

PERANCANGAN DAN IMPLEMENTASI APLIKASI *WEBSITE* UNTUK PEMANTAUAN *SMART BATHROOM* BERBASIS IOT

Design and Implementation Website Application for Smart Bathroom Monitoring Based on IOT

Christhoper Helmi Rudijanto ¹, Denny Darlis, S.Si., M.T ², Aris Hartaman, S.T., M.T ³
^{1,2,3}Prodi D3 Teknologi Telekomunikasi, Universitas Telkom
[1 yunusimsal@gmail.com](mailto:yunusimsal@gmail.com), [2 asepm267@gmail.com](mailto:asepm267@gmail.com), [3 atiknov@gmail.com](mailto:atiknov@gmail.com)

Abstrak

Kamar mandi merupakan salah satu bagian penting yang harus tersedia di setiap rumah. Pada umumnya ada 3 (tiga) komponen utama dalam kamar mandi yaitu air, sabun/sampo dan lampu sebagai pencahayaan. Namun pada saat ini kebanyakan masyarakat masih harus memantau kondisi air, sabun/sampo yang tersedia dan juga menyalakan lampu dengan menekan sakelar secara manual. Oleh karena itu, pemantauan kondisi kamar mandi secara otomatis dan *realtime* merupakan salah satu langkah yang praktis untuk mempermudah masyarakat dalam memantau kondisi kamar mandi pribadinya.

Pada proyek akhir ini penulis membuat alat pemantau kamar mandi jarak jauh yang berbasis *website*. *Website* tersebut nantinya akan diintegrasikan dengan *hardware* (mikrokontroler) yang terhubung dengan jaringan internet secara *realtime* menggunakan Google Firebase untuk mengirimkan data. Data yang diterima dari *hardware* kemudian akan diolah di *database* lalu akan ditampilkan di halaman *website*. Kemudian data tersebut akan ditampilkan pada sisi pengguna yang akan memberikan informasi tentang volume air (liter), volume sampo (liter), kondisi lampu dan *exhaust fan* (on/off). Selain itu, *website* juga akan memberikan notifikasi melalui *email* apabila terjadi perubahan kondisi.

Dari hasil pengujian penulis mengenai fungsionalitas dan sistem kerja *website* didapatkan hasil dari *website* sudah berjalan dengan baik dan sesuai dengan rancangan awal. Dengan adanya *website* ini pengguna dapat memantau kondisi kamar mandi untuk mengetahui berapa banyak penggunaan air dan sampo secara berkala harian dan bulanan. Hasil rata-rata *delay* yang didapatkan sebesar 0.1 detik. Berdasarkan hasil pengujian yang didapat pada pengujian tersebut dapat disimpulkan bahwa *website* berhasil di implementasikan dan diintegrasikan dengan *hardware smart bathroom*.

kata kunci : *smart bathroom*, mikrokontroler, *website*, *realtime*, Google Firebase, *hardware*, *delay*

Abstract

The bathroom is one of the important parts that must be available in every home. In general there are 3 (three) main components in the bathroom, namely water, soap / shampoo and lights as lighting. But at this time most people still have to monitor the condition of water, available soap / shampoo and also turn on the lights by pressing the switch manually. Therefore, monitoring the condition of the bathroom automatically and realtime is one of the practical steps to make it easier for the public to monitor the condition of their private bathroom.

In this final project the author makes a website-based remote monitoring device. The website will be integrated with hardware (microcontroller) that is connected to the internet network in realtime using Google Firebase to send data. Data received from hardware will then be processed in the database and will be displayed on the website page. Then the data will be displayed on the user side which will provide information about the volume of water (liters), volume of shampoo (liters), condition of lights and exhaust fan (on / off). In addition, the website will also provide notifications via email in the event of changing conditions.

From the results of the author's testing of the functionality and work system of the website the results of the website have been running well and in accordance with the initial design. With this website, users can monitor the condition of the bathroom to find out how much water and shampoo usage is regularly daily and monthly. The average result of delay obtained is 0.1 seconds. Based on the test results obtained on the test it can be concluded that the website was successfully implemented and integrated with smart bathroom hardware.

Keywords : *smart bathroom*, microcontroller, *website*, *realtime*, Google Firebase, *hardware*, *delay*

1. Pendahuluan

Kamar mandi adalah sebuah ruangan dimana seseorang dapat membersihkan tubuhnya melalui proses mandi. Sedangkan, pada umumnya kamar mandi dibedakan menjadi dua jenis, yaitu kamar mandi basah dan kamar mandi kering. Kamar mandi basah biasanya mempunyai tempat penampungan air (bak). Jenis kamar mandi seperti ini umum digunakan di Indonesia. Karena aliran air tidak terjamin, atau kurangnya aliran air pada suatu tempat, sehingga dibutuhkan suatu bak yang berisi air yang dapat dimanfaatkan setiap waktu. Sedangkan kamar mandi kering biasanya disediakan sebuah tempat khusus untuk orang mandi dengan menggunakan pancuran (*shower*) atau dengan setengah berbaring pada sebuah bak mandi (*bathtube*). Kamar mandi jenis ini biasanya banyak ditemukan di negara-negara Barat, dimana ketersediaan air dan kondisi ruangan yang lebih baik. Sedangkan pada setiap sebelum dan ketika penggunaan di kedua jenis kamar mandi, pengguna kamar mandi kurang dapat melihat kondisi air, penerangan cahaya lampu, kelembapan ruangan dan ketersediaan perlengkapan mandi seperti sabun dan sampo. Kekurangan ini dapat diatasi dan ditingkatkan dengan sistem *Smart Bathroom* yang berupa bagian dari sistem *Smart Home*. Sensor dan perangkat pada sistem ini dapat dimonitor dan dikontrol secara otomatis [1].

Dalam membangun sistem *Smart Bathroom*, *IoT* memiliki peran penting. Melalui *IoT*, hampir setiap benda sehari-hari mampu di hubungkan ke internet [2]. Kemudian sesuai survei pada *WHAN* (*Wireless Home Automation Network*), beberapa sistem yang sesuai atau cocok digunakan untuk *WHAN* memiliki fitur utama meliputi *light control*, *remote control*, *smart energy*, *report*, *monitor*, *security and safety*. Beberapa sistem yang berbeda juga memiliki aspek fitur yang berbeda pula [3]. Sedangkan implementasi dan perancangan *Smart Bathroom* berbasis *IoT* ini berfokus pada kontrol kelembapan udara dan penerangan kamar mandi serta *monitoring* kondisi air dan penerangan cahaya lampu serta mengukur kuantitas ketersediaan sabun dan sampo. Fitur *monitoring* dibuat menggunakan *website* dan terintegrasi dengan mikrokontroler, sensor dan aktuator. Pada sistem ini, data dari sensor mampu di kirim oleh mikrokontroler ke *database* melalui jaringan internet. Data yang telah diterima di *database* akan diolah dan ditampilkan *website*. Aksesibilitas informasi dan peringatan dari sistem ini dapat diperoleh *user* melalui *website*.

Jadi dengan sistem ini, diharapkan informasi tentang kondisi air dan penerangan cahaya lampu di kamar mandi diperoleh pengguna. Pada sistem ini, peringatan juga diharapkan diperoleh pengguna ketika ketersediaan perlengkapan sabun dan sampo di kamar mandi akan dan telah habis. Sehingga sistem ini mampu menjadi nilai tambah dalam penggunaan kamar mandi.

2. Dasar Teori

2.1 Internet Of Things (IoT)

Internet of Things (IoT) merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus-menerus. 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 [4].

2.2 Google Firebase

Firebase adalah suatu layanan dari Google yang digunakan untuk mempermudah para pengembang aplikasi dalam mengembangkan aplikasi. Dengan adanya Firebase, pengembang aplikasi bisa fokus mengembangkan aplikasi tanpa harus memberikan usaha yang besar. Ada beberapa fitur yang disediakan oleh Google Firebase yaitu *realtime database*, *crash reporting*, *authentication*, *cloud storage*, *cloud messaging*, *hosting* dan masih banyak lagi.

2.2.1 Firebase RealTime Database

Firebase menyediakan *realtime database* dan *backend* sebagai layanan. Layanan ini menyediakan pengembang Antarmuka pemrograman aplikasi yang memungkinkan data aplikasi disinkronkan di seluruh klien dan disimpan di *Firebase cloud*. [5][6] Perusahaan menyediakan pustaka *client* yang memungkinkan integrasi dengan Android, iOS, JavaScript, Java, Swift dan Node.js aplikasi. Database juga dapat diakses melalui REST API dan mengikat untuk beberapa JavaScript Frameworks seperti AngularJS, React, Ember.js dan Backbone.js.[7] Developer yang menggunakan *realtime database* dapat mengamankan data dengan menggunakan aturan keamanan yang diberlakukan oleh *server* perusahaan.[8] *Cloud Firestore* yang merupakan generasi mendatang dari *Firebase Realtime Database* dirilis untuk penggunaan beta. *Firebase Realtime Database* merupakan *database* yang di-host pada *cloud*. Data disimpan sebagai JSON dan disinkronkan secara *realtime* ke setiap klien yang terhubung. [9]

2.3 Website[10]

Website merupakan tempat penyimpanan data dan informasi berdasarkan topik tertentu. Diibaratkan *website* ini seperti buku yang berisi topik tertentu. *Web Pages* (halaman *website*): merupakan sebuah halaman khusus dari situs *website* tertentu. Diibaratkan halaman *website* ini seperti halaman khusus dari buku. *Homepage*: merupakan sampul halaman yang berisi daftar isi atau menu dari sebuah situs *website*. Beberapa unsur dalam halaman *website* yaitu:

1. Domain

Domain dapat diibaratkan sebuah *merk*/nama dalam sebuah produk dan juga digunakan untuk menemukan *website*. Oleh sebab itu dalam penamaan *domain* pada *website* disarankan untuk memberikan nama yang unik untuk menarik perhatian pengunjung *website*. Istilah umumnya adalah URL (*Uniform Resource Locator*)

2. Hosting

Hosting juga diperoleh dengan menyewa. Besarnya *hosting* ditentukan ruangan *harddisk* dengan ukuran MB (*Mega Byte*) atau GB (*Giga Byte*). Lama penyewaan *hosting* rata-rata dihitung per tahun. Penyewaan *hosting* dilakukan dari perusahaan-perusahaan penyewa *website hosting* yang banyak dijumpai baik di Indonesia maupun luar Negeri.

3. Scripts/Bahasa Pemrograman

Adalah bahasa yang digunakan untuk menerjemahkan setiap perintah dalam situs yang pada saat diakses. Jenis *scripts* sangat menentukan statis, dinamis atau interaktifnya sebuah situs. Semakin banyak ragam *scripts* yang digunakan maka akan terlihat situs semakin dinamis, dan interaktif serta terlihat bagus. Bagusnya situs dapat terlihat dengan tanggapan pengunjung serta frekuensi kunjungan.

Beragam *scripts* saat ini telah hadir untuk mendukung kualitas situs. Jenis *scripts* yang banyak dipakai para designer antara lain HTML, ASP, PHP, JSP, JavaScripts, Java applets dsb. Bahasa dasar yang dipakai setiap situs adalah HTML sedangkan ASP dan lainnya merupakan bahasa pendukung yang bertindak sebagai pengatur dinamis, dan interaktifnya situs.

2.4 Hypertext Markup Language (HTML) [11]

Hypertext Markup Language (HTML) merupakan standar bahasa yang digunakan untuk menampilkan dokumen *website*, yang bisa anda lakukan:

- Mengontrol tampilan dari web page dan kontennya.
- Memublikasikan dokumen secara online sehingga dapat di akses.
- Membuat online form yang bisa digunakan untuk menangani pendaftaran, transaksi secara online.
- Menambahkan objek seperti gambar, suara, video dan juga dapat java applet dalam dokumen HTML.

