
    locationID:"1621176", //ID from
    http://bulk.openweathermap.org/sample/city.list.json.gz; unzip the gz file and
    find your city
  },
  #Open Weather Map API
  appid: "120e5b15ab4bb7f79a6277a15fd05d2c",
  type: "current",
},]
...

```

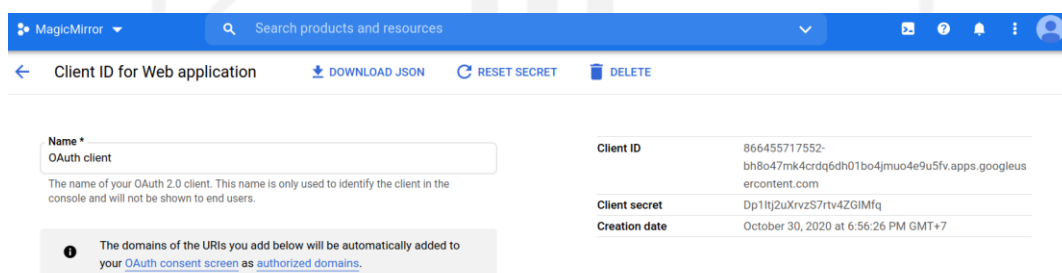




Gambar 4.7 Tampilan cuaca pada sistem

### c. Penerapan Konfigurasi Google Calendar API (*Application Programming Interface*)

Salah satu layanan yang disediakan oleh Google ini merupakan sebuah layanan yang dapat digunakan untuk melakukan penjadwalan. Untuk dapat mengakses Google Calendar dari akun pribadi, maka diperlukan sebuah *key* yang bisa didapat melalui halaman awal Google Calendar API pada tautan terlampir <https://console.cloud.google.com/apis/credentials> yang nantinya dapat di *download* dengan format *file .json*. Sehingga layanan Google ini dapat berinteraksi langsung pada sistem *Magic Mirror* dengan standar keamanan seperti Gambar 4.8.



Gambar 4.8 Kunci keamanan untuk akses Google Calendar API

Langkah selanjutnya yang harus dilakukan adalah mengunduh modul *googleapis* menggunakan *NPM (Node Package Manager)*, dikarenakan akan kebutuhan autentikasi pada layanan Google Calendar API menggunakan *oauth*. Untuk mengunduh modul *googleapis* dapat dilakukan menggunakan perintah seperti pada Tabel 4.4.

Tabel 4.4 Perintah mengunduh googleapis

```
$ npm install googleapis@39 --save
```

Setelah mengunduh dependensi googleapis selanjutnya mengunduh *file* json pada tautan berikut <https://console.cloud.google.com/apis/credentials> , lalu ganti nama *file* *.json* yang sudah di unduh menjadi *credentials*. Pada *file credentials* perlu adanya modifikasi kode seperti pada Tabel 4.5.

Tabel 4.5 Kode program *credentials*

```
{
  "installed": {
    "client_id": "866455717552-bh8o47mk4crdq6dh01bo4jmuo4e9u5fv.apps.googleusercontent.com",
    "project_id": "magicmirror-1604058952122",
    "auth_uri": "https://accounts.google.com/o/oauth2/auth",
    "token_uri": "https://oauth2.googleapis.com/token",
    "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
    "client_secret": "Dp1Itj2uXrvzS7rtv4ZGIMfq",
    "redirect_uris": ["https://developers.google.com/oauthplayground/"]
  }
}
```

Langkah selanjutnya adalah membuat *file* baru dengan nama *index.js* yang berisi kode program seperti pada Tabel 4.6. Pada cakupan penelitian ini hanya memiliki hak akses *read and write* untuk *events* pada layanan Google calendar, dan untuk melihat informasi cakupan pada layanan Google Calendar dapat dilihat pada halaman resmi dari Google.

Tabel 4.6 Kode program *index.js*

```
const fs = require('fs');
const readline = require('readline');
const {google} = require('googleapis');

const SCOPES = ['https://www.googleapis.com/auth/calendar.events'];
const TOKEN_PATH = 'token.json';

fs.readFile('credentials.json', (err, content) => {
  if (err) return console.log('Error loading client secret file:', err);
  // Authorize a client with credentials, then call the Google Calendar API.
  authorize(JSON.parse(content), listEvents);
});

function authorize(credentials, callback) {
  const {client_secret, client_id, redirect_uris} = credentials.installed;
  const oAuth2Client = new google.auth.OAuth2(
    client_id, client_secret, redirect_uris[0]);

