PEMBANGUNAN *SMART MIRROR* MENGGUNAKAN RASPBERRY PI BERBASIS IOT

Rasmandika Ramdhan Nugraha[1], Iskandar Ikbal[2]

[1,2] Teknik Informatika – Universitas Komputer Indonesia
Jalan Dipatiukur No. 112-114 Bandung, Jawa Barat 40132
Email : rasmandika95@gmail.com, iskandar.ikbal@email.unikom.ac.id

ABSTRAK

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 artist* khususnya dalam pengingat jadwal dan data *client*. Sistem ini diharapkan dapat terus dikembangkan dan dapat membantu tidak hanya pada bagian *makeup artist* saja.

Kata Kunci : *Internet of Thing*, Aplikasi, Android, Bandung Lautan Photographer, *Makeup Artist*

1. PENDAHULUAN

Bandung Lautan Photographer merupakan sebuah komunitas yang bergerak dibidang photographer. Didalam komunitas Bandung Lautan Photographer terdapat tiga bagian yang berperan penting yaitu photographer, model dan *makeup artist*. Setiap tahunnya komunitas Bandung Lautan Photographer mengadakan sebuah event yang bertema photography. Event tersebut merupakan sebuah ajang untuk membangun sebuah relasi baik pertemanan ataupun relasi kerja antar photographer, model maupun *makeup artis*. Selain itu *event* bandung lautan photographer merupakan ajang promosi untuk menambah *client* dan meningkatkan popularitas kerja.

Berdasarkan hasil wawancara dengan Zihan Nurul A selaku koordinator bagian *makeup artist* Bandung Lautan Photograper (BLP), hampir semua *makeup artist* di BLP memiliki lebih dari satu pekerjaan yaitu sebagai seorang *makeup artist* dan pekerjaan lain seperti mengajar, wirausaha dan lain sebagainya. Oleh karena itu kebanyakan makeup artist memiliki jadwal yang tidak menentu setiap harinya. Jadwal yang tak menentu tersebut terkadang membuat para *makeup artist* memiliki masalah dalam pengaturan jadwal. Baik itu jadwal kegiatan diluar luang lingkup *makeup artist* maupun jadwal bertemu *client*. Selain dari permasalahan jadwal, permasalahan lain yang sering terjadi adalah ketika banyak *client* yang membuat janji pertemuan pada hari tertentu terkadang membuat seorang *makeup artist* salah saat mengkategorikan *client* yang akan dirias. Hal ini berpengaruh pada peralatan yang akan mereka bawa dan gunakan untuk merias.

Meskipun telah mencatat pada perangkat *mobile*, permasalahan diatas tetap terjadi. Hal ini disebabkan karena tidak semua *makeup artist* di Bandung Lautan Photographer suka membuka perangkat *mobile* saat sedang bersiap melakukan aktifitas atau saat sedang merias. Baik merias orang lain maupun merias dirinya sendiri karena saat membuka perangkat *mobile* fokus jadi terganggu oleh konten yang ada di perangkat *mobile* mereka.

Berdasarkan permasalah diatas maka dibutuhkan suatu media yang dapat menampilkan setiap jadwal ataupun data *client* seorang *makeup artist* dan dapat dilihat saat sedang bersiap melakukan aktifitas atau merias tanpa harus selalu membuka perangkat *mobile*.

2. ISI PENELITIAN

2.1 Landasan Teori

2.1.1 Aplikasi

Aplikasi merupakan sebuah perangkat lunak yang digunakan pemakan komputer untuk membantu menyelesaikan pekerjaannya. Dibutuhkan suatu Bahasa pemrograman untuk yang dapat

berbentuk *assembler, compiler* ataupun *interpreter* jika ingin mengembangkan aplikasi sendiri [1].

Bahasa pemrograman berfungsi untuk menulis program dengan Bahasa yang lebih mudah, setelah itu baru dirubah kedalam Bahasa yang dapat dimengerti oleh komputer. Dibutuhkan perancangan dan prosedur yang baik untuk mengembangkan sebuah aplikasi untuk memecahkan permasalahan yang rumit [1].

2.1.2 Android

Berikut merupakan pembahasan mengenai pengertian android, sejarah, versi :

1. Pengertian Android

Android merupakan sistem operasi pada handphone yang terbuka dan berbasis pada sistem operasi *linux*. Pada awalnya *Android Inc*. yang merupakan pendatang baru yang membuat piranti lunak untuk ponsel dibeli oleh *Google Inc*. Kemudian dibentuklah *Open Handset Alliancei*, konsorsium dari perusahaan peranti keras, peranti lunak dan telekomunikasi yang bertujuan untuk mengembangkan *Android* [2].