2.5 Framework CSS[10]

Cascading Style Sheet (CSS) adalah mekanisme *seederhana style sheet* yang mengijinkan desiner dan *user* untuk menempatkan *style*, misalnya huruf, warna dan spasi ke dalam dokumen HTML. Aturan dalam CSS terdiri atas 2 bagian utama, yaitu:

- Selektor, yaitu sebuah string yang mengidentifikasi apakah sebuah elemen berhubungan dengan aturan yang akan diimplementasikan. Contoh, H1
- Deklarasi, yaitu properti misalnya *font-size* dengan nilai yang sesuai misalnya 12pt. Contoh, *font-size: 12pt*

2.5.1 Bootstrap

Bootstrap adalah sebuah *library framework* CSS yang di buat khusus untuk bagian pengembangan *front-end website*. Bootstrap merupakan salah satu *framework* HTML, CSS dan JavaScript yang paling populer di kalangan *website developer*. Pada saat ini hampir semua *web developer* telah menggunakan bootstrap untuk membuat tampilan *front-end* menjadi lebih mudah dan sangat cepat. karena anda hanya perlu menambahkan *class-class* tertentu untuk misalnya membuat tombol, *grid*, navigasi dan lainnya.

2.6 JavaScript[11]

Javascript merupakan salah satu *script* pemrograman yang dijalankan pada *pc client* melalui *web browser* (*client side programming*), berbeda dengan PHP yang merupakan pemrograman yang dijalankan di server (*server side programming*). Harap diingat bahwa JavaScript berbeda dengan Java. Meskipun hampir sama dalam penamaannya, namun struktur pemrogramannya benar-benar berbeda. Pada *Java programming* juga dikenal beberapa ekstensi untuk *web programming* seperti Java Server Page, Java Applet, dan Java Servlet, namun JavaScript bukan merupakan bagian dari itu. Kelebihan menggunakan *client side scripting* ini adalah bisa digunakan untuk membuat tampilan yang atraktif dan lebih dinamis. Kekurangannya adalah karena merupakan *client side scripting*, maka kode program bisa dilihat oleh *user* dan bisa menimbulkan celah keamanan bila salah dalam penggunaannya. Biasanya javascript digunakan untuk merespon *input user* dengan lebih atraktif, memvalidasi input *user*, mengganti konten maupun atribut agar lebih dinamis, bahkan bisa juga untuk menuliskan perintah html dari kode JavaScript.

2.6.1 Charts.js

Chart.js adalah modul tambahan JavaScript untuk membuat suatu diagram statistik dari data yang tersedia. Diagram yang dapat dibuat contohnya ada *line* diagram, *pie* diagram, diagram batang dan lainnya.

2.7 MySQL[12]

MySQL merupakan salah satu DBMS *open source* yang paling populer pada saat ini. Meskipun dahulu MySQL pernah dikritisi karena tidak memiliki beberapa fitur yang ada dalam DBMS pada umumnya, namun saat ini MySQL sudah banyak dikembangkan.

2.8 PHP[12]

PHP (PHP: *Hypertext Preprocessor*) merupakan bahasa pemrograman web yang dapat disisipkan dalam *script* HTML. Banyak sintaks di dalamnya yang mirip dengan bahasa C, Java dan Perl. Tujuan dari bahasa ini adalah membantu para pengembang web untuk membuat web dinamis dengan cepat.

Ketika seseorang mengunjungi web berbasis PHP, web server akan memproses kode-kode PHP. Beberapa perintah atau kode dari PHP tersebut selanjutnya ada yang diterjemahkan ke dalam HTML dan beberapa ada yang disembunyikan (misalnya proses kalkulasi dan operasi). Setelah diterjemahkan ke dalam HTML, web server akan mengirim kembali ke web browser pengunjung tersebut.

2.9 Cron Jobs

Cron Jobs adalah *software* yang berguna untuk melakukan penjadwalan dan memungkinkan untuk menjalankan *script* tertentu secara otomatis dan terjadwal. Pengguna tidak perlu melakukan hal yang sama secara berulang-ulang seperti *mendownload file*, melakukan *backup*, mengirim *email* dan lainnya. Salah satu tugas Cron Jobs yaitu menghapus *file* yang bersifat sementara untuk membersihkan ruang penyimpanan *disk*.

2.10 Perhitungan Total Penggunaan Air dan Sampo

Untuk mendapatkan nilai total penggunaan air dan sampo, maka digunakan persamaan dibawah ini :

1. Total Penggunaan Air

$$\frac{\text{Kapasitas Maksimum bak mandi(Liter)}}{\text{Ketinggian Maksimum bak mandi(cm)}} \times \text{Nilai Pemakaian(cm)}$$

Keterangan :

- a) Kapasitas maksimum bak mandi (V_{tabung}) = 32 Liter

$$\begin{aligned} V_{tabung} &= \pi r^2 t \\ V_{tabung} &= 3,14 \times 19,5 \text{ cm} \times 19,5 \text{ cm} \times 27 \text{ cm} \\ V_{tabung} &= 32.237 \text{ cm}^3 \\ V_{tabung} &= 32 \text{ Liter} \end{aligned}$$

- b) Ketinggian maksimum bak mandi (t) = 27 cm

2. Total Penggunaan Sampo

$$\frac{\text{Kapasitas Maksimum wadah sampo(Liter)}}{\text{Ketinggian Maksimum wadah sampo(cm)}} \times \text{Nilai Pemakaian(cm)}$$

Keterangan :

- a) Kapasitas maksimum wadah sampo (V_{sampo}) : 0,48 Liter

$$\begin{aligned} \text{Volume wadah sampo} &= \text{panjang} \times \text{lebar} \times \text{tinggi} \\ \text{Volume wadah sampo} &= 5 \text{ cm} \times 8 \text{ cm} \times 12 \text{ cm} \\ \text{Volume wadah sampo} &= 480 \text{ cm}^3 \\ \text{Volume wadah sampo} &= 0,48 \text{ Liter} \end{aligned}$$

b) Ketinggian maksimum wadah sampo : 12 cm

2.11 Perhitungan Sisa Air dan Sampo Yang Tersedia

Untuk mendapatkan nilai total penggunaan air dan sampo, maka digunakan persamaan dibawah ini :

1. Sisa Air yang tersedia

$$\text{Kapasitas Maksimum bak mandi(Liter)} - \text{Total Penggunaan Air (Liter)}$$

Keterangan :

a) Kapasitas maksimum bak mandi : 32 Liter

2. Sisa Sampo yang tersedia

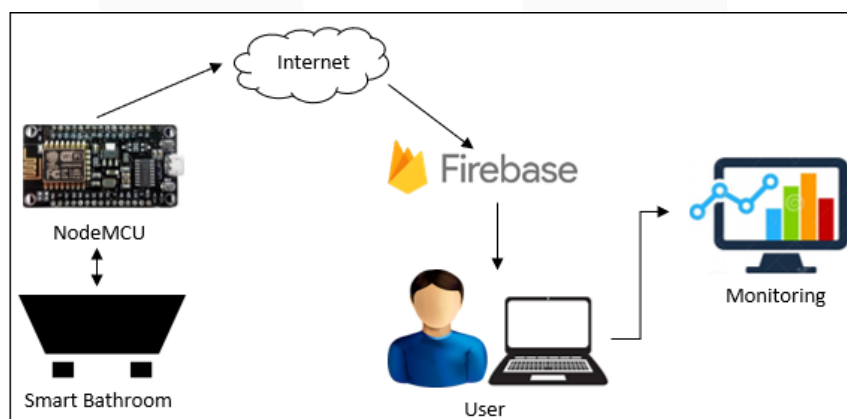
$$\text{Kapasitas Maksimum wadah Sampo (Liter)} - \text{Total Penggunaan Sampo (Liter)}$$

Keterangan :

a) Kapasitas maksimum bak mandi : 0,48 Liter

3. Blok Diagram Sistem

3.1 Blok Diagram



Gambar 3. 1 Blok Diagram Perancangan Sistem Smart Bathroom

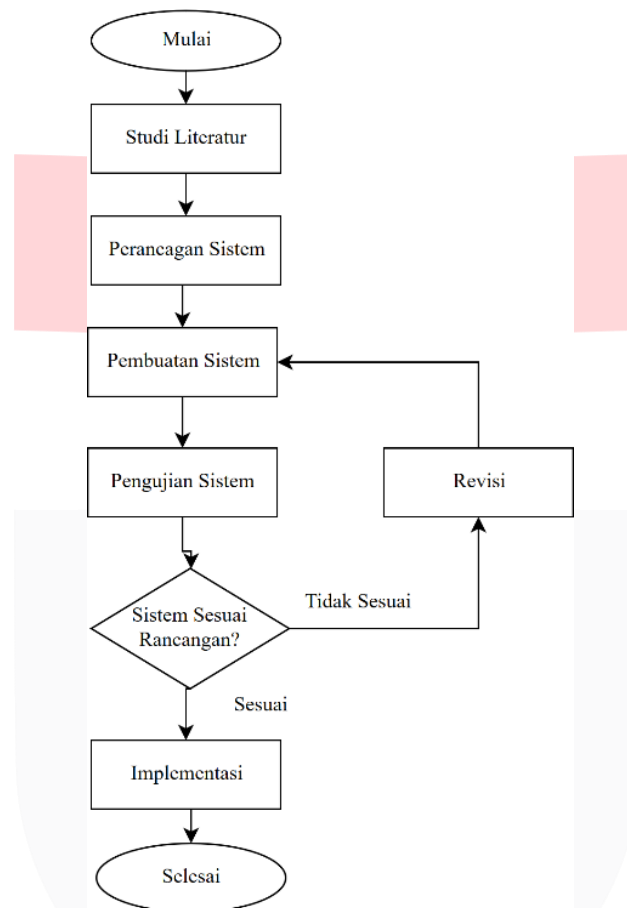
Pada gambar 3.1 menunjukkan bahwa aplikasi yang dibuat akan terintegrasi dengan sebuah *hardware* yaitu *Smart Bathroom* yang akan dikerjakan pada penelitian lainnya dan fokus Proyek Akhir ini hanya pada *software*. Dimana *hardware* tersebut sebagai sensor pendeteksi secara *realtime* apabila ada perubahan kondisi pada alat yang terhubung yang nantinya akan ditampilkan di aplikasi. *Software* yang digunakan terintegrasi dengan Google Firebase yang berfungsi sebagai *Mobile Apps Developer*, seperti *realtime Database*. Dalam hal ini data air, shampo, lampu dan fan yang diterima oleh sensor akan dikirimkan ke Firebase. Hasil yang sudah masuk ke Firebase akan ditampilkan di halaman *website*. Pengguna juga dapat memantau kondisi kamar mandi, apakah sedang digunakan atau tidak (lampu dan exhaust fan akan otomatis menyala), berapa % (persen) kadar air dan shampo yang tersedia di kamar mandi milik pengguna. Lalu sistem akan memberikan notifikasi melalui email saat ada perubahan kondisi kamar mandi.

Pada proyek akhir ini web interface diintegrasikan dengan menggunakan Google Firebase *realtime database* dan hosting rumah web Indonesia yang memiliki *Cron*.

3.2 Diagram Alir Sistem

Pada bab ini akan dijelaskan mengenai perancangan website yang akan dibuat menggunakan Google Firebase dan My SQL yang berguna sebagai *database*. Aplikasi ini memiliki fitur untuk *monitoring* penggunaan air, shampo, lampu dan *exhaust fan* dari aplikasi mikrokontroller *smart bathroom*. Monitoring ini berguna agar user dapat tetap mengetahui kondisi kamar mandi miliknya walaupun dari jarak yang jauh karena sistem akan diaplikasikan menggunakan *website*.

