

PROYEK
TUGAS AKHIR



POLITEKNIK NEGERI BALI

**PENGEMBANGAN APLIKASI *EMAIL SERVICE* BERBASIS
WEB MENGGUNAKAN RABBITMQ DAN ELASTIC STACK
UNTUK PENGIRIMAN *EMAIL* TERSENTRALISASI PADA
PT. BANK PEMBANGUNAN DAERAH BALI**

OLEH :

**[I PUTU DIO MAHARDIKA / 2215323086]
[I PUTU AGUS SENIARTAWAN / 2215323039]
[DIKA PUTRA SUSANTO / 2215323019]**

**PROGRAM STUDI MANAJEMEN INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI BALI
2025**

ABSTRAK

Di era digital saat ini, pengelolaan komunikasi melalui *email* menjadi salah satu kebutuhan penting, khususnya dalam sektor perbankan yang menuntut layanan informasi yang cepat, andal, dan terstruktur. Proyek tugas akhir ini bertujuan untuk mengembangkan aplikasi *email service* berbasis *web* yang tersentralisasi menggunakan teknologi RabbitMQ dan Elastic Stack. Sistem ini dirancang untuk memenuhi kebutuhan PT. Bank Pembangunan Daerah Bali dalam mengelola pengiriman *email* secara efisien, dengan dukungan antrian pesan asinkron dan pemantauan *log* yang *real-time*. *Framework* Laravel digunakan sebagai kerangka utama pengembangan *backend*, sedangkan *frontend* dikembangkan dengan Vue.js dan Tailwind CSS untuk antarmuka pengguna yang responsif. Proyek ini mencakup proses analisis kebutuhan, perancangan sistem menggunakan pendekatan UML, implementasi fitur manajemen aplikasi dan *email*, serta pengujian fungsional menggunakan metode *black box testing*. Hasil akhir dari proyek ini adalah sistem pengiriman *email* yang mampu menangani pesan dalam jumlah besar, meminimalisasi risiko kegagalan pengiriman, serta menyediakan visibilitas penuh terhadap aktivitas *email* melalui *dashboard monitoring*. Sistem ini diharapkan dapat meningkatkan efisiensi operasional dan kualitas komunikasi internal maupun eksternal di lingkungan perbankan.

Kata kunci: *Email Service, RabbitMQ, Elastic Stack, Laravel, Pengiriman Email Tersentralisasi.*

DAFTAR ISI

LEMBAR PENGESAHAN PROYEK TUGAS AKHIR	ii
LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI.....	iii
FORM PERNYATAAN BEBAS PLAGIARISME	iv
KATA PENGANTAR	v
ABSTRAK.....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	ix
DAFTAR GAMBAR	x
BAB I INFORMASI UMUM PROYEK	1
1.1 Informasi Global Proyek.....	1
1.2 Latar Belakang	1
1.3 Deskripsi Proyek/ Gambaran Umum Proyek.....	3
1.3.1 Menu dan fitur yang akan dihadirkan pada proyek.....	3
1.3.2 <i>Flowmap</i> Sistem <i>Email Service</i>	6
1.4 Tujuan Proyek	15
1.5 Keuntungan Proyek.....	16
BAB II PERENCANAAN PROYEK	17
2.1 Teknologi Digunakan.....	17
2.2 Pembagian Tugas dan Pelaksanaan.....	18
2.3 Perancangan Proyek	19
2.3.1 Perancangan Sistem (Rancang Bangun Sistem)	19
2.4 Anggaran Biaya.....	58
BAB III PELAKSANAAN PROYEK.....	59
3.1 Hasil Proyek Aplikasi	59
3.1.1 Hasil Tampilan <i>Website</i>	59
3.1.2 Hasil Manajemen <i>Log</i> Menggunakan Elastic Stack	72
3.2 Implementasi Proyek.....	72
BAB IV PENUTUP	83
4.1 Kesimpulan	83
4.2 Saran.....	83
DAFTAR PUSTAKA	xiii
LAMPIRAN.....	xiv