  // Check if we have previously stored a token.
  fs.readFile(TOKEN_PATH, (err, token) => {
    if (err) return getAccessToken(oAuth2Client, callback);
    oAuth2Client.setCredentials(JSON.parse(token));
    callback(oAuth2Client);
  });
}
```

```

function getAccessToken(oAuth2Client, callback) {
  const authUrl = oAuth2Client.generateAuthUrl({
    access_type: 'offline',
    scope: SCOPES,
  });
  console.log('Authorize this app by visiting this url:', authUrl);
  const rl = readline.createInterface({
    input: process.stdin,
    output: process.stdout,
  });
  rl.question('Enter the code from that page here: ', (code) => {
    rl.close();
    oAuth2Client.getToken(code, (err, token) => {
      if (err) return console.error('Error retrieving access token', err);
      oAuth2Client.setCredentials(token);
      // Store the token to disk for later program executions
      fs.writeFile(TOKEN_PATH, JSON.stringify(token), (err) => {
        if (err) return console.error(err);
        console.log('Token stored to', TOKEN_PATH);
      });
      callback(oAuth2Client);
    });
  });
}

function listEvents(auth) {
  const calendar = google.calendar({version: 'v3', auth});
  calendar.events.list({
    calendarId: 'primary',
    timeMin: (new Date()).toISOString(),
    maxResults: 10,
    singleEvents: true,
    orderBy: 'startTime',
  }, (err, res) => {
    if (err) return console.log('The API returned an error: ' + err);
    const events = res.data.items;
    if (events.length) {
      console.log('Upcoming 10 events:');
      events.map((event, i) => {
        const start = event.start.dateTime || event.start.date;
        console.log(`${start} - ${event.summary}`);
      });
    } else {
      console.log('No upcoming events found.');
```

Agar dapat terhubung dengan Google Calendar diperlukan adanya autentikasi terlebih dahulu dengan mengeksekusi perintah pada Tabel 4.7.

Tabel 4.7 Perintah autentikasi

```
$ node index.js
```

Setelah berhasil melakukan autentikasi pada Google Calendar, maka terdapat *authorization code* yang nantinya akan di *input* pada terminal. Selanjutnya *authoziation code* ini akan menghasilkan sebuah *file json* baru yaitu *token* sebagai *output* dari hasil autentikasi

yang dilakukan sebelumnya, dan *token* ini akan disimpan pada sistem *Magic Mirror* seperti Gambar 4.9.

```
pi@raspberrypi:~/MagicMirror $ node index.js
Authorize this app by visiting this url: https://accounts.google.com/o/oauth2/v2/auth?access_type=offline&scope=calendar.events%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcalendar&response_type=code&client_id=866455717552-bh8c...
nt.com&redirect_uri=https%3A%2F%2Fdevelopers.google.com%2Foauthplayground%2F
Enter the code from that page here: 4/0AY0e-g7c74y20FSFSbmnySq-joDcd5HLC-bT-0Mx06RxDP7ci0e02c-asUwwoPror0I0DA
Token stored to token.json
ts
[]
ini ts
BEGIN:VCALENDAR
VERSION:2.0
PRODID://lanyrd.com//Lanyrd//EN
X-ORIGINAL-URL:http://lanyrd.com/2016/xpdays/xpdays-schedule.ics
X-WR-CALNAME;CHARSET=utf-8:XP Days Ukraine 2016 schedule
METHOD:PUBLISH
X-MS-OLK-FORCEINSPECTOROPEN:TRUE
END:VCALENDAR
ts done
Saved!
pi@raspberrypi:~/MagicMirror $
```

Gambar 4.9 Mendapatkan *file* token

Langkah selanjutnya adalah mengurai objek data dari Google Calendar yang sudah berhasil didapatkan. Dikarenakan adanya pemilihan informasi terkait data yang ingin ditampilkan pada sistem *Magic Mirror*, perlu ditambahkan kode program seperti Tabel 4.8 pada *file* index.

Tabel 4.8 Kode program *parsing* objek