Adapun perancangan sistem dari aplikasi yang dibuat ditunjukkan pada gambar 3.2, sebagai berikut:



Gambar 3. 2 Diagram alir perancangan sistem

Diagram alir pada gambar 3.2 menjadi dasar dalam pembuatan aplikasi *website* pemantau untuk *smart bathroom*. Beberapa tahapannya yaitu membuat *site map* aplikasi, *use case diagram* dan *activity diagram*. Selanjutnya adalah merancang website dengan beberapa bahasa pemrograman yaitu Javascript, PHP, dan HTML. Jika *interface website* sudah bisa dijalankan, langkah selanjutnya yaitu mengintegrasikan *website* dengan Firebase dan My SQL karena menu yang terdapat pada *website* membutuhkan penyimpanan data atau *database*. Tahap selanjutnya adalah pengujian *website* yang sudah terintegrasi dengan alat dan *database*, dimana pada tahap ini akan dilihat apakah aplikasi *website* sudah berjalan sesuai dengan yang diinginkan. Jika terjadi kesalahan dalam pengujian, maka akan dilakukan analisa ulang terhadap kesalahan yang terjadi pada aplikasi, lalu akan dilakukan pengujian ulang di tahap ini. Apabila sistem sudah bekerja dengan baik dan sesuai dengan yang diharapkan maka aplikasi *website* sudah siap untuk digunakan.

3.3 Perancangan Sistem

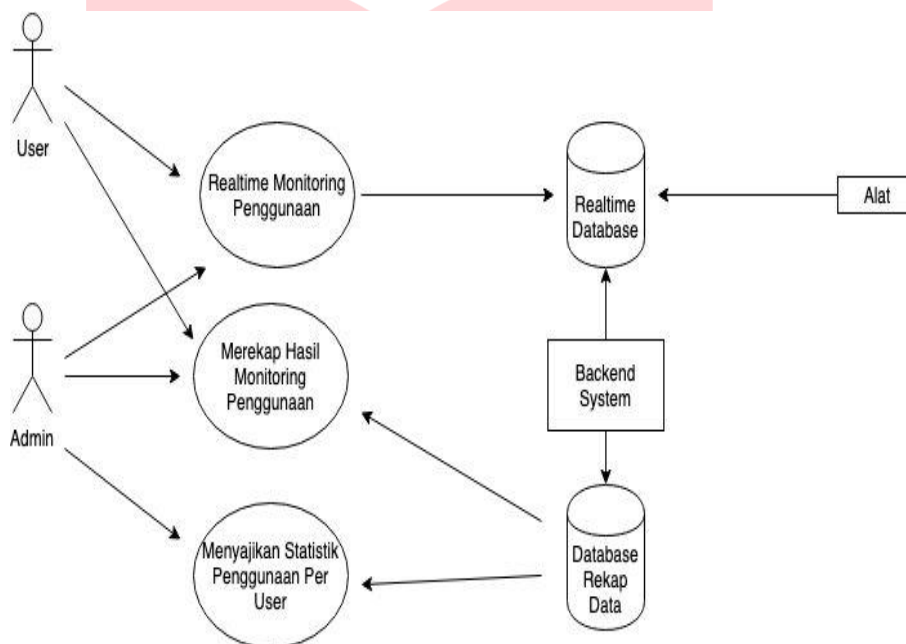
Sistem yang dibuat adalah aplikasi *website* untuk memantau aplikasi *Smart Bathroom*. Untuk mengakses *website* pengguna diharuskan untuk melakukan registrasi terlebih dahulu dengan memilih menu *register*. *Register* dibagi menjadi 2 (dua) bagian yaitu register sebagai *user* dan *register* sebagai pengelola. Perbedaannya terdapat pada saat mendaftarkan diri sebagai pengelola, maka pendaftar akan mendapatkan kode verifikasi dari *website*

sedangkan apabila mendaftarkan diri sebagai *user* tidak mendapatkan kode verifikasi. Setelah proses registrasi berhasil pengguna sudah bisa melakukan *log in* dan mengakses halaman utama *website*. Pada halaman utama *user* akan mendapatkan informasi tentang kondisi kamar mandi sedang digunakan atau tidak, kondisi lampu dan *exhaust fan* menyala atau tidak, berapa persentase (%) air dan sampo yang tersedia. Kemudian terdapat menu laporan penggunaan yang dapat merekap riwayat penggunaan air dan sampo selama 1 bulan. Apabila *user* telah selesai dapat melakukan *log out* dari halaman *website*.

Fitur yang terdapat pada *website* ini adalah *live statistic, reporting, dan management*. Untuk fitur *live statistic* dapat melakukan pemantauan kondisi kamar mandi yang terhubung dengan perangkat aplikasi *Smart Bathroom* secara *real time* apabila terjadi perubahan kondisi pada kamar mandi pengguna. Fitur lainnya yaitu *reporting* yang berguna untuk memberikan *report* kepada pengguna secara *real time* dan berkala sampai dengan jangka waktu 1 bulan. Dan fitur terakhir yaitu *management* yang berguna untuk melakukan pendaftaran perangkat dan pengguna baru yang mendaftarkan akun ke aplikasi *website* pemantau *Smart Bathroom*.

Use Case Diagram

3.3.2 Desain (Perancangan Sistem Aplikasi dan Tampilan Antarmuka)



Gambar 3.3 Use Case Diagram

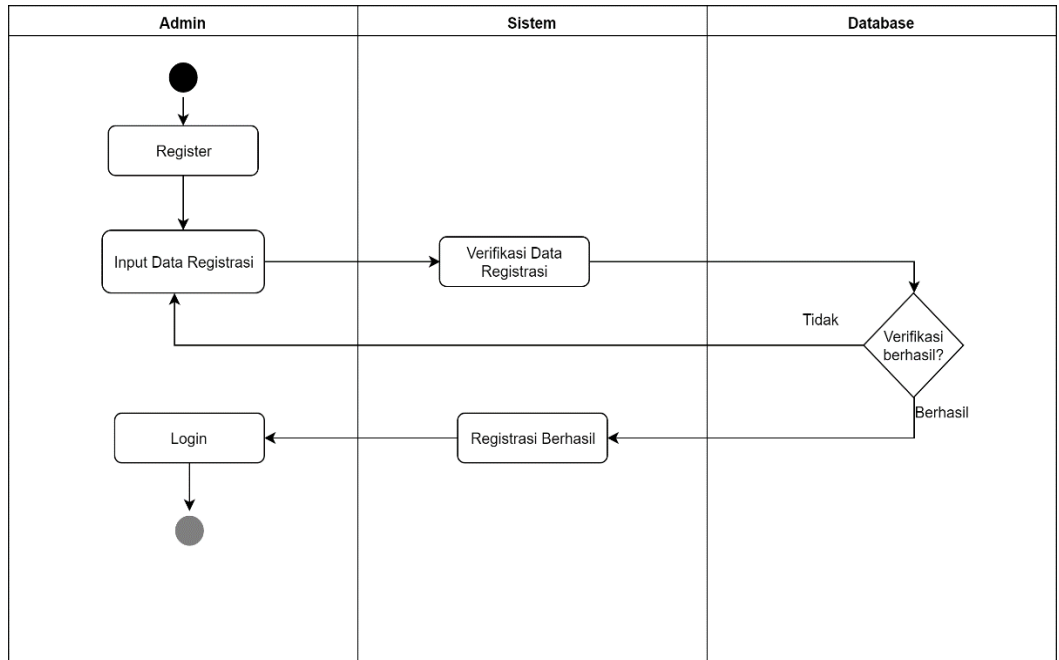
Dari gambar 3.3 diatas menjelaskan tentang alir aplikasi yang digambarkan pada *Use Case Diagram*. Alir pada *website* dibagi menjadi 2 (dua) bagian yaitu pada sisi *admin* dan sisi *user*. Adapun kebutuhan dari *website* ini adalah sebagai berikut:

1. Pada sisi *admin* dan *user* diharuskan untuk melakukan registrasi terlebih dahulu sebelum melanjutkan ke halaman *login*. Karena untuk terintegrasi dengan alat *Smart Bathroom* membutuhkan nama, *E-mail* dan *password*.
2. Lanjut ke tahap selanjutnya yaitu menu *Login*. Di menu ini, *admin* ataupun *user* harus memasukkan *email* dan *password* yang sudah didaftarkan saat proses registrasi. Khusus untuk *user* harus memasukkan kode admin yang *valid* untuk terdaftar sebagai anggota *admin* tertentu.
3. Setelah *Login* berhasil maka kategori keberhasilan dibagi menjadi 2 (dua) yaitu:
 1. Pada sisi *user*:
 - *User* berhasil masuk ke *dashborad user* dan memonitoring air, sampo, lampu dan *exhaust fan*.
 - *User* dapat melihat laporan penggunaan harian dan memilih *range* tanggal yang akan dipilih.
 - *User* berhasil melakukan *print* data untuk mencetak hasil penggunaan.
 2. Pada sisi *admin*:
 - *Admin* berhasil ke halaman *dashboard* admin dan melihat data grafik statistik penggunaan *user* yang dipilih dan memilih *range* pemantauan penggunaan *user*.

- Admin berhasil *memonitoring user* yang terdaftar secara *realtime* dihari itu dan akan ditampilkan dalam bentuk grafik.
- Admin dapat melihat laporan penggunaan harian dan memilih *range* tanggal yang akan dipilih.
- Admin berhasil melakukan *print* data untuk mencetak hasil penggunaan.

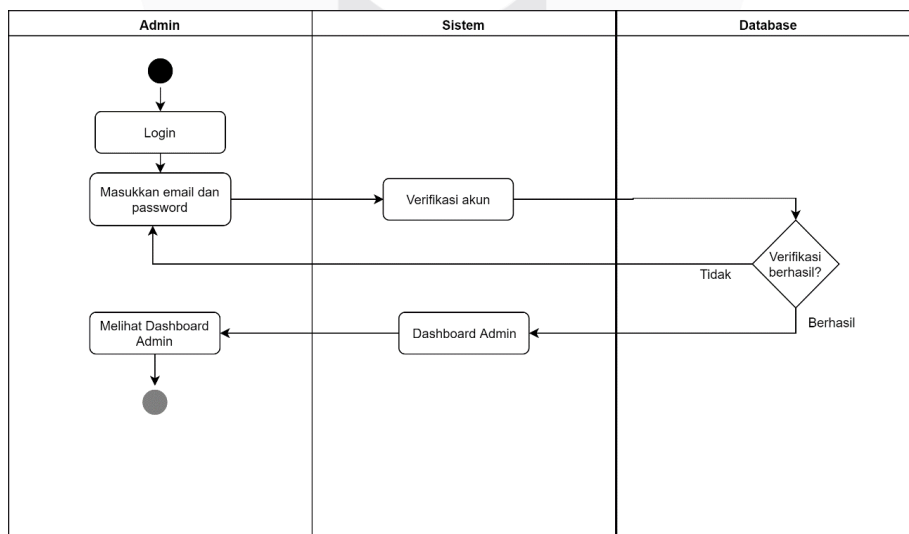
3.3.3 Diagram Activity

3.3.2.1 Pada Sisi Admin



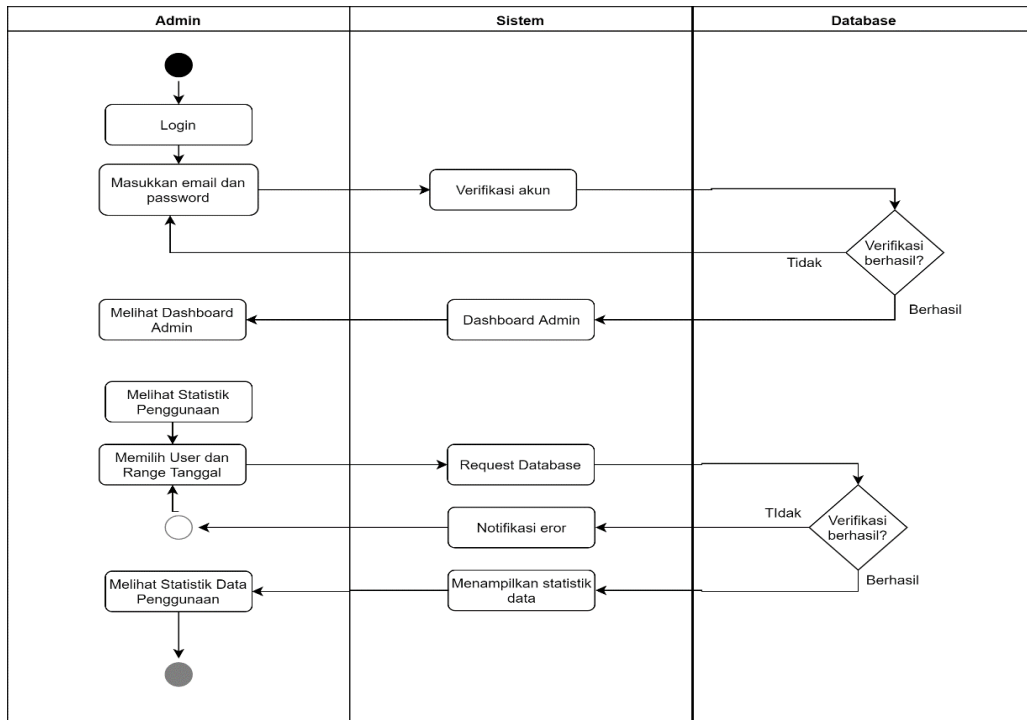
Gambar 3. 4 Diagram Activity Registrasi Admin