DAFTAR TABEL

Tabel 1.1 Tabel Informasi Proyek.....	1
Tabel 2.1 Pembagian Tugas	18
Tabel 3.1 Hasil Pengujian	73

DAFTAR GAMBAR

Gambar 1.1 <i>Flowmap Login</i>	6
Gambar 1.2 <i>Flowmap Edit Profil Admin atau Supervisor</i>	7
Gambar 1.3 <i>Flowmap Edit Password</i>	8
Gambar 1.4 <i>Flowmap Pendaftaran Aplikasi</i>	9
Gambar 1.5 <i>Flowmap Pengiriman Email</i>	10
Gambar 1.6 <i>Flowmap Regenerate Secret Key</i>	11
Gambar 1.7 <i>Flowmap Pengiriman Ulang Email</i>	13
Gambar 1.8 Ubah Status Aplikasi	15
Gambar 2.1 <i>Use Case Diagram Admin</i>	20
Gambar 2.2 <i>Use Case Diagram Supervisor</i>	21
Gambar 2.3 <i>Activity Diagram Login</i>	23
Gambar 2.4 <i>Activity Diagram Edit Profil</i>	24
Gambar 2.5 <i>Activity Diagram Edit Password</i>	25
Gambar 2.6 <i>Activity Diagram Pendaftaran Aplikasi</i>	26
Gambar 2.7 <i>Activity Diagram Pengiriman Email</i>	27
Gambar 2.8 <i>Activity Diagram Ubah Status Aplikasi</i>	28
Gambar 2.9 <i>Activity Diagram Regenerate Secret Key</i>	29
Gambar 2.10 <i>Activity Diagram Pengiriman Ulang Email</i>	31
Gambar 2.11 <i>Class Diagram Sistem Email Service</i>	33
Gambar 2.12 <i>EERD Sistem Email Service</i>	34
Gambar 2.13 <i>Sequence Diagram Login</i>	35
Gambar 2.14 <i>Sequence Diagram Registrasi Aplikasi</i>	37
Gambar 2.15 <i>Sequence Diagram Ubah Status</i>	39
Gambar 2.16 <i>Sequence Diagram Regenerate Secret Key</i>	40
Gambar 2.17 <i>Sequence Diagram Kirim Email</i>	42
Gambar 2.18 <i>Sequence Diagram Kirim Ulang Email</i>	44
Gambar 2.19 <i>Sequence Diagram Ubah Informasi Pribadi</i>	46
Gambar 2.20 <i>Sequence Diagram Ubah Password</i>	47
Gambar 2.21 <i>Rancangan Sistem Manajemen Log</i>	48
Gambar 2.22 <i>Desain UI Login</i>	51

Gambar 2.23 Desain <i>UI Dashboard</i>	51
Gambar 2.24 Desain <i>UI Manajemen Email</i>	52
Gambar 2.25 Desain <i>UI Modal Kirim Email</i>	53
Gambar 2.26 Desain <i>UI Modal Input Pesan</i>	53
Gambar 2.27 Desain <i>UI Modal Form Input Pesan</i>	54
Gambar 2.28 Desain <i>UI Modal Upload Excel</i>	54
Gambar 2.29 Desain <i>UI Manajemen Aplikasi</i>	55
Gambar 2.30 Desain <i>UI Modal Tambah Aplikasi</i>	56
Gambar 2.31 Desain <i>UI Halaman Approval</i>	56
Gambar 2.32 Desain <i>UI Update Data Profil</i>	57
Gambar 2.33 Desain <i>UI Update Password</i>	57
Gambar 3.1 Halaman <i>Login</i>	59
Gambar 3.2 <i>Dashboard</i> Pengguna (Admin)	60
Gambar 3.3 <i>Dashboard</i> Pengguna (Supervisor)	60
Gambar 3.4 Halaman <i>Manajemen Email</i>	61
Gambar 3.5 Modal Pilihan Metode Pengiriman <i>Email</i>	61
Gambar 3.6 Modal Pilihan Metode Pengiriman <i>Email</i>	62
Gambar 3.7 Form Tambah <i>Email</i> Baru.....	62
Gambar 3.8 Form <i>Upload</i> File Excel	63
Gambar 3.9 Tampilan Detail <i>Email</i>	63
Gambar 3.10 Modal Kirim Ulang <i>Email</i>	64
Gambar 3.11 Modal Konfirmasi Hapus Satu <i>Log Email</i>	64
Gambar 3.12 Modal Konfirmasi Hapus Banyak <i>Log Email</i>	65
Gambar 3.13 Halaman <i>Manajemen Aplikasi</i>	65
Gambar 3.14 Form Tambah Aplikasi Baru.....	66
Gambar 3.15 Tampilan Detail Aplikasi	66
Gambar 3.16 Modal <i>Regenerate Secret Key</i>	67
Gambar 3.17 Modal Ubah Status Aplikasi	67
Gambar 3.18 Modal Konfirmasi Hapus Satu Aplikasi	68
Gambar 3.19 Modal Konfirmasi Hapus Banyak Aplikasi	68
Gambar 3.20 Halaman <i>Approve</i> Aplikasi	69
Gambar 3.21 Modal Konfirmasi <i>Approve</i>	69

