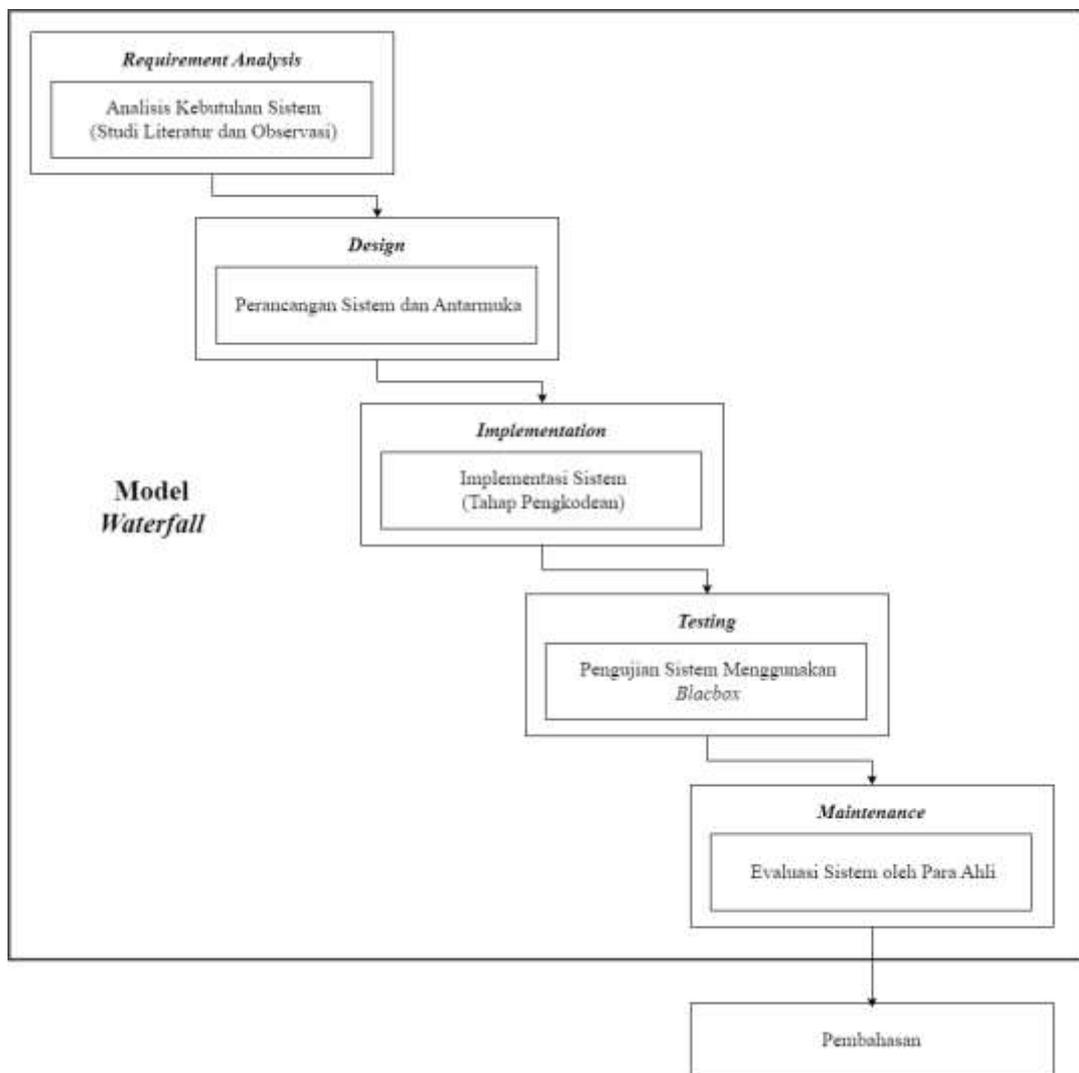


## BAB 3 METODE PENELITIAN

### 3.1. Desain Penelitian

Metode penelitian yang digunakan pada penelitian ini adalah metode kualitatif dengan cara studi literatur dan melakukan analisis terkait *website* tata surut Fakultas Pendidikan Matematika dan Ilmu Pengetahuan Alam (FPMIPA) Universitas Pendidikan Indonesia (UPI). Pada desain penelitian ini, peneliti menggambarkan rangkaian kegiatan yang dilakukan peneliti dalam melaksanakan penelitiannya. Desain penelitian berupa diagram yang menjelaskan alur penelitian.