2. Sejarah Android

Pada awalnya terdapat berbagai macam sistem operasi pada perangkat selular diantaranya yaitu sistem operasi *Symbian, Microsoft Windows Mobile, Mobile Linux, Iphone* dan lainnya. Namun diantara sistem operasi yang ada belum mendukung standar dan penerbitan API yang dapat dimanfaatkan secara keseluruhan dan dengan biaya yang murah. Kemudian Google ikut berkecimpung didalamnya dengan platform Android, yang menjanjikan keterbukaan, keterjangkauan, open source dan framework berkualitas.

Pada tahun 2005, Google mengakuisisi perusahaan Android Inc. untuk mamulai perkembangan platform Android. Dimana terlibat pengembangan ini adalah Andy Rubin, Rich Miner, Nick Sears, dan Chris White. Pada pertengahan 2007 sekelompok pemimpin industri bersama-sama membentuk analisis aliansi perangkat selular terbuka, *Open Handset Alliance* (OHA). Bagian dari tujuan aliansi ini adalah berinovasi dengan cepat dan menaggapi kebutuhan konsumen dengan lebih baik, dengan produk awalnya adalah platform Android. Dimana Android dirancang untuk melayani kebutuhan operator telekomunikasi, *manufaktur handset*, dan pengembangan aplikasi, OHA berkomitmen untuk membuat Android *open source* dengan lisensi *Apache* Versi 2.0 [2].

2.1.3 Java Script

JavaScript adalah bahasa *script* yang berdasar pada objek yang memperbolehkan pemakai untuk mengendalikan banyak aspek interaksi pemakai pada suatu dokumen HTML. Dimana objek tersebut dapat berupa suatu window, frame, URL, dokumen, form, button, atau item yang lain. Yang semuanya itu mempunyai properti yang saling berhubungan dengannya dan masing-masing memiliki nama, lokasi, warna nilai dan atribut lain [3].

2.1.4 *Internet of Thing*

Internet of Thing merupakan sebuah konsep yang memanfaatkan konektifitas yang tersambung secara terus menerus untuk memiliki berbagai kemampuan seperti data *remote control*, yang semuanya tersambung ke jaringan local maupun global [4].

2.1.5 Raspberry Pi

Rasberry pi adalah rangkaian elektronika berupa komputer papan tunggal yang berukuran seperti kartu kredit dan berfungsi layaknya komputer lengkap menggunakan *system on a chip* ARM yang diintegrasikan di atas PCB. Rapberry pi dapat terhubung dengan layar monitor melalui konektor yang sudah tersedia seperti HDMI atau dapat menggunakan Connector Converter seperti HDMI to VGA. *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* dan banyak lainnya, melalui saluran GPIO (General-purpose input/output) via antarmuka I²C (Inter-Integrated Circuit) [5].

2.1.6 Sensor Sidik Jari

Sensor sidik jari sudah banyak digunakan mulai dari *attendance system* (Sistem Absensi / Kehadiran), *access control* (Sistem pengontrol) hingga sebagai identitas pribadi. Meskipun dalam cara kerja system sidik jari sangat rumit, namun dalam beberapa detik dapat diketahui apakah cocok dengan pengguna atau tidak [6].

2.1.7 *Web Service*

Web service merupakan sekumpulan dari data (*database*), perangkat lunak, atau bagian dari sebuah perangkat lunak yang dapat diakses oleh berbagai perangkat secara *remote* dengan sebuah perantara tertentu [7]. Secara umum,web service dapat diidentifikasikan dengan menggunakan URL seperti hanya web pada umumnya. Namun yang membedakan web service dengan web pada umumnya adalah interaksi yang diberikan oleh web service. Berbeda dengan URL web pada umumnya,

URL web service hanya menggandung kumpulan informasi, perintah, konfigurasi atau sintaks yang berguna membangun sebuah fungsi-fungsi tertentu dari aplikasi [8].