Gambar 3.22 Modal Konfirmasi <i>Reject</i>	70
Gambar 3.23 Form Edit Informasi Pribadi	70
Gambar 3.24 Form Ubah <i>Password</i>	71
Gambar 3.25 Gambar Tombol <i>Logout</i>	71
Gambar 3.26 Hasil Manajemen <i>Log</i> Dengan Elastic Stack	72

BAB I

INFORMASI UMUM PROYEK

1.1 Informasi Global Proyek

Proyek ini merupakan pengembangan aplikasi *email service* berbasis *web* dengan memanfaatkan teknologi *message broker* RabbitMQ serta sistem pengelolaan *log* menggunakan Elastic Stack di PT. Bank Pembangunan Daerah Bali. Pengembangan *email service* ini bertujuan untuk menciptakan sistem terpusat yang mampu mengelola dan menyimpan proses pengiriman *email* secara efektif. Sistem ini dirancang agar dapat diintegrasikan dengan berbagai aplikasi milik PT. Bank Pembangunan Daerah Bali yang memerlukan *email service*, sehingga setiap aktivitas pengiriman *email* dapat dipantau dan dikelola secara lebih optimal.

Tabel 1.1 Tabel Informasi Proyek

Jenis Proyek	Proyek Dari Tempat PKL / Penugasan Dari Prodi
Pengerjaan Proyek	Kelompok / Individu
Pemilik Proyek	PT. Bank Pembangunan Daerah Bali
Manajer Proyek	I Gusti Ngurah Ary Widiatmika, ST., MM.
Ketua Tim Proyek	I Gede Udayana Putra, S.TI
Anggota Proyek	● I Putu Dio Mahardika ● I Putu Agus Seniartawan ● Dika Putra Susanto

1.2 Latar Belakang

Komunikasi melalui email adalah kebutuhan mendasar di berbagai sektor industri, termasuk perbankan. Sebagai institusi yang menangani transaksi keuangan, bank memerlukan sistem komunikasi yang cepat, andal, dan aman guna memastikan layanan optimal bagi nasabah serta kelancaran operasional internal. *Email* tidak hanya berperan sebagai sarana komunikasi antara bank dan nasabah, tetapi juga digunakan untuk keperluan administratif, transaksi keuangan, serta

penyampaian pemberitahuan resmi. Oleh karena itu, efektivitas sistem *email* memiliki dampak yang signifikan terhadap kinerja suatu institusi perbankan.

Sejalan dengan meningkatnya digitalisasi dalam layanan perbankan, jumlah *email* yang dikirim dan diterima setiap hari oleh bank semakin bertambah. PT. Bank Pembangunan Daerah Bali, sebagai salah satu lembaga keuangan yang terus berkembang, menghadapi tantangan dalam mengelola *email* secara efisien dan terstruktur. Sistem pengiriman *email* yang digunakan saat ini masih memiliki keterbatasan, seperti penggunaan metode tradisional untuk pengiriman *email* ke masing-masing nasabah. Penggunaan metode tersebut berpengaruh terhadap kecepatan pemrosesan yang tidak optimal, keterbatasan dalam pemantauan status pengiriman, serta kurangnya persiapan dalam menangani pengiriman email dalam jumlah yang besar. Hal ini dapat menimbulkan berbagai permasalahan, seperti keterlambatan penyampaian informasi penting kepada nasabah, kegagalan pengiriman email akibat kendala sistem, serta kesulitan dalam melakukan audit dan pelacakan aktivitas pengiriman *email*.