Gambar 3. 1 Diagram Desain Penelitian

Gambar 3.1 merupakan diagram desain penelitian yang dilakukan peneliti dalam mengembangkan aplikasi. Model pengembangan sistem yang digunakan pada penelitian ini menggunakan model *waterfall*. Berikut adalah penjelasan tahap-tahap pada diagram desain penelitian:

1. Tahap awal yaitu menganalisis kebutuhan sistem dengan cara melakukan studi literatur dimana peneliti memperoleh data dari penelitian-penelitian dan literatur yang sudah dibuat sebelumnya kemudian mengidentifikasi, mengevaluasi, dan menginterpretasikan secara deskriptif dengan tujuan untuk mengetahui sistem *automatic notification* dalam konteks manajemen perkantoran khususnya dalam pengelolaan tata surat. Selanjutnya, melakukan observasi pada website tata surat Fakultas Pendidikan Matematika dan Ilmu Pengetahuan (FPMIPA) Universitas Pendidikan Indonesia untuk memperoleh data yang dibutuhkan dan mendukung data hasil dari studi literatur.
2. Setelah mengetahui kebutuhan yang diperlukan, tahap selanjutnya pembuatan perancangan sistem. Pada tahap pembuatan perancangan sistem langkah yang dilakukan yaitu membuat rancangan UML yang terdiri dari *use case diagram*, *activity diagram*, *sequence diagram*, dan *class diagram*.
3. Setelah dilakukan perancangan dan mengidentifikasi gambaran umum tentang perangkat lunak yang akan dibuat, pada tahap selanjutnya yaitu membuat implementasi/pengkodean model *sistem automatic notifications* dan simulasi dari sistem manajemen perkantoran khususnya tata surat masuk dan surat keluar. Tujuan dibuat aplikasi simulasi tata surat yaitu untuk mengetahui apakah model tersebut dapat berjalan dengan baik atau tidak.
4. Tahap selanjutnya yaitu pengujian sistem dengan menggunakan metode *blackbox*. Tujuan dilakukan pengujian sistem ini adalah untuk mengetahui seberapa baik fungsional dari sistem ini.
5. Hasil *prototype* digunakan dan dievaluasi dengan memberikan umpan balik yang digunakan untuk lebih menyempurnakan persyaratan atau kebutuhan. Hasil *prototype* tersebut, dipresentasikan kepada para ahli. Pada tahap ini, peneliti mendapatkan *feedback* untuk memperbaiki proses *coding* atau melanjutkan pada tahap selanjutnya.

6. Terakhir peneliti membuat pembahasan. Pembahasan yang dibuat berdasarkan penelitian.

### 3.1.1. Pencarian Literatur

Berikut akan dijelaskan mengenai pencarian literatur pada penelitian ini untuk membantu dalam mencari jurnal terkait.

#### 1. Kata Kunci

Hal yang terpenting dalam pencarian literatur dengan menggunakan *database* Google Scholar, kata kunci yang digunakan adalah “Sistem notifikasi otomatis dalam manajemen tata surat”.

#### 2. *Database* Pencarian

Bagian terpenting dari pencarian literatur, selain penggunaan kata kunci adalah *database*. *Database* pencarian yang digunakan oleh penelitian adalah artikel nasional dengan Google Scholar.

#### 3. Strategi Pencarian

Strategi dalam pencarian literatur ini adalah dengan menggunakan sistem operator logika, khususnya perintah yang digunakan di mesin pencari seperti penggunaan kata *AND* untuk menghasilkan artikel yang hanya berisi kata kunci tersebut, *OR* untuk memperluas jumlah hasil pencarian, dan *NOT* untuk mempersempit hasil pencarian di Google Scholar.

**Tabel 3. 1 Strategi pencarian studi literatur**

Mesin Pencarian	Kata Kunci Pencarian
Google Scholar.	Sistem notifikasi otomatis AND manajemen tata surat.

### 3.1.2. Kriteria Inklusi dan Eksklusi

Berikut merupakan kriteria inklusi dan eksklusi pada penelitian ini yang dapat dilihat pada Tabel 3.2.