Pada gambar 3.4 menunjukkan bahwa sebelum dapat melakukan poses *Login*, *admin* diharuskan untuk melakukan proses registrasi terlebih dahulu dengan memasukkan data berupa nama, *E-mail* dan *password*.



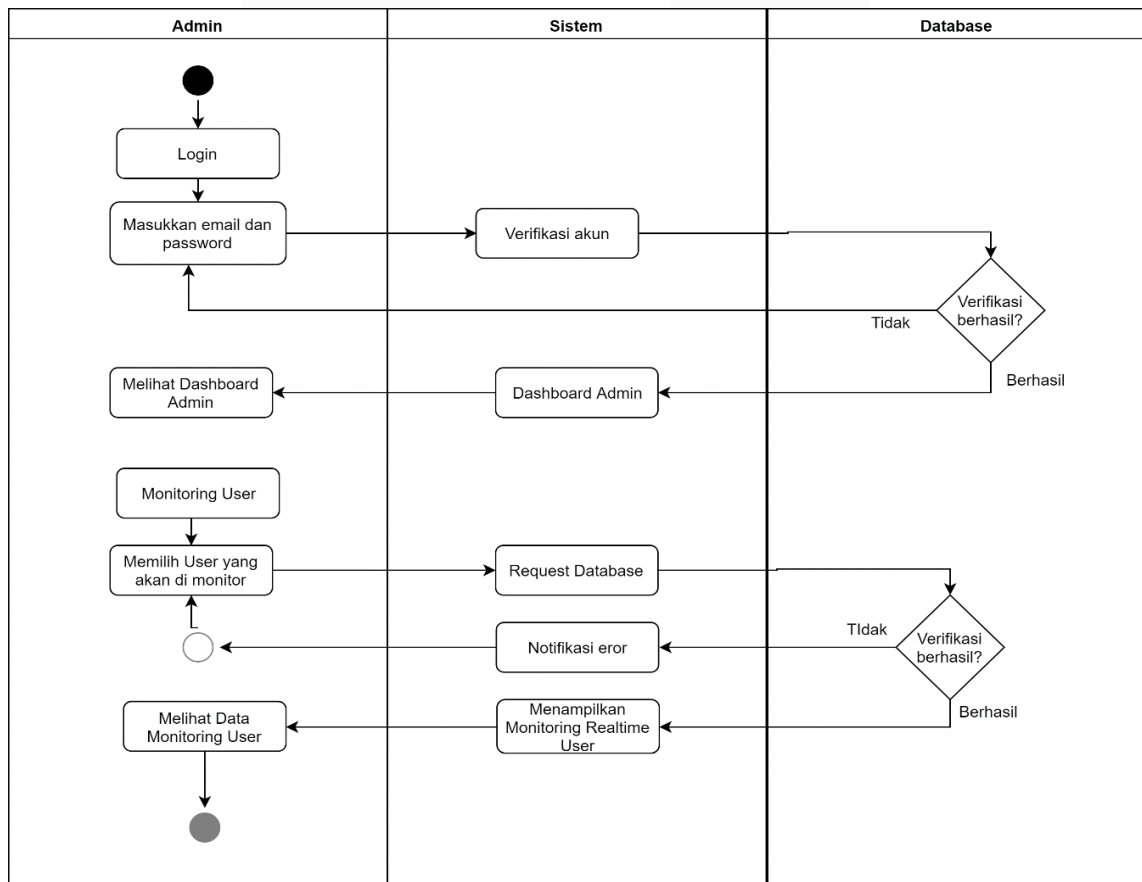
Gambar 3. 5 Diagram Activity Login Admin

Pada gambar 3.5 diatas menunjukkan bahwa untuk dapat mengakses data informasi aplikasi *Smart Bathroom user* diharuskan melakukan *Login* terlebih dahulu.



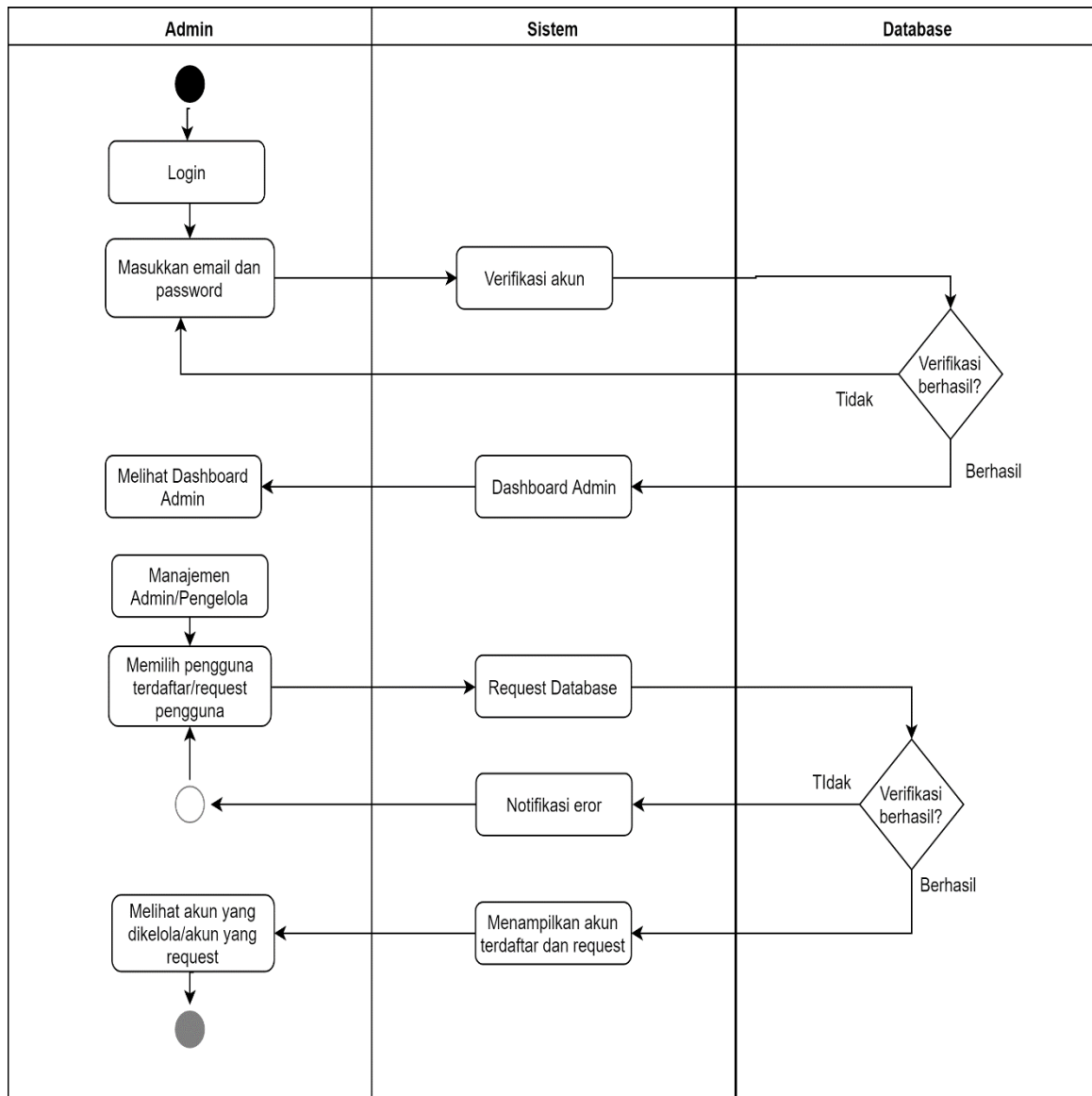
Gambar 3. 6 Diagram Activity Statistik Data User di Admin

Pada gambar 3.6 diatas menunjukkan setelah *admin* berhasil melakukan *login* maka *admin* akan masuk ke halaman dashboard admin. Setelah itu admin dapat memantau statistik penggunaan harian *user* yang terdaftar sebagai anggota *admin*.



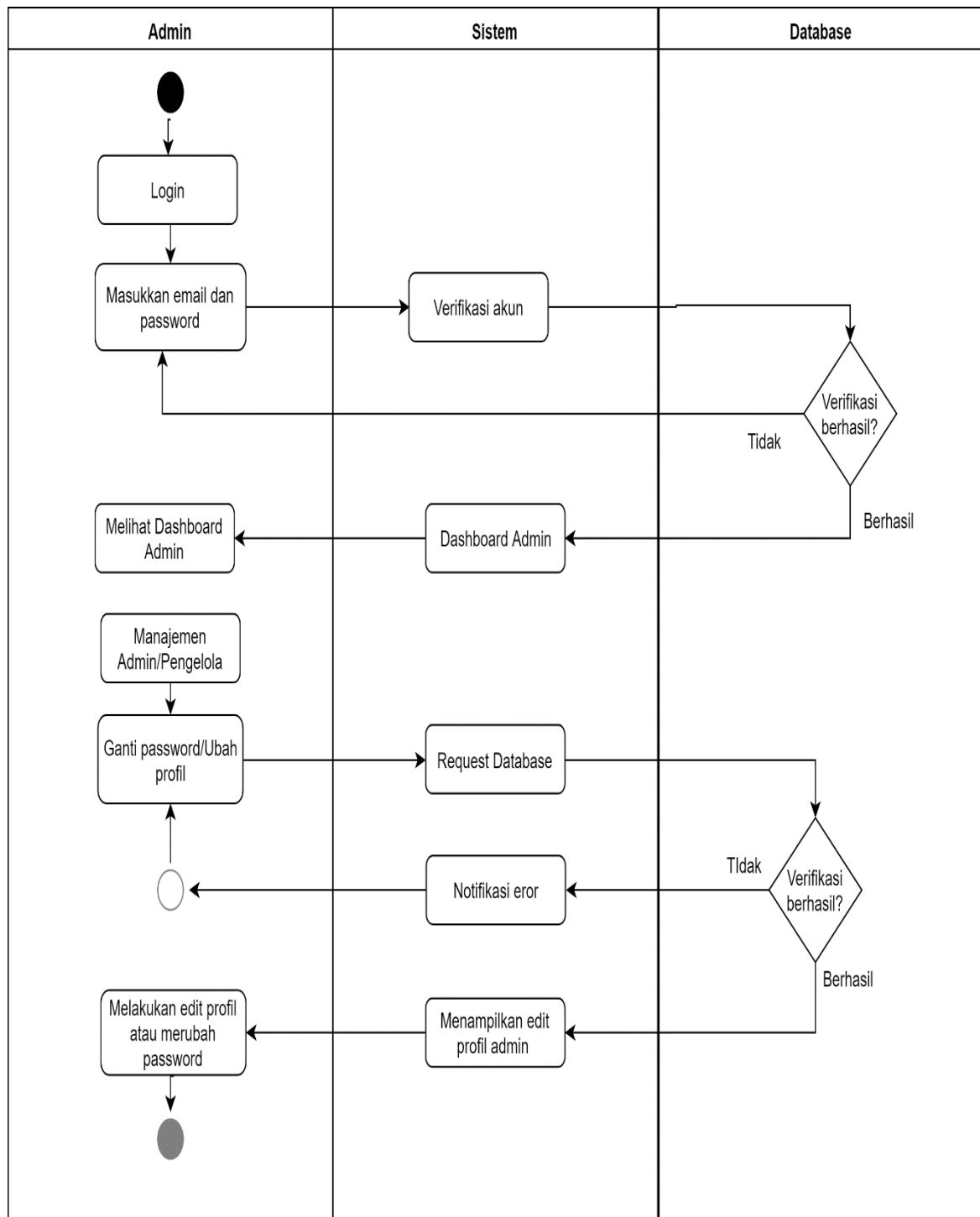
Gambar 3. 7 Diagram Activity Monitoring Real-time User Pada Admin