2.2 Analisis dan Perancangan

2.2.1 Analisis Masalah

Pembangunan smart mirror menggunakan raspberry pi dan android merupakan suatu sistem yang dirancang dan digunakan untuk melihat jadwal seorang pengguna. Selain jadwal pengguna juga dapat melihat data client dan catatan sehingga pengguna tidak perlu selalu membuka *smartphone* saat hendak atau siap-siap dalam bekerja. Adapun berdasarkan penelitian yang dilakukan terdapat permasalahan yang terjadi sehingga dibuat sistem ini adalah :

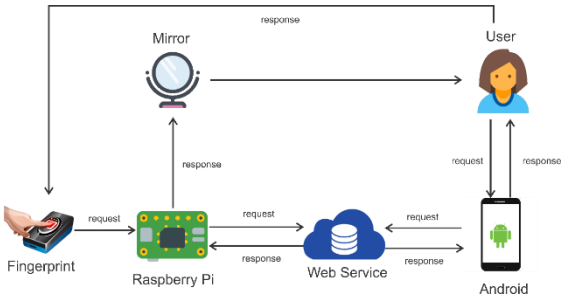
1. Memiliki lebih dari satu pekerjaan membuat jadwal tak menentu yang menjadikan seorang *makeup artis* di Bandung Lautan Photographer terkadang mengalami permasalahan dalam pengaturan jadwal khususnya jadwal merias seorang *client*.
2. Ketika banyak client yang membuat janji pertemuan di hari tertentu terkadang membuat seorang *makeup artist* salah dalam mengkategorikan client. Hal ini berakibat pada barang yang akan dipersiapkan untuk proses merias.

Oleh karena itu dibutuhkan solusi untuk menangani permasalahan yang sedang terjadi yaitu suatu sistem smart mirror yang bertujuan untuk :

1. Sistem diharapkan dapat mempermudah *makeup artist* dalam melihat jadwal maupun data *client* khususnya saat bersiap melakukan aktifitas atau saat sedang merias.
2. Sistem diharapkan dapat mempermudah makeup artist dalam melihat kategori *client* untuk mempersiapkan peralatan yang dibutuhkan saat merias.
3.

2.2.2 Analisis Arsitektur Perancangan Sistem

Analisis Arsitektur Perancangan Sistem merupakan sebuah perancangan dari sistem yang akan dibangun. Sistem akan berupa aplikasi android dan juga aplikasi pada perangkat Raspberry Pi. Aplikasi android akan digunakan untuk menyimpan data yang dibutuhkan kedalam firebase dalam bentuk JSON. Kemudian data akan diproses didalam perangkat Raspberry Pi dan akan ditampilkan jika proses login berhasil. Adapun rancangan sistem yang akan dibangun dapat dilihat pada gambar berikut :



Gambar 1. Arsitektur Perancangan Sistem

Berikut penjelasan mengenai arsitektur perancangan sistem *smart mirror* dan aplikasi *smartphone* android :

1. Pengguna dapat memasukan data jadwal, data *client* dan catatan pada perangkat *mobile*. Selain itu pengguna juga dapat mengelola data-data yang sudah dimasukan pada perangkat *mobile*.
2. Data akan dikirim kedalam *web service* dan disimpan didalam firebase. Firebase sendiri merupakan database yang di-*host* di cloud. Data disimpan sebagai JSON dan disinkronisasi secara *realtime* ke setiap klien yang terhubung [9].
3. Selain dari perangkat android, pengguna dapat melihat data yang sebelumnya telah dimasukan, pada perangkat cermin.
4. Sebelum perangkat cermin menampilkan data, pengguna diharuskan untuk melakukan *login* menggunakan sidik jari.
5. Perangkat akan mengecek status login dari pengguna berdasarkan sidik jari yang telah terdaftar.
6. Jika proses *login* berhasil, maka data akan diambil oleh perangkat cermin dan akan ditampilkan kepada pengguna.

2.2.3 Analisis Teknologi yang Digunakan

Analisis ini bertujuan untuk mengetahui teknologi apa saja yang digunakan dalam penelitian ini. Adapun teknologi yang digunakan dalam penelitian ini adalah sebagai berikut :

1. Firebase

Firebase merupakan database yang di-host di cloud. Data disimpan sebagai JSON dan disinkronisasi secara *realtime* ke setiap klien yang terhubung [9]. Firebase digunakan sebagain *web service* untuk menyimpan data. Data akan dikoneksikan antara perangkat *mobile* dan perangkat *smart mirror*.