**Tabel 3. 2 Kriteria Inklusi dan Eksklusi**

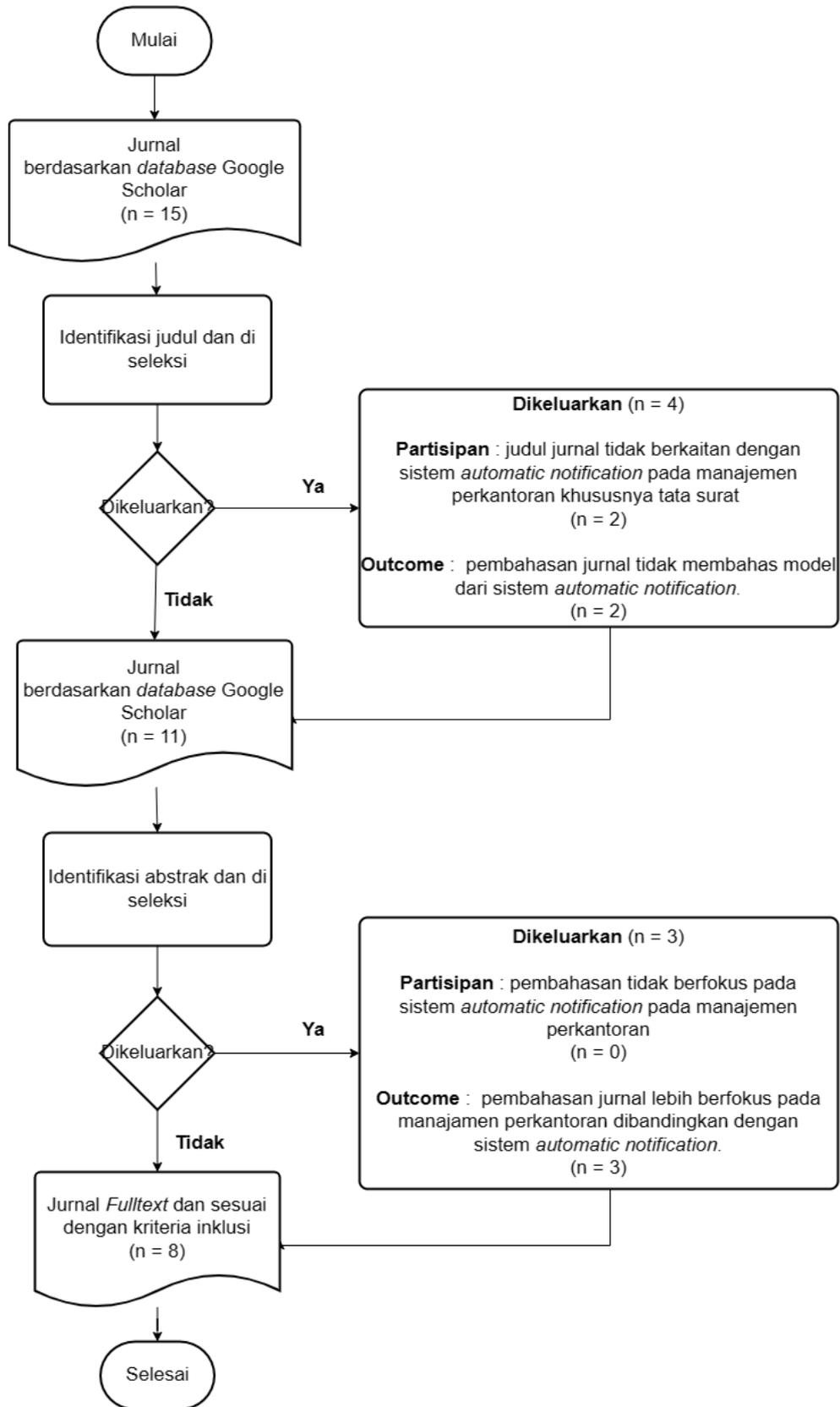
<b>Kriteria Inklusi</b>	<b>Kriteria Eksklusi</b>
Data yang diambil mengenai sistem informasi manajemen perkantoran khususnya tata surat.	Jurnal tidak mencantumkan bagaimana model notifikasi dari manajemen perkantoran.
Data yang diambil melibatkan sistem <i>automatic notification</i> .	Data yang ditampilkan di jurnal tidak lengkap.
Data yang digunakan tidak lebih dari 5 tahun terakhir.	Data yang diambil lebih berfokus pada manajemen perkantoran

### **3.1.3. Sintesis Hasil Literatur**

Dari beberapa jurnal penelitian yang sudah dikumpulkan, selanjutnya jurnal-jurnal tersebut diidentifikasi topik dan abstrak berdasarkan kriteria inklusi dan eksklusi, yang kemudian dibuatkan ringkasan. Untuk lebih memperjelas analisis abstrak dan *fulltext* jurnal dibaca dan dicermati, sehingga data jurnal tersebut di cari persamaan dan pebedaannya yang selanjutnya dibahas untuk menghasilkan kesimpulan. Berdasarkan hasil pencarian dengan kata kunci sistem notifikasi otomatis dalam konteks manajemen perkantoran:

- Google Scholar: 15 jurnal

Dari 15 jurnal yang sesuai dengan kata kunci. Kemudian dilakukan seleksi dengan mengacu pada kriteria eksklusi karena tidak sesuai dengan tema yang diangkat oleh peneliti. Dari hasil seleksi, di dapatkan sebanyak 11 jurnal dan kemudian diseleksi kembali sehingga didapatkan 8 jurnal *fulltext* yang akan dilakukan *review*.