```
...
//jsonToicl.js
const beginICalendar = "BEGIN:VCALENDAR\nVERSION:2.0\nPRODID:-
//lanyrd.com//Lanyrd//EN\nX-ORIGINAL-URL:http://lanyrd.com/2016/xpdays/xpdays-
schedule.ics\nX-WR-CALNAME;CHARSET=utf-8:XP Days Ukraine 2016
schedule\nMETHOD:PUBLISH\nX-MS-OLK-FORCEINSPECTOROPEN:TRUE\n";
const endICalendar = "END:VCALENDAR";
...
function eventToICalendar(event, googleEvents){
  var newLine = "\n";
  googleEvents += "BEGIN:VEVENT\n";
  googleEvents += "CLASS:" + "\n";
  googleEvents += "SUMMARY:" + event.summary + "\n";
  googleEvents += "LOCATION:" + event.location + "\n";
  googleEvents += "UID:" + event.id + "\n";
  googleEvents += "DTSTAMP:" + dtStamp + "\n";
  googleEvents += "DTSTART:" + moment(event.start.dateTime).format("YYYY-MM-DD ,
hh:mm a")+ "\n" ;
}
```

```

    googleEvents += "DTEND:" + moment(event.end.dateTime).format("YYYY-MM-DD,
hh:mm a") + "\n" ;
    googleEvents += "DESCRIPTION:"+ event.description + "\n";
    //diplayname penanggung jawab lurd
    googleEvents += "ATTENDEES:"+ event.attendees[0].displayName + "\n";
    //nambahin status
    googleEvents += "ATTENDEES:" + event.attendees[2].responseStatus + "\n";
    googleEvents += "END:VEVENT\n";
    return googleEvents;
  }}
  ...

```

Setelah menambahkan kode program pada Tabel 4.8 terlampir pada Gambar 4.10 data yang sudah berhasil diurai sesuai kebutuhan yang sudah disebutkan pada Bab 3.

```

[2021-04-16 17:41:53.521] [LOG] {
  '1pm9mnc0ob028phk7brvr034in': {
    type: 'VEVENT',
    params: [],
    class: '',
    summary: 'Test Event current',
    location: 'Gedung FIAI-1-Lab Komputer (50)',
    uid: '1pm9mnc0ob028phk7brvr034in',
    dtstamp: '20210416T174252',
    start: '2021-04-16 , 07:00 pm',
    end: '2021-04-16, 08:00 pm',
    description: 'test current event',
    attendees: 'ADE BAGAS PERMATA PUTRA'
  },
  '6971rg4eee1e16omhkvocvclte': {
    type: 'VEVENT',
    params: [],
    class: '',
    summary: 'Test Event upcoming',
    location: 'Gedung FIAI-1-Lantai 1-RSEMU (100)',
    uid: '6971rg4eee1e16omhkvocvclte',
    dtstamp: '20210416T174252',
    start: '2021-04-17 , 09:30 pm',
    end: '2021-04-17, 10:30 pm',
    description: 'Test event upcoming',
    attendees: 'MUHAMMAD RYAN APRIYUDAN -'
  }
}
[2021-04-16 17:41:54.451] [LOG] Shutting down server...
[2021-04-16 17:41:54.453] [LOG] Stopping module helper: updatenotification
[2021-04-16 17:41:54.454] [LOG] Stopping module helper: MMM-MyCalendar
pi@raspberrypi:~/MagicMirror $

```

Gambar 4.10 *parsing* data

#### 4.1.5 Implementasi Perancangan Antarmuka Pengguna

Implementasi perancangan antarmuka pengguna ini adalah penerapan dari perancangan antarmuka pengguna yang sudah disebutkan pada Bab 3. Pada tahap implementasi perancangan antarmuka pengguna, penulis membagi implementasi menjadi tiga bagian yaitu implementasi konfigurasi *module calendar*, penerapan tabel *current event*, dan penerapan tabel *upcoming event*. Implementasi perancangan antarmuka pengguna pada sistem penjadwalan *Magic Mirror* menggunakan *Raspberry Pi* adalah sebagai berikut:

### a. Implementasi Konfigurasi Module Calendar

Langkah pertama yang dilakukan dalam instalasi *module calendar* pada *Magic Mirror* dapat dilihat pada halaman dokumentasi resmi MMM-MyCalendar. Adapun cara yang diperlukan untuk mengunduh MMM-MyCalendar pada *Magic Mirror* yaitu dengan mengeksekusi perintah pada Tabel 4.9.

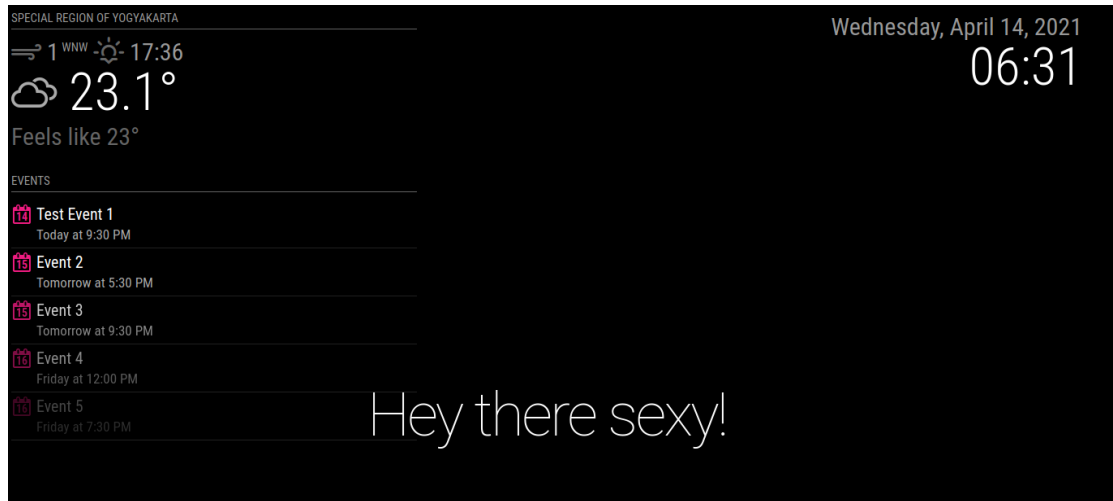
Tabel 4.9 Perintah mengunduh modul MMM-MyCalendar