2. Sensor Sidik Jari

Sensor sidik jari digunakan sebagai alat untuk melakukan autentikasi pada perangkat cermin. Sensor ini akan mengecek sidikjari berdasarkan pengguna yang terdaftar pada perangkat *mobile* dan

sudah melakukan penyimpanan data sidik jari pada perangkat cermin.

2.2.4 Analisis Kebutuhan Non Fungsional

1. Analisis Kebutuhan Perangkat Keras

Analisis perangkat keras dimaksudkan untuk mengetahui jenis perangkat keras serta spesifikasi perangkat keras yang akan digunakan untuk membangun sistem smart mirror. Adapun analisis kebutuhan perangkat keras diantaranya:

1. Perangkat membutuhkan koneksi internet, dalam hal ini perangkat smear mirror akan menggunakan *wifi* sebagai konektifitas.
2. Cermin memiliki bentuk persegi panjang.
3. Cermin memiliki ukuran 40 x 70 cm dan ketebalan bingkai yang disesuaikan.
4. Bagian atas cermin akan digunakan sebagai *output* dari data yang ada dan bagian bawah akan digunakan murni sebagai cermin.
5. Perangkat dapat menampilkan jadwal, data *client* dan catatan.
6. Perangkat akan menggunakan *fingerprint* sensor sebagai keamanan untuk data pada cermin.
7. Perangkat dapat bekerja *realtime* selama 24 jam.
8. Perangkat dapat secara otomatis menyala saat mendapat tegangan listrik.

2. Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak merupakan proses analisis yang lebih menekankan kepada aspek pemanfaatan perangkat lunak. Dimana perangkat lunak yang digunakan dapat mendukung atau tidaknya sistem yang akan dibangun. Adapun analisis perangkat lunak pada smart mirror diantaranya:

1. Sistem operasi *mobile* harus memenuhi standar minimum yaitu Android 4.4 Kitkat
2. Sistem harus mampu menjalankan *fingerprint sensor module*.
3. Sistem harus mampu menjalankan *web service*.
4. Sistem harus mampu menampilkan jadwal, data *client* dan catatan.

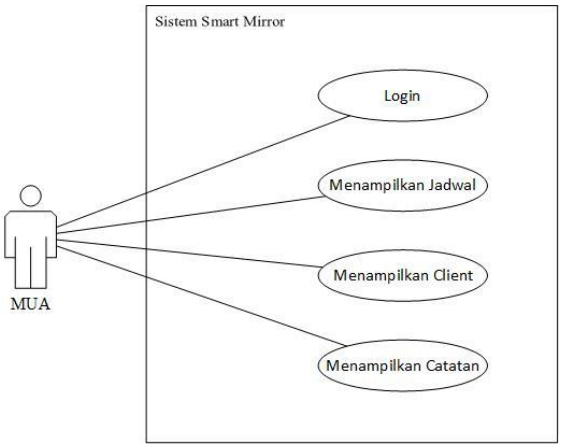
3. Analisis Kebutuhan Fungsional

3.1 Use Case Diagram

Use case adalah situasi dimana sebuah sistem digunakan untuk memenuhi satu atau lebih kebutuhan pengguna. Use case digunakan untuk menggambarkan fungsionalitas pada bagian yang disediakan oleh sistem. Use case menjelaskan apa yang harus dilakukan oleh sistem tetapi tidak menjelaskan apa yang seharusnya sistem tidak lakukan [10].

1. Use Case *Smart Mirror*

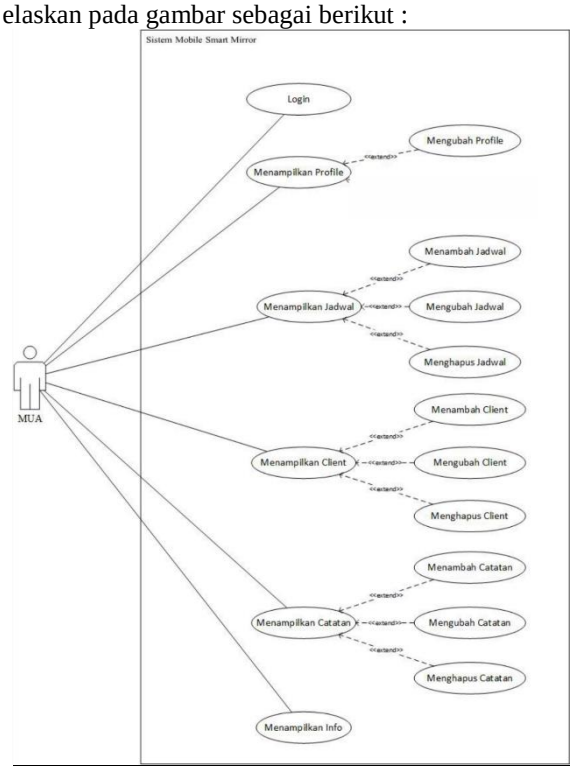
Use case diagram perangkat *smart mirror* dapat dilihat pada gambar berikut:



Gambar 2. Use Case Smart Mirror

2. Use Case Aplikasi Mobile

Use case diagram aplikasi *mobile* dijelaskan pada gambar sebagai berikut :



Gambar 3. Use Case Aplikasi Mobile

3.2 Implementasi dan Pengujian

3.2.1 Implementasi Perangkat Keras

Spesifikasi perangkat keras yang digunakan dalam pembuatan *smart mirror* dapat dilihat pada tabel 1

Table 1. Spesifikasi Perangkat Keras

NO	Perangkat Keras	Spesifikasi
1	Prosessor	Interl Core i5 1.8 GHz
2	RAM	4 GB
3	Hardisk	1 TB
4	VGA	NVIDIA GeForce 930 MX
5	Keyboard	Standart
6	Mouse	Standart
7	Mini Pc	Raspberry Pi

3.2.2 Implementasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam pembuatan *smart mirror* dapat dilihat pada tabel 2

Table 2. Spesifikasi Perangkat Lunak

NO	Perangkat Lunak	Spesifikasi
1	Sistem Operasi	Windows 10
2	Bahasa Pemrograman	Java Android & Javascript
3	Text Editor	Android Studio & Sublime Text
4	Framework	NodeJs

3.2.3 Implementasi Kelas

Implementasi *class* pada sistem *smart mirror* merupakan implementasi dari perancangan class diagram kedalam bentuk file yang akan digunakan dalam pembuatan dan jalannya sistem. Berikut merupakan beberapa *class* yang digunakan pada aplikasi mobile sebagai berikut:

Table 3. Implementasi Kelas

No	Nama *Class*	Nama File
1	MainActivity	MainActivity.java
2	ProfileFragment	ProfileFragment.java
3	EditProfileActivity	EditProfileActivity.java
4	Profile	Profile.java
5	AboutFragment	AboutFragment.java
6	HomeFragment	HomeFragment.java
7	Schedule	Schedule.java
8	ScheduleListActivity	ScheduleActivity.java
9	ScheduleListFragment	ScheduleFragment.java
10	AddScheduleActivity	AddScheduleActivity.java
11	ScheduleDetailActivity	ScheduleDetailActivity.java
12	Client	Client.java
13	ClientListActivity	ClientActivity.java
14	ClientListFragment	ClientFragment.java
15	AddClientActivity	AddClientActivity.java
16	ClientDetailActivity	ClientDetailActivity.java
17	NoteListActivity	NoteListActivity.java
18	NoteAddActivity	NoteAddActivity.java
29	Note	Note.java

3.2.4 Implementasi Antarmuka

Implementasi antarmuka merupakan implementasi yang dilakukan pada perancangan antarmuka atau *mockup* kedalam bentuk file. Pada implementasi antarmuka aplikasi *mobile*, perancangan diubah kedalam file .xml. adapun implementasi antarmuka aplikasi *mobile* adalah sebagai berikut:

Table 4. Implementasi Antarmuka

No	Nama Antarmuka	Nama File
1	Halaman Login	activity_login.xml
2	Halaman Home	fragment_home.xml
3	Halaman Profile	fragment_profile.xml
4	Halaman Edit Profile	activity_edit_profile.xml
5	Halaman Info	fragment_about.xml
6	Halaman Jadwal	fragment_schedule_list.xml
7	Halaman Tambah Jadwal	activity_schedule_add.xml
8	Halaman Detail Jadwal	activity_schedule_detail.xml
9	Halaman Client	fragment_client _list.xml
10	Halaman Tambah Client	activity_client _add.xml
11	Halaman Detail Client	activity_client _detail.xml
12	Halaman Catatan	activity_note_list.xml
13	Halaman Tambah Catatan	activity_note_add.xml

3.2.5 Implementasi Raspberry Pi

Pada penelitian ini Raspberry Pi yang digunakan adalah Raspberry Pi 3 tipe B. Raspberry Pi sendiri dipasang pada bagian belakang perangkat cermin. Adapun spesifikasi Raspberry Pi 3 tipe B adalah sebagai berikut :