**Gambar 3. 2 Diagram Flow Studi Literatur**

Berikut merupakan daftar jurnal yang memenuhi kriteria inklusi pada penelitian ini antara lain:

1. Perancangan notifikasi pesan disposisi pada sistem pengelolaan surat dinas menggunakan layanan API sosial media.
2. Sistem pengelolaan surat dengan memanfaatkan *Application Programming Interface* (API) untuk notifikasi pesan disposisi.
3. Integrasi *SMS Gateway* untuk Pengembangan Sistem Informasi Surat Padna Kantor Kepala Desa.
4. Pembangunan sistem aplikasi pelayanan administrasi desa berbasis *website* dengan fitur notifikasi *SMS Gateway*.
5. Implementasi *Web Push Notification* pada sistem informasi manajemen arsip menggunakan PushJS.
6. *Realtime notification* pada aplikasi berbasis website menggunakan *Firebase Cloud Messaging* (FCM).
7. Pengembangan aplikasi layanan notifikasi surat resmi digital di Universitas Negeri Jakarta.
8. Implementasi *Progressive Web Apps* pada sistem informasi disposisi surat dengan teknologi *Service Worker*.

#### **3.1.4. Alat Penelitian**

Alat penelitian berupa komputer yang akan diimplementasikan pada sistem *automatic notification* dalam konteks manajemen perkantoran adalah komputer dengan spesifikasi komputer *desktop* dan *mobile* pada umumnya. Namun, implementasi pada lingkungan kerja nyata ketika sistem telah benar-benar operasikan, tidak menutup kemungkinan spesifikasi komputer akan berubah mengikuti kebutuh sistem.

Penelitian ini menggunakan alat penelitian berupa perangkat keras dan perangkat lunak sebagai berikut:

1. Perangkat Keras
  - *Processor* Intel (R) Core (TM) i3-3217U CPU @ 1.80GHz (4 CPUs), ~1.8GHz,

- *Graphic card* Intel(R) HD Graphics 4000,
- Besar memori RAM 4096MB,
- Kapasitas Harddisk 1TB,
- Monitor dengan resolusi 1366 x 768 (60Hz),
- Perangkat mouse dan keyboard standar.

## 2. Perangkat Lunak

- Sistem operasi Windows 10 Home Edition 64-bit,
- Android 11 sebagai sistem operasi *mobile*.
- *Web browser* Google Chrome versi 96.0.4664.45 (64-bit),
- Visual Studio sebagai IDE dalam pembuatan model notifikasi dan simulasi manajemen surat masuk dan surat keluar,
- *Firebase Cloud Messaging* sebagai layanan *cross platform* untuk mengelola notifikasi.
- *Firebase Firestore* sebagai layanan untuk mengelola basis data sistem notifikasi manajemen tata surat.

### 3.1.5. Bahan Penelitian

Bahan penelitian yang akan digunakan oleh peneliti mencakup hasil observasi, hasil studi literatur, dan hasil evaluasi sistem *automatic notification* oleh para ahli.

## 3.2. Desain Sistem

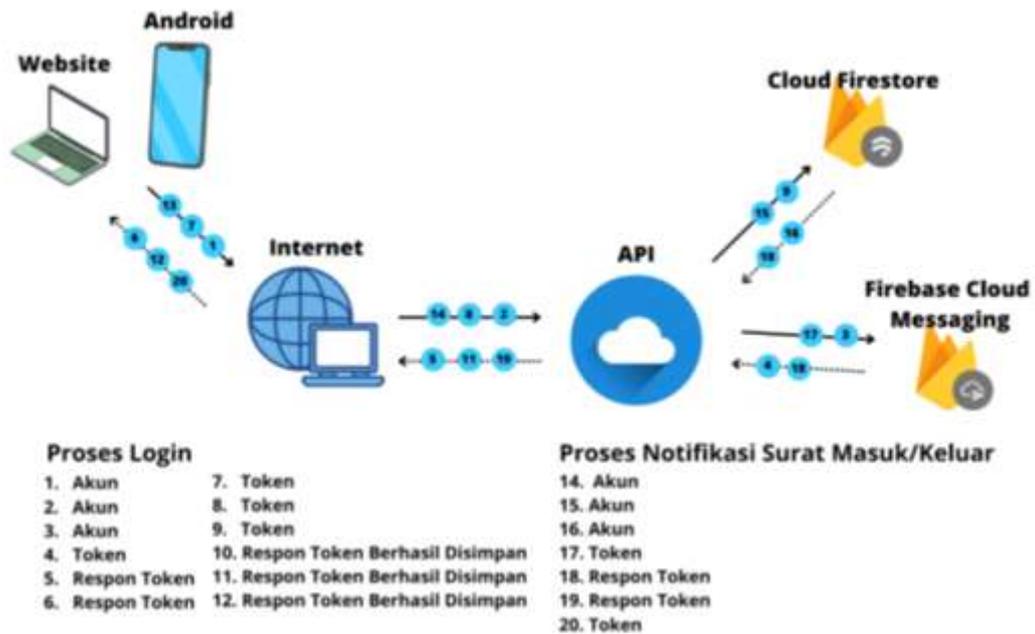
Pada Gambar 3.3, desain sistem berisi tentang arsitektur sistem yang akan digunakan dan diterapkan pada aplikasi. Aplikasi harus terhubung internet untuk mengakses model notifikasi melalui API. Pertama ketika aplikasi dijalankan, pengguna harus melakukan proses login terlebih dahulu untuk mendapatkan hak akses melalui API. API akan mengirimkan akun tersebut untuk didaftarkan oleh FCM menjadi sebuah token yang bersifat unik. Lalu API akan mengirimkan respon kepada aplikasi berupa token yang sudah di daftarkan oleh FCM. Token tersebut berfungsi untuk mengirim dan menerima notifikasi. Aplikasi akan memanggil API kembali untuk menyimpan token tersebut ke dalam *database* Firestore. Pada API terdapat kondisi dimana jika level pengguna dari aplikasi adalah Pengusul maka token tersebut akan disimpan pada dokumen dengan jenis surat yaitu surat keluar.