```
$ cd MagicMirror/modules
$ git clone https://github.com/jclarke0000/MMM-MyCalendar.git
```

Pada saat setelah berhasil mengunduh modul MMM-MyCalendar perlu dilakukan konfigurasi pada *file config.js* yang berada pada folder konfigurasi. *File* tersebut berguna untuk mendefinisikan *module* yang akan digunakan, mulai dari *position*, *date format*, *color*, dan konfigurasi lainnya. Bagian isi konfigurasi *file config.js* pada penelitian ini dapat dilihat pada Tabel 4.10. Terlampir pada Gambar 4.11 merupakan halaman antarmuka pengguna yang sudah diintegrasikan dengan module MMM-MyCalendar.

Tabel 4.10 Kode program MMM-MyCalendar

```
...
{
  module: "MMM-MyCalendar",
  header: "Events",
  position: "top_left",
  config: {
    maximumEntries: "10",
    colored: true,
    animationSpeed: "1000",
    defaultSymbol: "calendar",
    dateFormat: "MMM Do",
    showEnd: true,
    calendars: [
      {
        symbol: "MyCalendar",
        url: `https:// google.com/`,
        color: "#ed1c80",
      },
    ],
  },
},
},
```



Gambar 4.11 Menambahkan Modul MMM-MyCalendar

#### b. Penerapan Tabel Upcoming Event

Langkah selanjutnya adalah melakukan pembuatan tabel *upcoming events* dengan cara menambahkan kode program pada *file* `modules/MMM-MyCalendar/MMM-mycalendar.js` seperti Tabel 4.11.

Tabel 4.11 Kode program *upcoming event*

```

var bungkus = document.createElement("table");
var thead = document.createElement("thead");
var tr_head = document.createElement("tr");
var th1 = document.createElement("th");
var th2 = document.createElement("th");
//body
var tbody = document.createElement("tbody");
var tr_body = document.createElement("tr");
var td_kiri = document.createElement("td");
var td_kanan = document.createElement("td");
var td_tengah = document.createElement("td");
...
        bungkus.style.cssText = 'background-color: ; width: 100%';
th1.innerHTML = "Current Events";
th2.innerHTML = "Upcoming Events";
//thead
tr_head.appendChild(th1);

```