Pada gambar 3.7 diatas menjelaskan bahwa salah satu aktivitas *admin* yaitu dapat melakukan *monitoring* ke perangkat *user* secara *realtime* yang terdaftar sebagai anggota dari *admin*. Data yang diperoleh oleh *admin* akan berupa grafik dinamis yang akan bergerak mengikuti perubahan yang terjadi pada kamar mandi *user*.



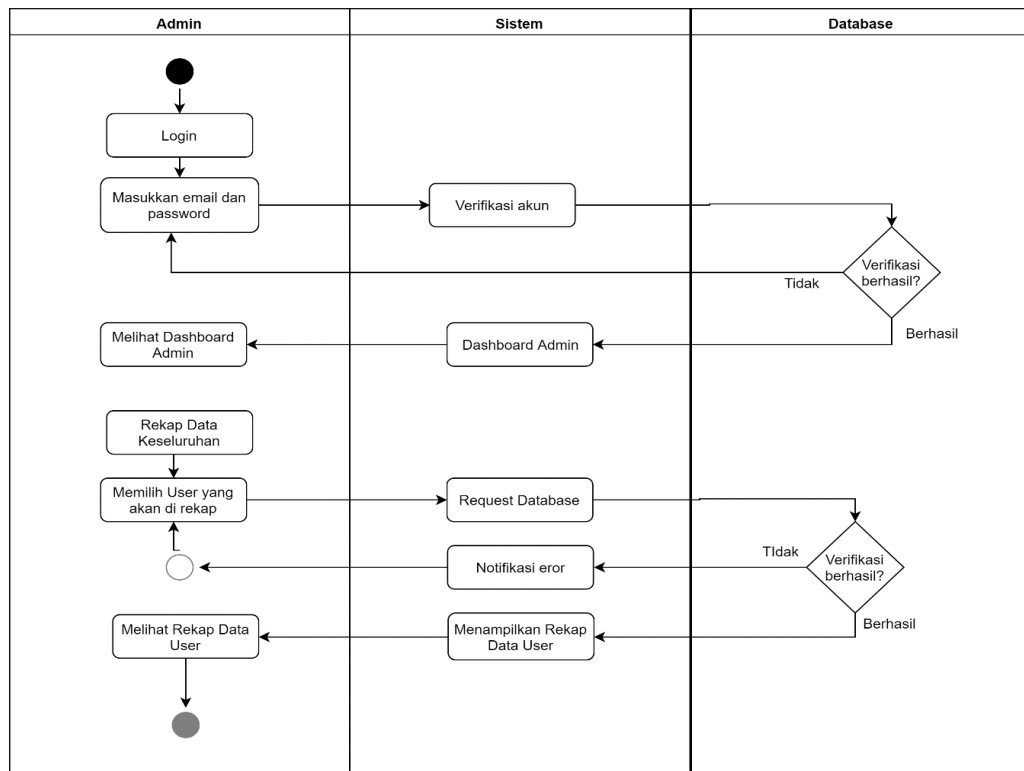
Gambar 3. 8 Diagram Activity Pengelolaan user

Pada gambar 3.8 diatas dijelaskan bahwa *admin* dapat melakukan pengelolaan terhadap *user* yang terdaftar atau yang melakukan *request*. *Admin* selaku pengelola akan menerima informasi berapa jumlah *user* yang terdaftar dan juga akan daftar *user* yang melakukan *request* sebagai anggota pengelolaannya. *Admin* bisa menghapus perangkat *user* yang terdaftar dan menerima atau menolak *request* dari *user*. Fitur ini dibuat untuk *admin* agar dapat melakukan kontrol terhadap jumlah *user* yang terdaftar. Karena setiap *admin* memiliki kode pengelolaan yang berbeda memungkinkan user untuk langsung terhubung dengan admin yang dipilihnya. Namun saat admin merasa user yang terdaftar sudah melebihi kapasitas atau user melakukan kesalahan dalam input kode pengelola maka admin dapat melakukan penolakan permintaan atau menghapus perangkat.



Gambar 3. 9 Diagram Activity Manajemen admin melakukan edit profil

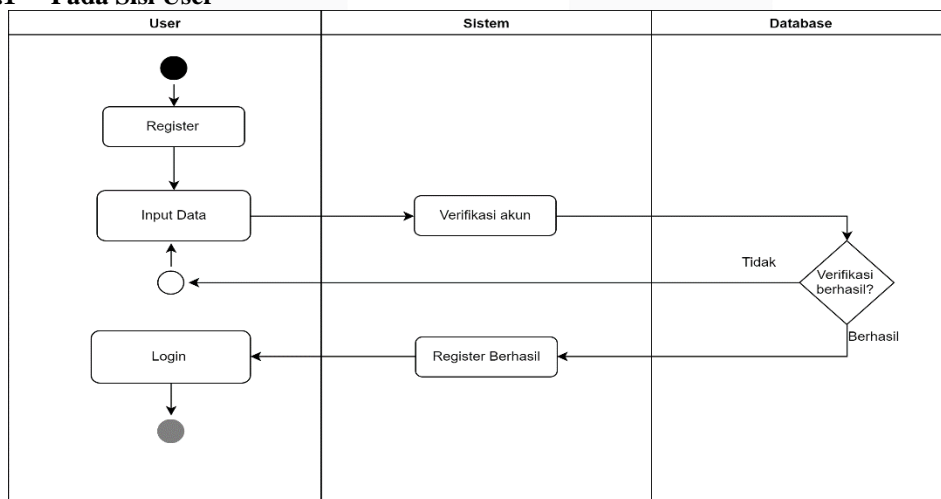
Pada gambar 3.9 diatas menjelaskan tentang aktivitas *admin* sebagai pengelola yang bisa melakukan *edit* profil dan mengganti *password* akun pribadinya. Dalam menu *edit* profil *admin* dapat merubah nama, *email*, *username* dan kode pengelolaannya. Fitur ini diberikan agar *admin* dapat merubah profilnya dan merubah *password* untuk alasan keamanan apabila ada yang berusaha untuk membajak akun *admin* atau pengelola.



Gambar 3. 10 Diagram Activity Rekap Data User Pada Admin

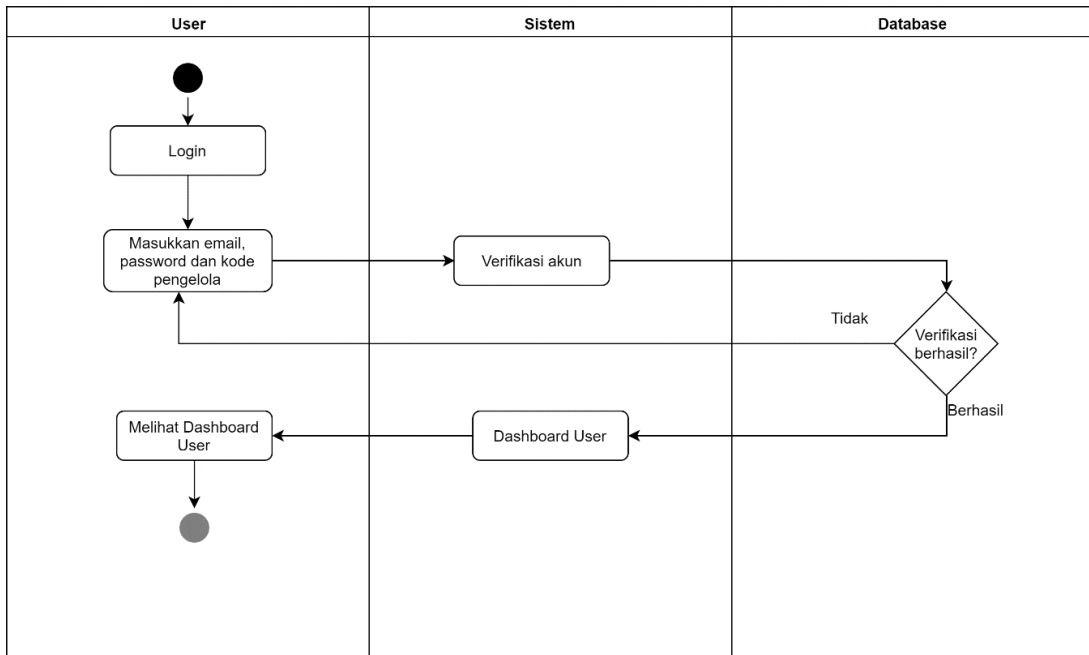
Pada gambar diatas menjelaskan bahwa *admin* dapat melihat rekap data *user* dalam range waktu tertentu.

3.3.2.1 Pada Sisi User



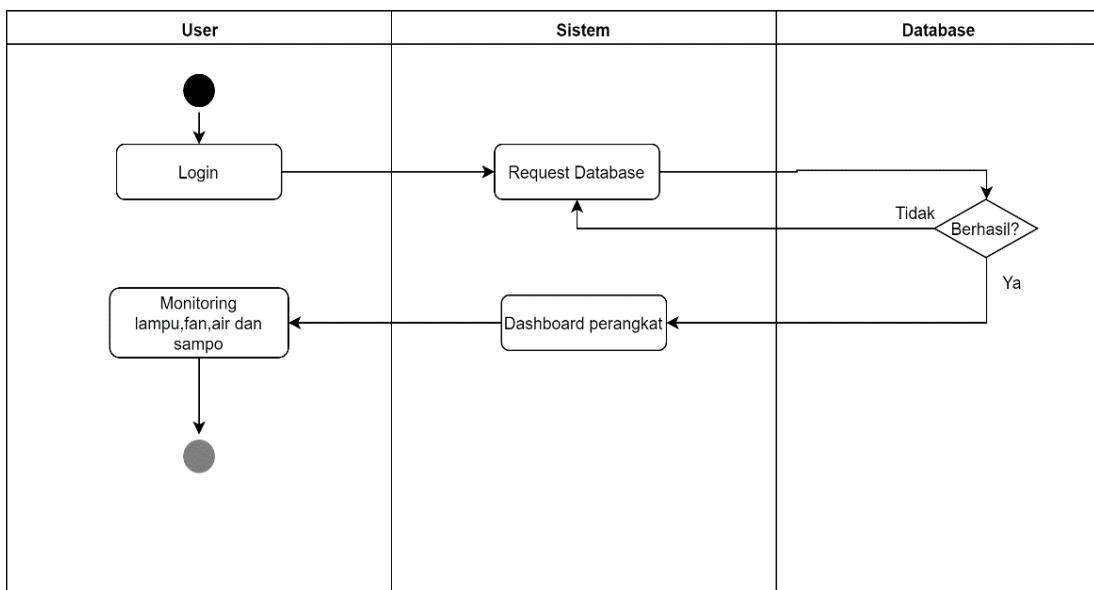
Gambar 3. 11 Diagram Activity Register User

Pada gambar 3.11 diatas menunjukkan bahwa sebelum melakukan *login user* diharuskan untuk melakukan proses *register* terlebih dahulu dan setelah berhasil maka *user* akan mendapatkan kode pengelola untuk di daftarkan ke anggota *admin*.