Deffin Achmaddifa, 2023

SISTEM *AUTOMATIC NOTIFICATION* DALAM KONTEKS MANAJEMEN PERKANTORAN

Universitas Pendidikan Indonesia | repository.upi.edu | perpeustakaan.upi.edu

Namun jika level pengguna dari aplikasi adalah Admin maka token tersebut akan disimpan pada dokumen dengan jenis surat yaitu surat masuk. Kemudian, pada API juga terdapat kondisi jika pengguna mengakses aplikasi melalui *smartphone* maka token tersebut akan disimpan dengan kunci yaitu mobile. Jika pengguna mengakses aplikasi melalui *website* maka token tersebut akan disimpan dengan kunci yaitu web. API akan memberikan respon kepada aplikasi berupa pesan bahwa token tersebut berhasil disimpan.



**Gambar 3. 3 Desain Sistem *Automatic Notification***

Pada proses pengiriman dan menerima notifikasi dalam konteks surat masuk, Pengusul harus melakukan penginputan data surat masuk terlebih dahulu. Dalam konteks surat keluar, Admin harus melakukan penginputan data surat keluar terlebih dahulu. Jika sudah melakukan penginputan data, aplikasi akan memanggil API untuk menyimpan data notifikasi ke dalam *database* lalu mengirimkan data notifikasi kepada aplikasi pengguna. Aplikasi akan memanggil API dengan *request body* yang terdapat pada Gambar 3.4.



```
{
  "appName": "TataSurat",
  "title": "Penelitian",
  "body": "Surat keluar tentang penelitian",
  "sender": "azisfaisal@gmail.com",
  "to": "deffinjr890@gmail.com",
  "route": "/detailSuratKeluar/SRM001"
}
```

**Gambar 3. 4 Request Body Data Notifikasi**

Pada API terdapat kondisi dimana jika *request body* dengan kunci “to” yaitu kosong atau tidak diisi maka API akan mengambil semua data token Admin dari *database*. Namun, jika *request body* dengan kunci “to” diisi maka API akan mengambil data token berdasarkan *request body* dengan kunci “to”. Kemudian, API akan mengirimkan notifikasi tersebut kepada FCM dan FCM akan mengirimkan pesan ke aplikasi yang sudah terdaftar pada *database* sehingga pengguna dapat melihat notifikasi tersebut di *smartphone* maupun *website*. Jika pengguna sedang tidak mengakes aplikasi, notifikasi tersebut akan diterima pada *background message* atau latar belakang.