Table 5. Implenentasi Raspberry Pi

Komponen	Spesifikasi
SoC	Broadcom BCM2837
CPU	4× ARM Cortex-A53, 1.2GHz
GPU	Broadcom VideoCore IV
RAM	1GB LPDDR2 (900 MHz)
Networking	10/100 Ethernet, 2.4GHz 802.11n
Bluetooth	Bluetooth 4.1 Classic
Storage	microSD
GPIO	40-pin header, populated
Ports	HDMI, 3.5mm, 4× USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI).
Power Sources	5V, 2A

3.2.6 Implementasi Monitor

Monitor yang digunakan pada penelitian ini merupakan monitor LCD 14” LG tipe W1642S. Pada pembangunan *smart mirror* ini, monitor dibongkar dan dilepas bagian *casing* luar. Hal ini dimaksudkan agar monitor dapat menempel dengen kaca yang akan dibuat. Adapun contoh monitor yang sudah dibuka bagian *casing*nya dapat dilihat pada gambar .



Gambar 4. Implementasi Monitor

Pada implementasi ini juga akan dijelaskan mengenai pemasangan monitor. Monitor sendiri dipasang pada bagian atas cemin. Monitor yang sudah dibuka bagian *casing* dirapatkan dengan cermin agar gambar pada monitor dapat terlihat.

3.2.7 Implementasi Cermin

Cermin merupakan bagian yang paling utama dalam *smart mirror*. Pada penelitian ini kaca dibuat menyesuaikan dengan ukuran monitor LCD adapun spesifikasi kaca yang digunakan dapat dilihat pada tabel

Table 6. Implementasi Cermin

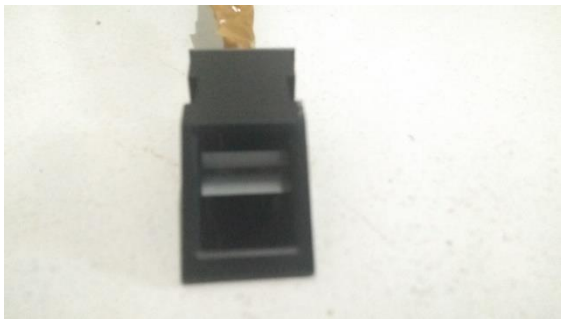
Jenis	Spesifikasi
Ukuran	38 x 70 cm
Ketebalan	2 mm
Lapisan	Kaca Film *One Way* 80%

3.2.8 Implementasi Sensor Sidik Jari

Pada penelitian ini digunakan sensor sidik jari sebagai alat untuk melakukan autentikasi pada perangkat cermin. Adapun sensor sidik jari yang digunakan adalah tipe FPM 10A. berikut merupakan spesifikasi dari sensor sidikari FPM 10A:

Table 7. Implementasi Sensor Sidik Jari

Spesifikasi	Keterangan
Supply	DC 3V – 5V
Waktu citra	< 1,6 detik
Ukuran	14 x 18 mm
File	256 byte
File template	512 byte
Kapasitas Penyimpanan	1000
Antarmuka PC	UART



Gambar 5. Sensor Sidik Jari FPM 10A

3.2.9 Pengujian Sensor Sidik Jari

Pengujian ini dimaksudkan untuk mengetahui apakah modul sensor sidik jari FPM 10A yang terpasang pada sistem dapat berfungsi dengan baik atau tidak. Pengujian dilakukan dengan memeriksa beberapa sidikjari secara terus menerus. Setelah itu akan diambil data keakuratan nya dengan melihat persentase keberhasilan dan seberapa besar persentase kegagalan yang terjadi.

Dari hasil pengujian yang dilakukan menggunakan 5 sidik jari dengan 50 kali pengujian untuk menentukan bagian jari yang lebih baik digunakan pada proses authentikasi, maka didapat berdasarkan tabel berikut.