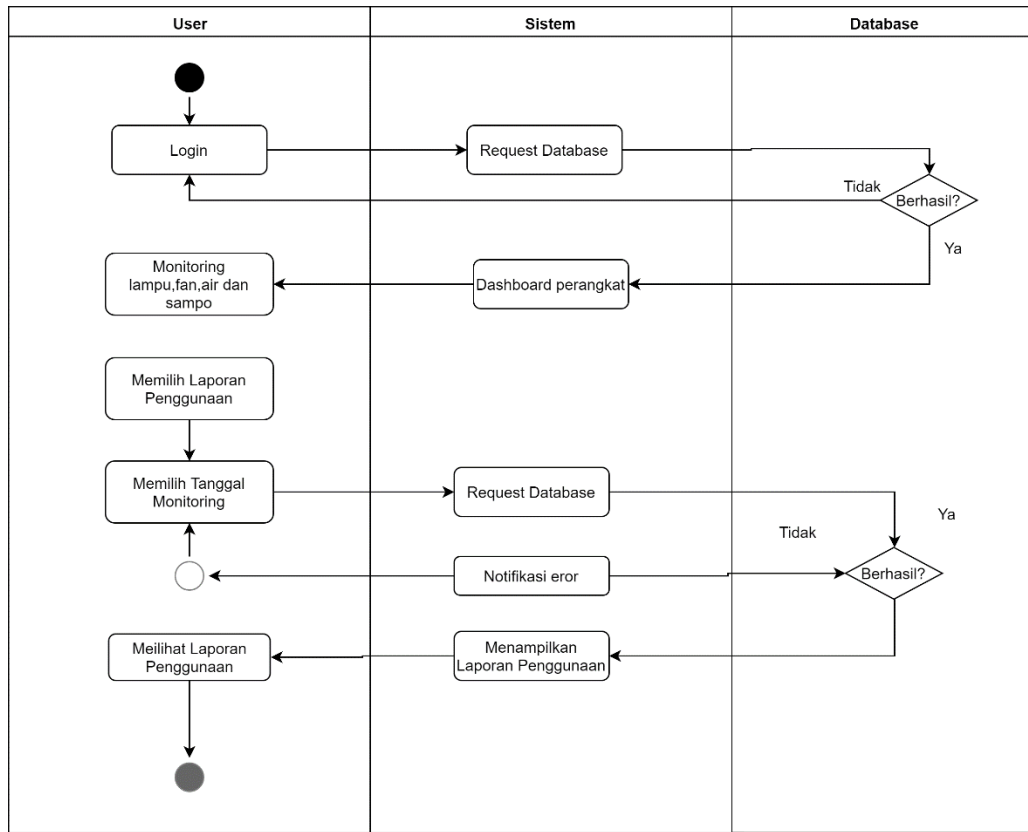
Gambar 3. 12 Diagram Activity Login User

Pada gambar 3.12 diatas menunjukkan setelah berhasil melakukan *register* pada halaman *website* maka *user* sudah bisa masuk ke halaman *dashboard user*.



Gambar 3. 13 Diagram Activity Monitoring

Pada gambar 3.13 menjelaskan bahwa setelah berhasil melakukan *login user* juga dapat melakukan *monitoring* kondisi air, sampo, lampu dan *exhaust fan* pada kamar mandi *user* dalam bentuk ilustrasi gambar yang dinamis.



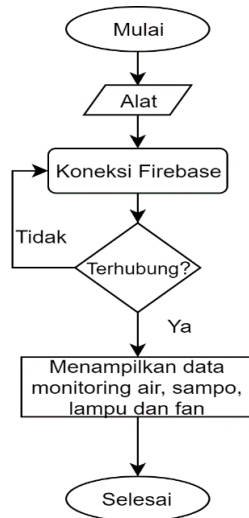
Gambar 3. 14 Diagram Activity Laporan Penggunaan

Pada gambar 3.14 diatas menjelaskan bahwa *user* juga dapat memilih menu laporan penggunaan yang akan menampilkan laporan penggunaan harian *user* pada jangka waktu tertentu yang bisa dipilih oleh *user*.

3.3.3 Sistem Perancangan Pemantauan

Aplikasi *Website* dari sisi pemantauan perangkat *smart bathroom* yang terintegasi dengan *website* dapat mengetahui beberapa fitur yang ada pada aplikasi. *Website* akan menampilkan kondisi air dan sampo dalam satuan liter dan persentasenya, kondisi lampu dan *exhaust fan* dalam bentuk gambar yang dinamis. Informasi yang diterima didapatkan dari perangkat *hardware* yang terhubung dengan *Firestore* yang akan mengirimkan data dan ditampilkan di *dashboard user*. Kemudian data tersebut ditunjukkan agar *user* dapat mengetahui seberapa banyak penggunaan setiap fitur yang tersedia pada kamar mandi pribadinya.

Flowchart perancangan sisteersebut dapat ditunjukkan pada gambar dibawah ini:



Gambar 3. 15 Flowchart Perancangan Sistem

Data yang diterima oleh website akan diolah pada sistem dan akan ditampilkan dalam bentuk gambar dan nilai untuk setiap fitur. Untuk lampu dan *exhaust fan* akan mendapatkan keterangan *on/off* yang berarti kamar mandi sedang digunakan.

Pada sistem pemantauan data yang dibaca oleh sensor PIR dan ultrasonik akan dikirim oleh NodeMCU melalui jaringan IP yang akan ditampilkan di *website*. *Website* akan mengambil data dari *firebase realtime database* dan akan ditampilkan pada halaman *website*. Karena menggunakan sistem *realtime* maka perubahan-perubahan kecil akan tetap tercatat sebagai perubahan. Untuk nilai kondisi air dan sampo menggunakan rumus volume yaitu luas penampang dikali ketinggian yang didapatkan yang terhubung dengan sensor ultrasonik. Tampilan lampu dan fan akan menyala bersamaan karena terhubung dengan satu sensor yaitu sensor PIR. Untuk menampilkan hasil pengolahan data air dan sampo pada halaman *website* maka rumus volume tadi akan dimasukkan kedalam *software* bahasa pemrograman untuk diolah dan ditampilkan di halaman *website*.

3.3.4 Perancangan Layout

Pada proses pembuatan halaman *website* untuk proyek akhir ini penulis menggunakan aplikasi Bootstrap untuk menyusun tampilan *website* secara keseluruhan. Bootstrap menyediakan banyak *framework css* yang berfungsi untuk memperindah tampilan *website* dan dapat diubah sesuai kebutuhan. Berikut merupakan tampilan *website* yang dibuat:

a. Dashboard

SMART BATHROOM	Register Login
<h1>Selamat Datang di Aplikasi Smart Bathroom</h1> <p>Solusi Monitoring Kamar Mandi Pintar</p>	
<h2>Apa itu Smartbathroom ?</h2> <p>Smart Bathroom adalah Solusi Monitoring sistem kamar mandi Pintar Anda</p>	
<h2>Fitur Kami</h2> <p>Live Statistic Reporting Management</p>	
Copyright @ 2019 All Right Reserved Created by Christhoper	

Gambar 3. 16 Layout Halaman Utama

Pada gambar 3.14 diatas menunjukkan halaman utama *website* yang dimana berisi tentang informasi pengenalan pada aplikasi *smart bathroom* dalam Proyek Akhir yang telah dibuat. Pada halaman utama ini ditunjukkan untuk semua orang yang mengakses *website* pada aplikasi *smart bathroom*.

b. Login dan Register

<table border="1"> <tr> <td>SMART BATHROOM</td> <td>Register Login</td> </tr> <tr> <td colspan="2"> <h1>Selamat Datang di Aplikasi Smart Bathroom</h1> <p>Solusi Monitoring Kamar Mandi Pintar</p> </td> </tr> <tr> <td colspan="2"> <h2>Apa itu Smartbathroom ?</h2> <p>Smart Bathroom adalah Solusi Monitoring sistem kamar mandi Pintar Anda</p> </td> </tr> <tr> <td colspan="2"> <h2>Fitur Kami</h2> <p>Live Statistic Reporting Management</p> </td> </tr> <tr> <td colspan="2">Copyright @ 2019 All Right Reserved Created by Christhoper</td> </tr> </table>	SMART BATHROOM	Register Login	<h1>Selamat Datang di Aplikasi Smart Bathroom</h1> <p>Solusi Monitoring Kamar Mandi Pintar</p>		<h2>Apa itu Smartbathroom ?</h2> <p>Smart Bathroom adalah Solusi Monitoring sistem kamar mandi Pintar Anda</p>		<h2>Fitur Kami</h2> <p>Live Statistic Reporting Management</p>		Copyright @ 2019 All Right Reserved Created by Christhoper		<table border="1"> <tr> <td>Register</td> <td>X</td> </tr> <tr> <td colspan="2"> <p>Register User Register Pengelola</p> </td> </tr> <tr> <td colspan="2"> <p>Nama</p> <input type="text"/> </td> </tr> <tr> <td colspan="2"> <p>Email</p> <input type="text"/> </td> </tr> <tr> <td colspan="2"> <p>Username</p> <input type="text"/> </td> </tr> <tr> <td colspan="2"> <p>Password</p> <input type="password"/> </td> </tr> <tr> <td colspan="2"> <p>Ulangi Password</p> <input type="password"/> </td> </tr> <tr> <td colspan="2"> <p>Kode Pengelola (Optional)</p> <input type="text"/> </td> </tr> <tr> <td colspan="2"> <p>Daftar</p> </td> </tr> </table>	Register	X	<p>Register User Register Pengelola</p>		<p>Nama</p> <input type="text"/>		<p>Email</p> <input type="text"/>		<p>Username</p> <input type="text"/>		<p>Password</p> <input type="password"/>		<p>Ulangi Password</p> <input type="password"/>		<p>Kode Pengelola (Optional)</p> <input type="text"/>		<p>Daftar</p>	
SMART BATHROOM	Register Login																												
<h1>Selamat Datang di Aplikasi Smart Bathroom</h1> <p>Solusi Monitoring Kamar Mandi Pintar</p>																													
<h2>Apa itu Smartbathroom ?</h2> <p>Smart Bathroom adalah Solusi Monitoring sistem kamar mandi Pintar Anda</p>																													
<h2>Fitur Kami</h2> <p>Live Statistic Reporting Management</p>																													
Copyright @ 2019 All Right Reserved Created by Christhoper																													
Register	X																												
<p>Register User Register Pengelola</p>																													
<p>Nama</p> <input type="text"/>																													
<p>Email</p> <input type="text"/>																													
<p>Username</p> <input type="text"/>																													
<p>Password</p> <input type="password"/>																													
<p>Ulangi Password</p> <input type="password"/>																													
<p>Kode Pengelola (Optional)</p> <input type="text"/>																													
<p>Daftar</p>																													

Gambar 3. 17 Layout Login dan Register

Pada gambar 3.17 adalah *layout* untuk tampilan *login* dan *register*. Merupakan menu di tampilan awal *website* yang digunakan untuk membuat akun baru atau masuk ke *website*.