```

tr_head.appendChild(th2);
thead.appendChild(tr_head);
bungkus.appendChild(thead);
//tbody
td_kiri.appendChild(wrapper);
tr_body.appendChild(td_kanan);
td_kanan.className = 'small';
tr_body.appendChild(td_kiri);
tr_body.appendChild(td_kanan);
tbody.appendChild(tr_body);
bungkus.appendChild(tbody);
return bungkus;
}

```

Pada file *MMM-mycalendar.js* merupakan konfigurasi tampilan awal pada sistem *Magic Mirror*, pada penerapannya kode program tersebut hanya akan menambahkan sebuah tabel baru. Tabel *upcoming events* hanya berisikan informasi yang berfokus pada *event* yang akan datang, terlampir pada Gambar 4.12 merupakan hasil antarmuka pengguna yang sudah berhasil di konfigurasi.



Gambar 4.12 Antarmuka pengguna Tabel *upcoming event*

### c. Penerapan Tabel Current Event

Langkah selanjutnya adalah pembuatan tabel *current events* dengan cara menambahkan kode program pada file *modules/MMM-MyCalendar/MMM-mycalendar.js* seperti Tabel 4.12.

Tabel 4.12 Kode program *current events*

```

var eventWrapperOuter = document.createElement("tr");
//logic add current event

```

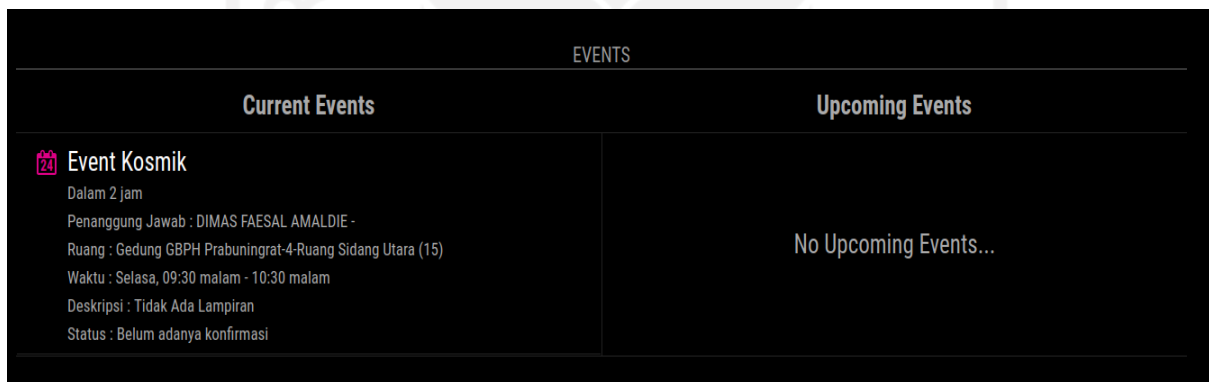