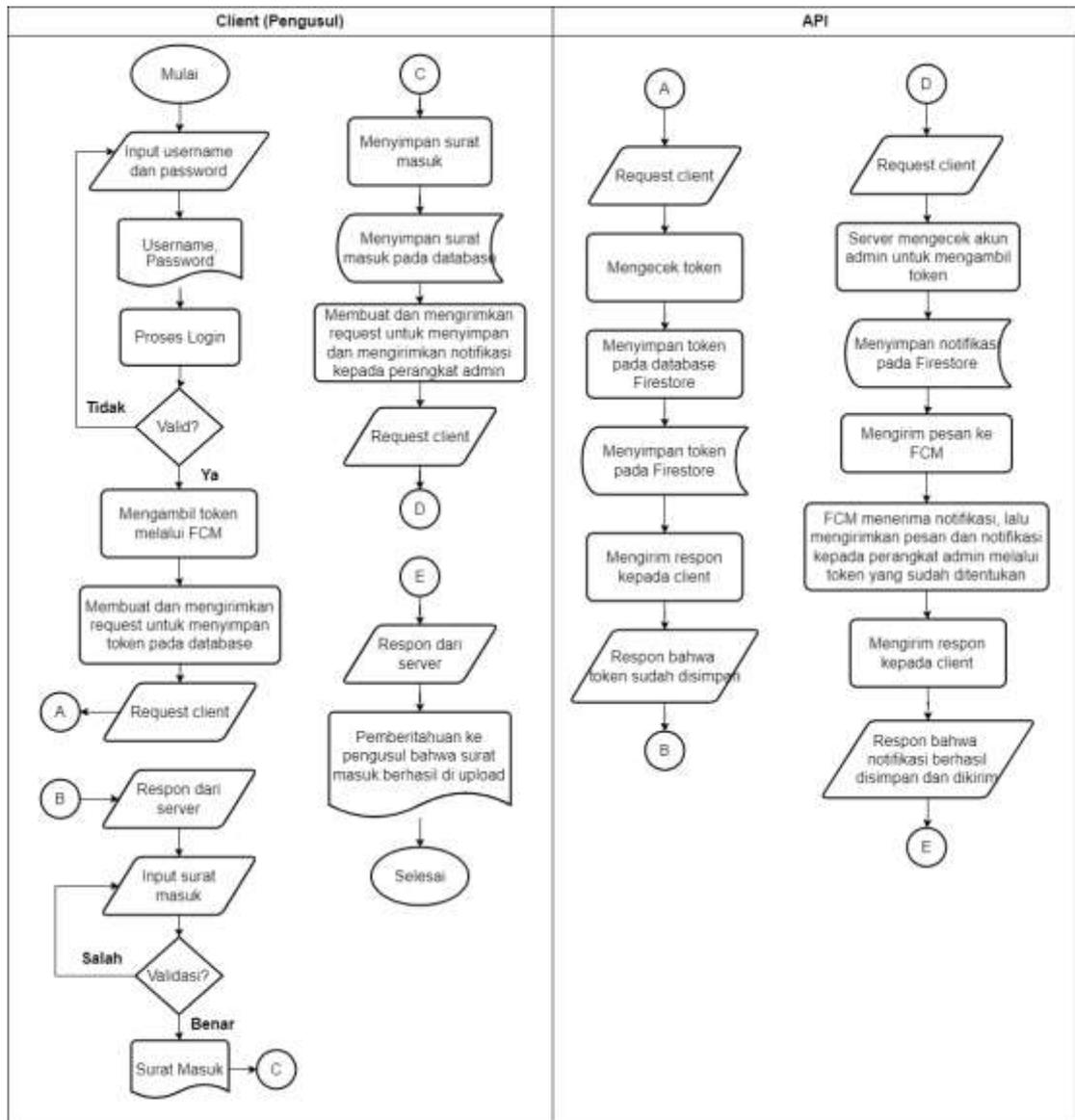
### **3.3. Diagram Alir**

Pada Gambar 3.5, diagram alir Pengusul dalam aplikasi ini untuk surat masuk sebagaimana penjelasan berikut ini:

1. Dimulai ketika Pengusul mengakses aplikasi ini, Pengusul diminta memasukan email dan password untuk melakukan proses login.
2. Sistem melakukan pengecekan apakah akun Pengusul tersebut sudah terdaftar atau belum.
3. Jika sudah terdaftar, sistem akan mengirimkan akun ke FCM untuk melakukan pendaftaran dan menghasilkan sebuah token yang bersifat unik. Sehingga akun tersebut dapat menerima *broadcast message*. Lalu aplikasi *client* akan membuat dan mengirimkan *request* untuk menyimpan token pada *database*.
4. *Server* akan menerima *request* dari aplikasi *client*, selanjutnya *server* akan mengecek token yang dikirimkan oleh aplikasi *client*. Jika level pengguna

Pengusul, maka token tersebut akan disimpan pada jenis surat yaitu surat keluar. Namun jika level pengguna Admin, maka token tersebut akan disimpan pada jenis surat yaitu surat masuk. *Server* juga akan mengecek token, lalu menyimpan token pada *database* berdasarkan *platform* yang digunakan oleh pengguna.

5. *Server* akan mengirimkan respon kepada aplikasi *client* bahwa token tersebut berhasil disimpan.
6. Aplikasi *client* akan menerima respon dari *server*.
7. Selanjutnya, pengusul dapat membuat pengajuan surat masuk.
8. Sistem akan memvalidasi apakah data surat masuk tersebut sudah benar atau tidak.
9. Jika benar, sistem akan menyimpan surat masuk tersebut ke dalam *database*. Jika salah, maka Pengusul harus melakukan penginputan ulang surat masuk.
10. Aplikasi *client* akan memanggil *server* kembali dengan membuat *request* untuk menyimpan dan mengirimkan notifikasi kepada semua perangkat Admin.
11. *Server* akan menerima *request* dari *client*.
12. *Server* akan mengecek semua akun Admin untuk mengambil token yang ada pada *database*.
13. Selanjutnya, *server* akan menyimpan notifikasi berdasarkan surat masuk tersebut ke dalam *database* dan mengirimkan notifikasi melalui FCM dengan token yang sudah ditentukan sebelumnya.
14. FCM akan menerima notifikasi, lalu mengirimkan notifikasi tersebut kepada semua perangkat Admin.
15. *Server* akan mengirimkan respon kepada aplikasi *client* bahwa notifikasi berhasil disimpan dan dikirim.
16. Aplikasi *client* akan menerima respon dari *server* dan menampilkan pemberitahuan kepada Pengusul bahwa surat masuk berhasil disimpan dan dikirimkan.