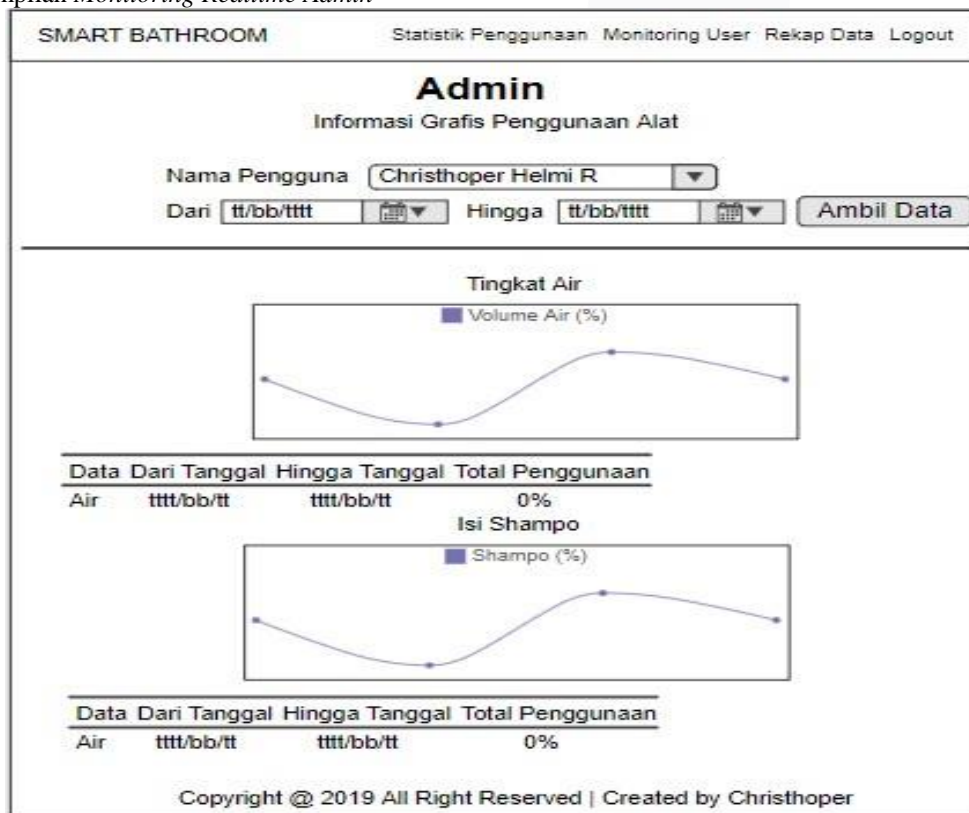
c. Halaman Admin

1. Dashboard Admin dan Statistik penggunaan

Gambar 3. 18 Layout Dashboard Admin

Pada gambar 3.18 merupakan *dashboard* admin dan tampilan statistik penggunaan. Di menu ini *admin* mendapatkan akses untuk melihat penggunaan *user* yang dikelola oleh *admin*. Data yang ditampilkan akan berupa grafik.

2. Tampilan Monitoring Realtime Admin



Gambar 3. 19 Layout Monitoring Real-time

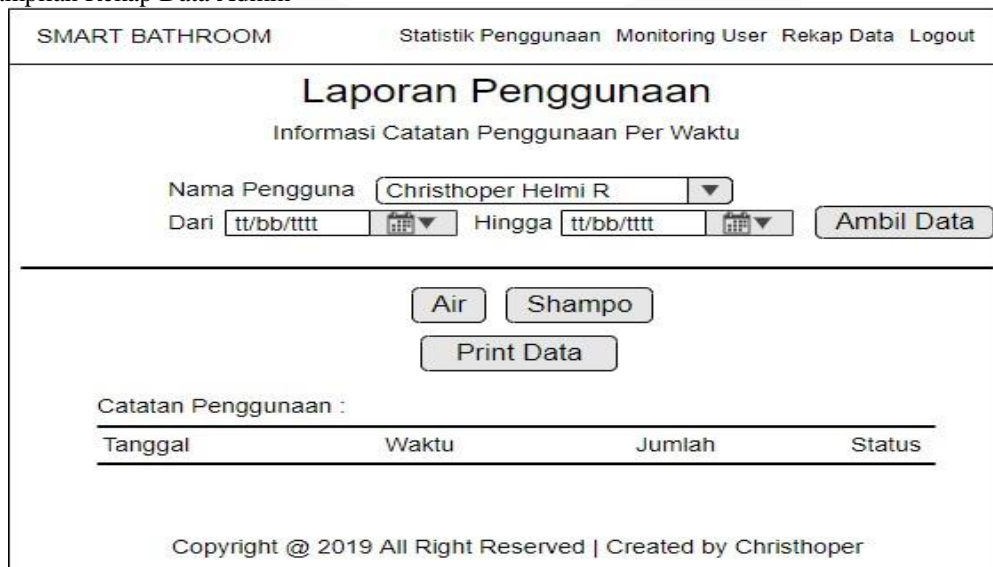
Pada gambar 3.19 diatas merupakan tampilan *monitoring realtime* untuk melihat kondisi kamar mandi *user* pada jangka waktu tertentu. Akan muncul grafik untuk tampilannya.



Gambar 3. 20 Layout Rekap Data

Pada gambar 3.20 diatas merupakan tampilan pengelolaan user yang berfungsi untuk admin mengetahui pengguna yang sudah terdaftar sebagai anggota pengelolaannya ataupun menerima request dari calon anggota.

3. Tampilan Rekap Data Admin



Gambar 3. 21 Layout Rekap Data

Pada gambar 3.21 diatas merupakan tampilan Rekap data untuk *admin*. Pada tampilan ini admin akan mendapatkan data dari *user* dalam jangka waktu tertentu berupa tabel seperti digambar berisi tentang rangkuman harian penggunaan *user*.

d. Halaman User

1. Halaman Dashboard User



Gambar 3. 22 Layout Dashboard User

Pada gambar diatas merupakan tampilan dashborad pada *user* untuk melakukan *monitoring* kamar mandinya.

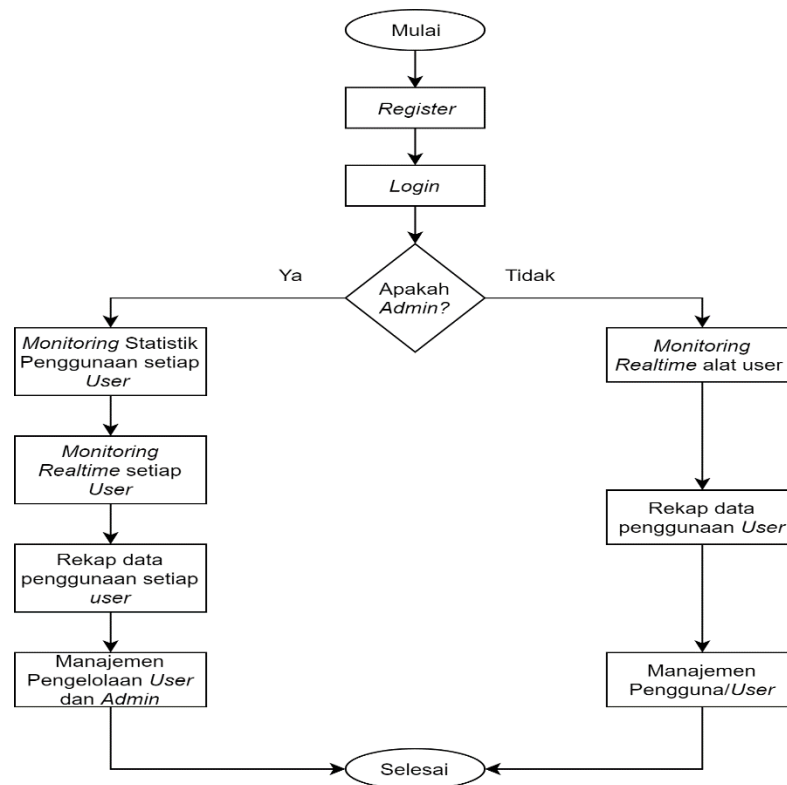
2. Laporan Penggunaan



Gambar 3. 23 Layout Laporan Penggunaan User

Pada gambar diatas merupakan tampilan Rekap data untuk *user*. Pada tampilan ini *user* akan mendapatkan informasi penggunaan dalam jangka waktu tertentu berupa tabel seperti digambar berisi tentang rangkuman harian penggunaan *user*.

3.1 Pembuatan Sistem



Gambar 3. 24 Flowchart Deskripsi Umum Sistem

Pada gambar 3.24 adalah *flowchart* deskripsi umum sistem dalam Proyek Akhir ini, Berikut dibawah ini merupakan penjelasan dari gambar 3.23:

1 Register

Pada tahap ini *admin* dan *user* diminta melakukan registrasi akun untuk membuat *username* dan *password* agar dapat mengakses halaman *website*. Pada halaman *register* akan dibagi menjadi dua bagian yaitu register sebagai *admin* dan register sebagai *user*

2 Login

Pada tahap ini, setelah *admin* dan *user* berhasil membuat akun untuk *login* maka secara otomatis sistem akan memverifikasi akun tersebut kedalam *database*, kemudian pada sisi *admin* dapat melakukan *monitoring* untuk mengetahui nilai volume air dan sampo yang di gunakan oleh *user*, dan juga mengetahui kondisi lampu dan *exhaust fan* yang menyala atau tidak.

3 Statistik Penggunaan Pada Sisi Admin

Pada tahap ini *admin* dapat mengetahui statistik penggunaan *user* yang terdaftar pada anggota *admin* melalui kode verifikasi yang diberikan oleh *admin* kepada *user*. Data tersebut akan ditampilkan dalam bentuk grafik yang akan memberikan informasi tentang statistik penggunaan harian *user*.

4 Monitoring Realtime Pada Sisi Admin

Pada tahap ini *admin* dapat melakukan pemantauan penggunaan alat terhadap *user* secara *realtime* pada hari itu.

5 Rekap Data Pada Sisi Admin

Pada tahap ini *admin* dapat melakukan rekap data *user* dalam jangka waktu yang dipilih dan akan ditampilkan oleh *website* dalam bentuk tabel yang merupakan hasil rangkuman penggunaan *user*.

6 *Monitoring Realtime User*

Pada tahap ini *user* dapat melakukan pemantauan secara *realtime*. Yang membedakan pada sisi admin hanya tampilan data, untuk tampilan di user menggunakan gambar dinamis yang akan mengikuti perubahan kondisi. Fitur yang bisa dipantau yaitu kondisi air dan sampo, lampu dan *exhaust fan*.

7 Laporan Penggunaan User

Pada tahap ini *user* dapat melihat penggunaan *user* dalam jangka waktu tertentu yang akan ditampilkan dalam bentuk tabel.