Table 8. Pengujian Sensor Sidik Jari

Sidik Jari	Total	Benar	Error
Jempol	50	22	28
Telunjuk	50	49	1
Jari Tengah	50	48	2
Jari Manis	50	42	8
Kelingking	50	39	11

3.2.10 Pengujian Jarak Pemakaian Cermin

Pengujian ini dimaksudkan untuk melihat seberapa jauh darak yang direkomendasikan untuk menggunakan cermn. Jarak yang terlalu jauh akan berakibat pada kurangnya penglihatan terhadap data yang ditampilkan pada cermin. Oleh karena itu dilakukan pengujian jarak yang sesuai agar pengguna dapat menyesuaikan pada saat menggunakan cermin. Adapun hasil dari pengujian jarak terdekat dan terjauh yang direkomendasikan adalah sebagai berikut :

Table 9. Pengujian Jarak Pemakaian Cermin

Keterangan	Jarak
Jarak Terdekat	50 cm
Jarak Terjauh	150

3.2.11 Pengujian Sistem Notifikasi

Pengujian sistem notifikasi bertujuan untuk melihat apakah notifikasi yang ada pada perangkat *smart mirror* dapat berjalan dengan baik atau tidak. Notifikasi sendiri dijalankan pada 5 menit, 15 menit, 45 menit, dan 1 jam sebelum waktu pada jadwal yang telah diinputkan dimulai. Notifikasi dimunculkan berupa *alert* dan suara pada perangkat cermin. Adapun hasil pengujian adalah sebagai berikut:

Table 10. Pengujian Sistem Notifikasi

Percobaan	Waktu diinputkan	Waktu menyala notifikasi	kondisi
5 menit	09.55	09.50	Suara menyala
15 menit	09.55	09.40	Suara menyala
45 menit	09.55	09.10	Suara menyala
1 jam	09.55	08.55	Suara menyala

3. PENUTUP

3.1 Kesimpulan

Berdasarkan hasil yang telah didapat dari penelitian yang dilakukan dalam penyusunan tugas akhir dan mengacu pada tujuan penelitian, maka dapat disimpulkan sebagai berikut:

1. Smart mirror cukup membantu sebagai mengingatkan serta untuk melihat jadwal saat sedang makeover.
2. Smart mirror cukup membantu dalam mencatat data *client* dan mengurangi kesalahan dalam menyiapkan atau membawa barang yang diperlukan untuk proses *makeover client*.

3.2 Saran

Sistem ini masih jauh dari kata sempurna dan masih banyak kekurangan, oleh karena itu sistem ini masih perlu dikembangkan lagi dan lebih menekankan pada fungsi *smart*, tidak hanya untuk melihat jadwal, data *client*, catatan serta *reminder*. Seperti dapat menambahkan sesuatu yang membantu bagian *makeup artist* dalam proses *makeover client*. Selain itu dalam penambahan pengguna untuk proses *login* menggunakan sidik jari agar dapat dibuat lebih dinamis.

DAFTAR PUSTAKA

[1] J. Hartono, Pengenalan Komputer, Yogyakarta: Andi, 2004.

[2] N. Safaat H, Pemograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android, Bandung: Informatika, 2012 (Edisi Revisi).

[3] Suryana, Taryana dan Koesheryatin (2014). Aplikasi Internet Menggunakan HTML, CSS & JavaScript. Jakarta: Penerbit PT Elex Media Komputindo.

[4] Iwan, Acep. (2017). *RANCANG BANGUN SISTEM PENGAIRAN DAN PEMANTAUAN UNTUK PRODUKSI TOUGE BERBASIS INTERNET OF THING (STUDI KASUS BLOK*

TOUGE KOTA CIMAHI). Bandung. UNIKOM.
[5] Kurniawan, Rio. (2016). *VIDEO CONVERENCE UNTUK E-LEARNING PADA MINI PC RASPBERRY*. Bandung. UNIKOM.
[6] Lukman, Arief. (2017). *SISTEM ABSENSI BERBASIS MIKROKONTROLLER MENGGUNAKAN SENSOR SIDIK JARI DI SMAN 1 CIMAHI*. Bandung. UNIKOM.
[7] R. Fauziah, "Pengertian dan Cara Kerja RPC, RMI, dan WebService," 27 November 2015. [Online]. Available: http://si283.ilearning.me/2015/11/27/pengertian-dan-cara-kerja-rpc-rmi-dan-webservice/. [Diakses 12 Juli 2017].
[8] Sampurna, Jaya. (2017). *PEMBANGUNAN SISTEM PEMANTAUAN RUMAH WALET BERBASIS IOT*. Bandung. UNIKOM.
[9] "Panduan Firebase Realtime Database" [Online]. Available: https://firebase.google.com/docs/database/
[10] R. Miles and K. Hamilton, Learning UML 2.0, O'Reilly Media, 2006.