**Gambar 3.5 Diagram alir Pengusul Untuk Surat Masuk**

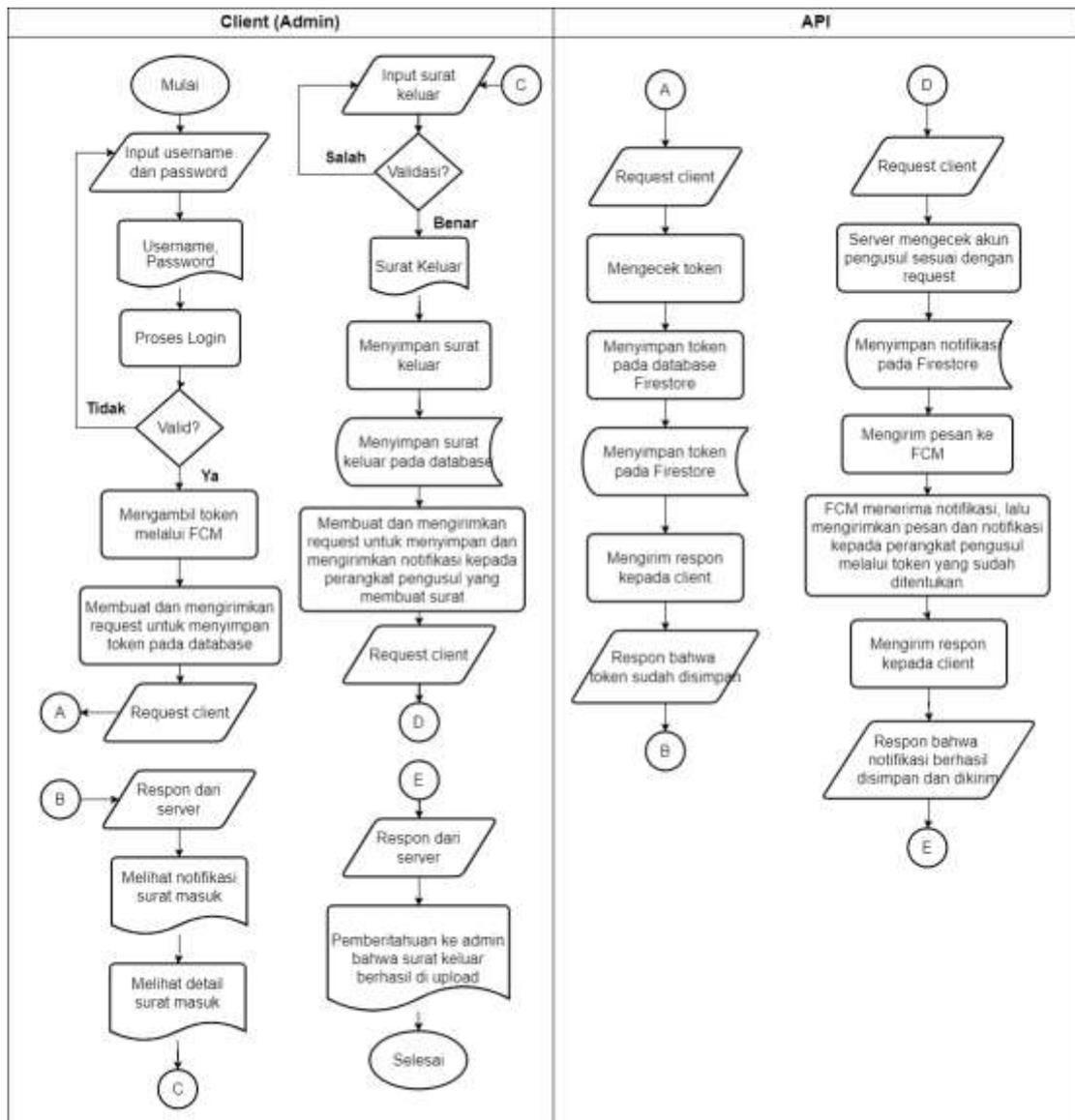
Pada Gambar 3.6, diagram alir Admin dalam aplikasi ini untuk surat keluar sebagaimana penjelasan berikut ini:

1. Dimulai ketika Admin mengakses aplikasi ini, Admin diminta memasukan email dan password untuk melakukan proses login.
2. Sistem melakukan pengecekan apakah akun Admin tersebut sudah terdaftar atau belum.
3. Jika sudah terdaftar, sistem akan mengirimkan akun ke FCM untuk melakukan pendaftaran dan menghasilkan sebuah token yang bersifat unik. Sehingga akun

tersebut dapat menerima *broadcast message*. Lalu aplikasi *client* akan membuat dan mengirimkan *request* untuk menyimpan token pada *database*.