4. Pengujian Sistem dan Analisa Hasil

4.1 Pengujian Fungsionalitas

Pengujian fungsionalitas ini bertujuan untuk mengetahui program yang dibuat sesuai dengan rancangan awal yang sudah dibuat sebelumnya dengan cara membandingkan data yang diharapkan dengan data pengujian langsung. Pengujian fungsionalitas tersebut ditunjukkan pada tabel 4.1

Tabel 4. 1 Tabel Pengujian Fungsionalitas

No	Nama Pengujian	Aksi	Harapan Hasil	Hasil Pengujian	Hasil
1	<i>Login</i>	<i>Input Username, Password, Kode Pengelola</i>	Berhasil <i>Login</i>	Berhasil <i>Login</i>	Berhasil
2	<i>Register</i>	<i>Input Username, Password, Kode Pengelola</i>	Berhasil menambahkan di <i>database</i>	Berhasil menambahkan di <i>database</i>	Berhasil
3	<i>Monitoring</i>	Memasuki Halaman Statistik Penggunaan dan <i>Monitoring User</i> .	Menampilkan Informasi Singkat Setiap Perangkat	Menampilkan Informasi Singkat Setiap Perangkat	Berhasil
		Menampilkan grafik dan gambar ilustrasi air, shampo, lampu, dan kipas	Grafik dan Gambar Muncul Sesuai Data Pada Halaman <i>Dashboard</i>	Grafik dan Gambar Muncul Sesuai Data Pada Halaman <i>Dashboard</i>	Berhasil
		Menampilkan Informasi Data, Tanggal dan Total Penggunaan	Berhasil Menampilkan Informasi Data, Tanggal dan Total Penggunaan	Berhasil Menampilkan Informasi Data, Tanggal dan Total Penggunaan	Berhasil

4	Rekap Data	Menampilkan Laporan Penggunaan Berisi Penggunaan Air dan Shampo Berupa Tanggal, Waktu, Penggunaan, Jumlah Volume dan Status.	Menampilkan Laporan Penggunaan Berisi Penggunaan Air dan Shampo Berupa Tanggal, Waktu, Penggunaan, Jumlah Volume dan Status.	Berhasil Menampilkan Laporan Penggunaan Berisi Penggunaan Air dan Shampo Berupa Tanggal, Waktu, Penggunaan, Jumlah Volume dan Status.	Berhasil
5	Manajemen Pengelolaan Pengguna	Menampilkan daftar <i>user</i> yang telah terdaftar sebagai anggota pengelolaan <i>admin</i> , <i>Admin</i> menerima <i>request</i> dari <i>user</i> , <i>Admin</i> bisa menghapus perangkat.	Menampilkan daftar <i>user</i> yang telah terdaftar sebagai anggota pengelolaan <i>admin</i> , <i>Admin</i> menerima <i>request</i> dari <i>user</i> , <i>Admin</i> bisa menghapus perangkat.	Berhasil Menampilkan daftar <i>user</i> yang telah terdaftar sebagai anggota pengelolaan <i>admin</i> , <i>Admin</i> menerima <i>request</i> dari <i>user</i> , <i>Admin</i> bisa menghapus perangkat.	Berhasil

Pada tabel 4.1 dapat dilihat bahwa setiap pengujian fungsionalitas yang telah dilakukan dinyatakan berhasil dan memiliki rata-rata tingkat keberhasilan yang tinggi.

4.2 Pengujian Delay Firebase ke Website

Pengujian *Delay* dilakukan untuk mengetahui rata-rata waktu yang diperlukan untuk mengirim dan menerima data dalam satuan detik. Pengujian *delay* ini melakukan pengukuran dari firebase ke halaman *website*.

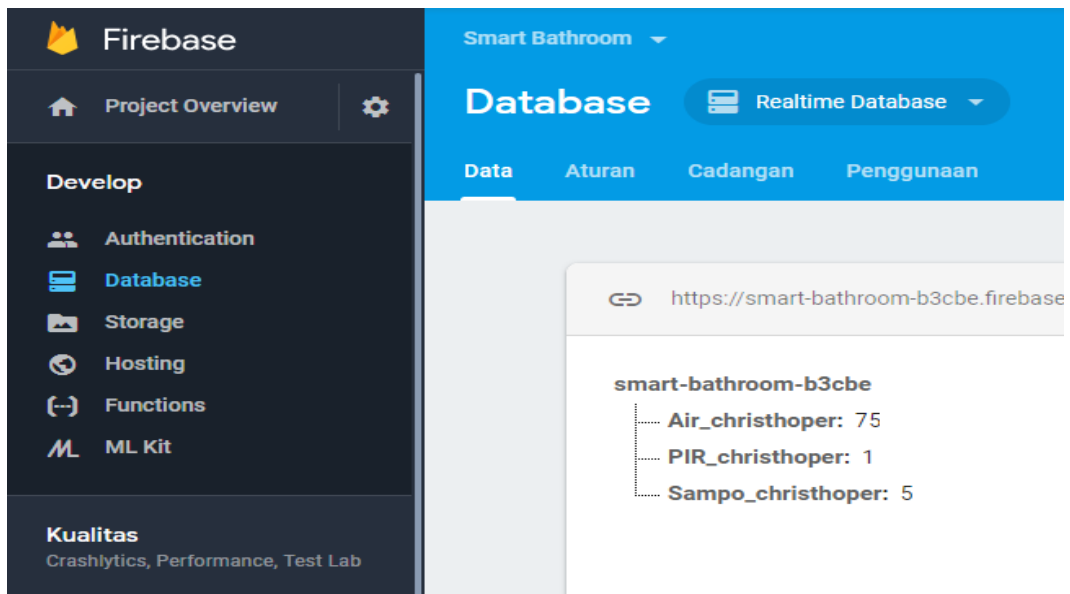
Tabel 4. 2 Tabel Pengujian Delay Firebase ke Database

<i>Admin</i>		<i>User</i>	
Pengujian ke-	<i>Delay</i> (detik)	Pengujian ke-	<i>Delay</i> (detik)
1	0.1	1	0.1
2	0.1	2	0.1
3	0	3	0.1
4	0.15	4	0
5	0.2	5	0.3
6	0	6	0
7	0.15	7	0.1
8	0.1	8	0.1
9	0.1	9	0.1
10	0.1	10	0.1
Rata-rata	0.1 detik	Rata-rata	0.1 detik

Pengujian pada tabel 4.2 dilakukan dengan cara *website* diakses oleh *admin* dan *user* yang berbeda untuk menerima data dari firebase ke halaman *website* dengan cara mengaktifkan dan menon-aktifkan perangkat atau alat kemudian didapatkan hasil *delay* yang berbeda dari *admin* dan *user*. *Delay* yang besar diakibatkan adanya proses dari sisi *website* ke *database* dan dari *database* ke alat.

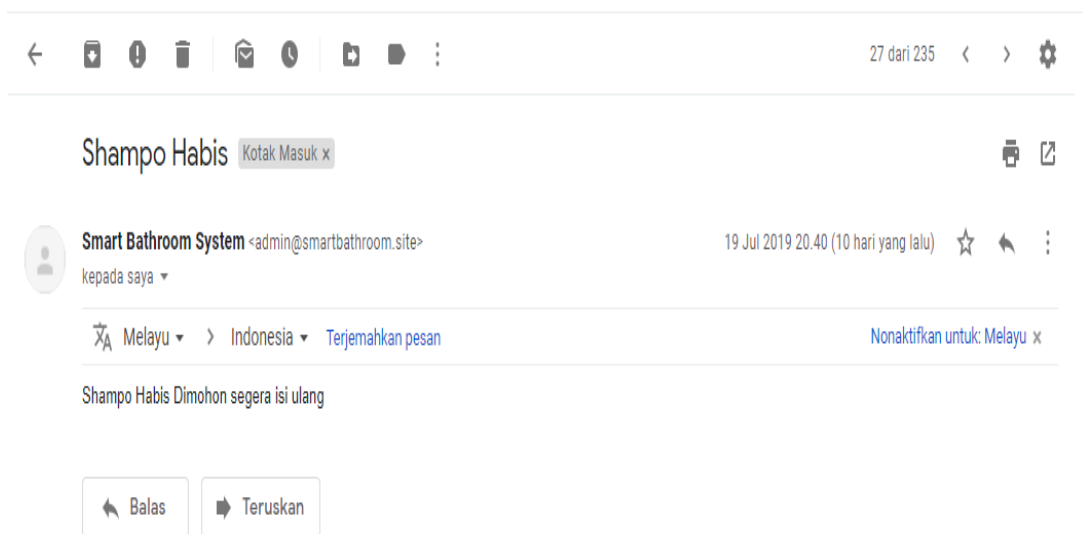
4.3 Pengujian Notifikasi Firebase ke Email

Pada tahap ini dilakukan pengujian notifikasi secara sistem melalui firebase yang dikirimkan melalui alamat *email* terhadap *user* yang telah didaftarkan sebelumnya pada laman *Register*. Dalam hal ini system akan mengirimkan notifikasi berisi tentang informasi penggunaan air dan shampo ke dalam *email user* yaitu christhope085r@gmail.com. Berikut dibawah ini merupakan hasil pengujian notifikasi firebase ke *email user*.



Gambar 4. 1 Data Pada Firebase

Pada Gambar 4.1 adalah isi data pada firebase, dalam hal Proyek Akhir ini pada smart-bathroom terdapat tiga (3) data *monitoring* yaitu Air, Lampu, *Fan* dan Shampo yang didapatkan melalui hasil yang pemantauan dari alat yang dipasang di kamar mandi. Kemudian data tersebut akan dikirimkan secara otomatis dari firebase ke alamat *email user* yang berfungsi memberikan informasi tentang penggunaan alat yang telah diintegrasikan.



Gambar 4. 2 Tampilan Notifikasi Email Dikirim Melalui Firebase

Pada Gambar 4.2 adalah tampilan notifikasi *email* yang dikirim melalui firebase yang berisi informasi tentang penggunaan alat (shampo). Notifikasi ini ditunjukkan untuk *user* yang berfungsi yaitu mengetahui seberapa banyak penggunaan alat (shampoo) pada kehidupan sehari-hari yang dilakukan oleh pengguna.

5. Kesimpulan

Berdasarkan tahapan pengujian fungsionalitas didapatkan hasil bahwa fitur aplikasi ini dapat berjalan dengan baik dan data yang ditampilkan sesuai dengan *database*. Berdasarkan data pengujian *delay* yang telah dilakukan, didapatkan nilai rata-rata *delay* sebesar 0,1 detik dan belum memenuhi standar *delay* ITU-T yaitu dibawah 100 ms. Berdasarkan tahapan pengujian notifikasi *email* yang dikirimkan secara otomatis oleh *server* disimpulkan bahwa *email* berhasil terkirim ke *user*. Berdasarkan survei *kuisionaire* yang telah dilakukan, di dapatkan kesimpulan bahwa *wesbsite* pemantau *smart bathroom* ini dibutuhkan oleh pemilik kamar mandi.

DAFTAR PUSTAKA

- [1] S. Kim, J. Y. Hong, S. Kim, S. H. Kim, J. H. Kim, and J. Chun, "Restful Design and Implementation of Smart Appliances for Smart Home," in Proceedings - 2014 IEEE International Conference on Ubiquitous Intelligence and Computing, 2014 IEEE International Conference on Autonomic and Trusted Computing, 2014 IEEE International Conference on Scalable Computing and Communications and Associated Sy, 2014.
- [2] T. Malche and P. Maheshwary, "Internet of Things (IoT) for building smart home system," in Proceedings of the International Conference on IoT in Social, Mobile, Analytics and Cloud, I-SMAC 2017, 2017.
- [3] W. Yan, Q. Wang, and Z. Gao, "Smart home implementation based on Internet and WiFi technology," in Chinese Control Conference, CCC, 2015.
- [4] Janssen, Cory. Internet of Things: IoT. Diakses dari situs techopedia pada 9 November 2013
- [5] Farr, Christina (February 13, 2013). "Firestore's scalable backend makes it '10 times easier' to build apps". *VentureBeat*. Diakses tanggal June 12,2014.
- [6] Marshall, Matt (August 29, 2013). "Firestore is building a Dropbox for developers". *VentureBeat*. Diakses tanggal June 12, 2014.
- [7] "Firestore Realtime Database". *Firestore, Inc*. Diakses tanggal May 19,2016.
- [8] Darrow, Barb (Dec 18, 2012). "Firestore secures its real-time back-end service". *Gigaom*. Diakses tanggal June 12, 2014.
- [9] "Firestore Realtime Database". Diakses tanggal Oktober 18, 2018.
- [10] Mahrus, A. A. (2010). *Web Design & Pemrograman Web*, (21), 1–36.
- [11] Riza, & Irsan. (2013). *Pengenalan HTML, CSS, dan Javascript*.
- [12] Systems, D. (2003). G.114 (05/2003). *Networks*.