```

if(events[e].today) {
    eventWrapperOuter.appendChild(eventWrapper)
    wrapper.appendChild(eventWrapperOuter);
} else {
    //upcoming
    var tr_n = document.createElement("tr");
    var td_n = document.createElement("td");
    td_n.className = 'normal calendar-event with-symbol';
    td_n.innerHTML = eventWrapper.innerHTML;
    tr_n.appendChild(td_n);
    td_kanan.appendChild(tr_n)
}

```

Tabel *current event* merupakan hasil dari pemesanan ruangan yang dikhususkan hanya untuk hari ini saja oleh pengguna, sehingga pengguna dapat melihat perbedaan detail informasi yang ada. Terlampir pada Gambar 4.13 merupakan hasil antarmuka pengguna yang sudah berhasil di konfigurasi.



Gambar 4.13 Antarmuka pengguna Tabel *current events*

## 4.2 Pengujian Penelitian

Pada tahap pengujian akan dilakukan pengujian terhadap sistem yang sudah dibuat. Pengujian ini dilakukan untuk mengetahui apakah sistem dapat berfungsi sesuai dengan perancangan yang telah dilakukan. Pada tahap pengujian diharapkan dapat mengetahui kekurangan dan kelebihan dari sistem penjadwalan *Magic Mirror* menggunakan *Raspberry Pi*. Penulis menggunakan pengujian *Black box* dan pengujian UAT (*User Accpetance Test*).

#### 4.2.1 Black Box Testing

Pengujian *black box* ini dilakukan untuk menguji fungsionalitas sistem agar dapat berjalan sesuai keinginan. Pada penelitian ini dilakukan pengujian *black box* dengan menggunakan *software* postman. Pengujian *black box* pada penelitian ini menggunakan 3 variabel pengujian diantaranya sebagai berikut:

- a. Pengujian membuat *event* baru pada Google Calendar.
- b. Pengujian merubah data *event*.
- c. Pengujian menghapus data *event*.

Adapun hasil dari pengujian *black box* pada sistem dapat dilihat pada Tabel 4.13.

Tabel 4.13 Hasil *Black box Test*

NO	Skenarion pengujian	Hasil Pengujian	Kesimpulan
1.	Membuat sebuah <i>event</i> baru berdasarkan permintaan pengguna pada Google Calendar meliputi waktu, tanggal, nama ruangan, dan nama pengguna.	Sistem berhasil membuat <i>event</i> dan berhasil ditampilkan pada <i>Magic Mirror</i> .	Sesuai Harapan
2.	Melakukan perubahan pada <i>event</i> yang sudah dibuat sebelumnya.	Sistem dapat memperbarui data <i>event</i> .	Sesuai Harapan
3.	Menghapus sebuah <i>event</i> .	Sistem berhasil menghapus data <i>event</i> .	Sesuai Harapan

Berdasarkan hasil pengujian *black box* dengan kasus uji diatas, dapat ditarik kesimpulan bahwa sistem *Magic Mirror* dapat berjalan sesuai dengan yang diharapkan tanpa ada masalah.

#### 4.2.2 Pengujian UAT (*User Acceptance Test*)

Skala pengukuran variabel dalam penelitian ini mengacu pada Skala Likert (Likert Scale), yang masing-masing jawaban diberi skor (nilai) atau bobot yaitu banyaknya skor antara 1 sampai 5 sebagai berikut (Pambudi & Arini, 2018):

1. Jawaban SS (Sangat Setuju) diberi nilai 81% - 100%
2. Jawaban S (Setuju) diberi rentang nilai 61% - 80%

3. Jawaban R (Ragu - Ragu) diberi rentang nilai 41% – 60%
4. Jawaban TS (Tidak Setuju) diberi rentang nilai 21% - 40%
5. Jawaban STS (Sangat Tidak Setuju) diberi rentang nilai 0% – 20%

Pengujian UAT dilakukan kepada 10 orang pengguna dengan cara mengisi kuesioner. Adapun hasil kuesioner dari 10 pengguna dapat dilihat pada Tabel 4.15.

Tabel 4. 14 Tabel Nilai

PK	TS	KS	CS	S	SS
Nilai	1	2	3	4	5

Keterangan:

PK = Pertanyaan Kuesioner      CS =Cukup Setuju  
 TS = Tidak Setuju                      S = Setuju  
 KS = Kurang Setuju                    SS = Sangat Setuju

Tabel 4. 15 Hasil Kuesioner

No	Variabel	Penilaian					Persentase (%)
		1	2	3	4	5	
	Desain						
1	Tampilan aplikasi <i>Magic Mirror</i> ini menarik.			1	3	6	90%
2	Warna tulisan dengan layar belakang ( <i>background</i> ) sudah sesuai.				4	6	92%
3	Penggunaan tulisan ( <i>font</i> ) mudah dibaca			2	4	4	84%
	Layanan						
4	Aplikasi ini membantu dalam proses penjadwalan <i>meeting</i>			1	5	4	86%
5	Informasi yang disampaikan sudah lengkap.				5	5	90%

6	Aplikasi sudah sesuai dengan kebutuhan.			2	3	5	86%
<i>Efisiensi</i>							
7	Penyampaian informasi mudah dimengerti dan dipahami.			1	3	6	90%
8	Sistem ini sudah efisien bagi pengguna			1	4	5	88%
Rata-Rata Persentase Skor							88.2%