4. *Server* akan menerima *request* dari aplikasi *client*, selanjutnya *server* akan mengecek token yang dikirimkan oleh aplikasi *client*. Jika level pengguna Pengusul, maka token tersebut akan disimpan pada jenis surat yaitu surat keluar. Namun jika level pengguna Admin, maka token tersebut akan disimpan pada jenis surat yaitu surat masuk. *Server* juga akan mengecek token, lalu menyimpan token pada *database* berdasarkan *platform* yang digunakan oleh pengguna.
5. *Server* akan mengirimkan respon kepada aplikasi *client* bahwa token tersebut berhasil disimpan.
6. Aplikasi *client* akan menerima respon dari *server*.
7. Admin dapat melihat notifikasi surat masuk yang diajukan oleh Pengusul. Ketika Admin menekan notifikasi tersebut, sistem akan menampilkan detail dari surat masuk.
8. Selanjutnya, Admin dapat membuat surat keluar.
9. Sistem akan memvalidasi apakah data surat keluar tersebut sudah benar atau tidak.
10. Jika benar, sistem akan menyimpan surat keluar tersebut ke dalam *database*. Jika salah, maka Admin harus melakukan penginputan ulang surat keluar.
11. Aplikasi *client* akan memanggil *server* kembali dengan membuat *request* untuk menyimpan dan mengirimkan notifikasi kepada perangkat Pengusul yang membuat surat masuk.
12. *Server* akan menerima *request* dari *client*.
13. *Server* akan mengecek akun Pengusul yang membuat surat masuk berdasarkan *request* dari *client* untuk mengambil token yang ada pada *database*.
14. Selanjutnya, *server* akan menyimpan notifikasi berdasarkan surat keluar tersebut ke dalam *database* dan mengirimkan notifikasi melalui FCM dengan token yang sudah ditentukan sebelumnya.
15. FCM akan menerima notifikasi, lalu mengirimkan notifikasi tersebut kepada perangkat Pengusul.

16. *Server* akan mengirimkan respon kepada aplikasi *client* bahwa notifikasi berhasil disimpan dan dikirim.
17. Aplikasi *client* akan menerima respon dari *server* dan menampilkan pemberitahuan kepada Admin bahwa surat keluar berhasil disimpan dan dikirimkan.



**Gambar 3. 6 Diagram Alir Admin Untuk Surat Keluar**

### 3.4. Teknik Pengujian Sistem

Pengujian software sangat diperlukan untuk memastikan software yang sudah ada dapat berjalan sesuai dengan fungsionalitas yang diharapkan (Hidayat &

Muttaqin, 2020). Pengujian software untuk melakukan verifikasi dan validasi bahwa *software* telah dibuat sesuai dengan kebutuhan perusahaan. Menurut (Jaya et al., 2019), pengujian merupakan bagian dari tahapan yang tidak dapat dipisahkan dari siklus hidup pengembangan software seperti halnya analisis, desain, dan pengkodean.

Dalam penelitian ini, peneliti menggunakan metode pengujian *black box* untuk mengetahui seberapa baik fungsional dari sistem ini. Menurut (Nursaid et al., 2020), pengujian *black box* merupakan sebuah metode yang digunakan untuk menguji sebuah software tanpa harus memperhatikan detail *software* dan hanya memeriksa nilai *output* berdasarkan nilai input masing-masing.

Selain itu, metode pengujian dilakukan dengan wawancara terhadap para ahli. Tabel 3.3 adalah pertanyaan yang digunakan sebagai wawancara kepada para ahli untuk tahap evaluasi pada sistem *automatic notification* dalam konteks manajemen perkantoran. Hasil wawancara para ahli dianalisa untuk menentukan apakah model notifikasi ini dapat berjalan dengan baik dan bermanfaat bagi pengguna.

**Tabel 3. 3 Pertanyaan Wawancara Evaluasi Sistem Oleh Para Ahli**

No	Pertanyaan
1	Bagaimana alur ketika notifikasi pada aplikasi sedang dibuka?
2	Bagaimana alur ketika notifikasi pada aplikasi sedang tidak dibuka?
3	Apakah menurut anda manajemen notifikasi dapat dilihat berdasarkan hak akses pengguna?
4	Apakah menurut anda manajemen notifikasi dapat menampilkan daftar dari notifikasi?
5	Bagaimana alur ketika menyaring/memfilter data notifikasi?
6	Apakah menurut anda sistem dapat menampilkan daftar notifikasi berdasarkan filter?
7	Bagaimana alur untuk mengubah status notifikasi?
8	Apakah menurut anda sistem notifikasi ini bermanfaat pada manajemen perkantoran khususnya tata surat masuk/keluar?