Berdasarkan dari hasil analisis dan rekapitulasi kuesioner dapat disimpulkan bahwa rata-rata skor persentase dari 10 responden yaitu 88.2%, di mana dalam variable skala likert jawaban Sangat Setuju yang diberi rentang nilai 81% - 100% yang artinya sistem penjadwalan *Magic Mirror* menggunakan *Raspberry Pi* yang telah dibangun ini sudah sangat baik dan dapat memudahkan pengguna dalam mendapatkan informasi *event* penjadwalan.

### 4.3 Analisis penelitian

Pada tahap analisis penelitian yang dilakukan terhadap kelebihan dan kekurangan oleh sistem yang sudah dibuat setelah dilakukannya tahapan pengujian. Kekurangan dan kelebihan dari penelitian ini akan menjadi pertimbangan untuk penulis dan pembaca dalam menilai keberhasilan pada penelitian ini. Kekurangan dalam penelitian ini diharapkan dapat menjadi acuan untuk pengembangan pada penelitian selanjutnya agar apa yang sudah dicapai dapat dikembangkan lebih jauh lagi. Berikut adalah kelebihan dan kekurangan dari hasil penelitian:

#### 4.3.1 Kelebihan Sistem

- Sistem dapat terhubung kapan pun dan di mana pun melalui jaringan internet.
- Sistem dapat dijalankan pada perangkat *Raspberry Pi* yang hemat listrik.
- Sistem dapat menampilkan data pemesanan ruangan secara *real time* bagi pengguna.
- Antarmuka yang ditampilkan oleh sistem kepada pengguna mudah dimengerti.
- Sistem dapat secara otomatis menyala pada saat mendapat tegangan listrik.

#### 4.3.2 Kekurangan Sistem

- Untuk melakukan pemesanan ruangan masih menggunakan mode manual dengan meminta pembuatan *event* baru kepada *admin*.

- b. Sistem belum dapat menampilkan ruangan yang tersedia.
- c. Antarmuka yang ditampilkan kepada pengguna masih sangat sederhana.



## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Berdasarkan hasil perancangan dan implementasi serta pengujian yang sudah dilakukan pada penelitian tugas akhir ini, adapun beberapa kesimpulan antara lain adalah sebagai berikut:

- a. Sistem Penjadwalan *Magic Mirror* Menggunakan *Raspberry Pi* dapat mempermudah pengguna di lingkungan kampus Universitas Islam Indonesia dalam melakukan penjadwalan ruangan.
- b. Sistem Penjadwalan *Magic Mirror* Menggunakan *Raspberry Pi* dapat diletakkan pada setiap ruangan yang dibutuhkan sehingga pengguna dapat mengakses informasi terkait penjadwalan.

#### 5.2 Saran

Sistem ini masih memiliki beberapa kekurangan. Maka dari itu, diharapkan pada penelitian selanjutnya agar dapat melakukan pengembangan atas saran yang diberikan pada penelitian ini. Adapun beberapa saran antara lain adalah sebagai berikut:

- a. Menggunakan perangkat *Raspberry Pi* terbaru agar meningkatkan performa pada sistem.
- b. Penulis menyarankan agar sistem ini dapat dikembangkan lagi dengan tampilan yang lebih menarik dan responsif.
- c. Penulis berharap kedepannya sistem mampu menampilkan ruangan yang tersedia.

**DAFTAR PUSTAKA**

- Assagaf, A., Ibrahim, A., & Suranto, C. (2018). Membangun Sistem Informasi Penjadwalan Dengan Metode Algoritma Genetika Pada Laboratorium Teknik Informatika Universitas Muhammadiyah Maluku Utara. *Jurnal Ilmiah Komputer dan Informatika* , 95-105.
- Burange, A. W., & Misalkar, H. (2015). Review of Internet of Things in Development of Smart Cities with Data Management & Privacy. *2015 International Conference on Advances in Computer Engineering and Applications* (pp. 189-195). IEEE.
- Nugraha, R. R. (2018). *Pembangunan Smart Mirror Menggunakan Raspberry Pi Berbasis IoT*. Bandung: UNIKOM.
- Pinedo, M. L. (2012). *Scheduling, Theory, Algorithms, and System 4th Edition*. New Jersey: Springer.
- Qorni, W. A. (2018). *Rancang Bangun Smart Home Menggunakan Rapsberry PI 3 Dengan Control Berbasis Web*. Jakarta: UIN Syarif Hidayatullah.
- Ramadhan, D. A. (2019). *Pengembangan Sistem Informasi Meeting Berbasis Web Menggunakan Framework CodeIgniter*. Bogor: Institut Pertanian Bogor.
- Ramli, M., Mamahit, D. J., & Wuwung, J. O. (2018). Rancang Bangun Sistem Pemantau Tamu Pada. *Smart Home Berbasis Raspberry Pi 3* , 1-8.
- Suyadi. (2016). *Aplikasi Kalender Untuk Mengelola Jadwal Kegiatan Di UMS Menggunakan Google Application API*. Surakarta: Universitas Muhammadiyah Surakarta.
- Yuwono, B., Nugroho, S. P., & Heriyanto. (2015). Pengembangan Model Public Monitoring Sistem Menggunakan Rapsberry PI. *Telematika* , 123-133.

LAMPIRAN