Untuk mengatasi kendala tersebut, diperlukan sistem yang mampu menangani pengiriman email secara terpusat dengan performa yang lebih optimal. Salah satu solusi yang dapat diterapkan adalah pemanfaatan RabbitMQ sebagai *message broker* guna mengelola antrian pengiriman *email*, serta Elastic Stack untuk pemantauan dan analisis *log* sistem *email*. RabbitMQ memungkinkan distribusi pesan yang lebih efisien dan dapat mengurangi risiko kehilangan pesan saat terjadi lonjakan permintaan. Sementara itu, Elastic Stack berperan dalam memantau performa sistem dengan menyajikan data log yang dapat dianalisis guna meningkatkan keandalan layanan. Kombinasi kedua teknologi ini memungkinkan sistem *email* beroperasi secara lebih efisien dan dapat diandalkan dalam mendukung kebutuhan komunikasi pada bidang perbankan.

Berdasarkan permasalahan yang telah diuraikan, aplikasi *email service* berbasis *web* dengan menggunakan RabbitMQ dan Elastic Stack ini dapat menjadi solusi, guna menciptakan sistem pengiriman *email* yang terpusat untuk PT. Bank Pembangunan Daerah Bali. Implementasi sistem ini diharapkan mampu meningkatkan efisiensi dan efektivitas dalam pengelolaan *email*, mengurangi risiko kegagalan pengiriman, serta memberikan visibilitas yang lebih jelas terhadap

aktivitas *email* dalam lingkungan perbankan. Dengan demikian, sistem ini dapat berkontribusi dalam mendukung kelancaran operasional serta meningkatkan kualitas layanan perbankan secara keseluruhan.

1.3 Deskripsi Proyek/ Gambaran Umum Proyek

Aplikasi *email service* berbasis *web* yang menggunakan RabbitMQ sebagai *message broker* dan Elastic Stack ini dirancang untuk PT. Bank Pembangunan Daerah Bali guna meningkatkan efisiensi dan efektivitas dalam pengelolaan *email* secara tersentralisasi. Dengan sistem ini, pengiriman *email* dapat dilakukan dengan lebih cepat, terstruktur, serta memiliki *monitoring* yang lebih baik untuk memastikan setiap *email* terkirim dengan sukses.

Aplikasi ini dikembangkan menggunakan Laravel sebagai *backend* utama. Proyek ini juga mengimplementasikan Docker untuk kemudahan *deployment* serta mendukung integrasi dengan sistem lain melalui *API*. Sistem ini akan mengelola pengiriman *email* dalam jumlah besar dengan mendukung antrian pesan yang lebih efisien melalui RabbitMQ, sehingga setiap *email* diproses berdasarkan prioritasnya. Elastic Stack akan digunakan untuk mencatat dan menganalisis *log email* secara *real-time*, memberikan visibilitas penuh terhadap status pengiriman serta kemungkinan kendala teknis yang terjadi.

1.3.1 Menu dan fitur yang akan dihadirkan pada proyek

Sistem yang dibangun memiliki beberapa menu dan fitur yang dapat diuraikan sebagai berikut:

1. Menu Dashboard: Halaman yang ditampilkan pertama kali saat pengguna (admin atau supervisor) melakukan *login* ke dalam sistem.
 - Fitur statistik pengiriman *email*: Fitur yang menampilkan data statistik pengiriman *email* berupa bentuk diagram dan persentase.
2. Menu Manajemen *Email*: Halaman yang menampilkan data aktivitas pengiriman *email*.
 - Fitur pencarian *email*: Fitur yang dapat digunakan untuk melakukan pencarian data aktivitas pengiriman *email*.

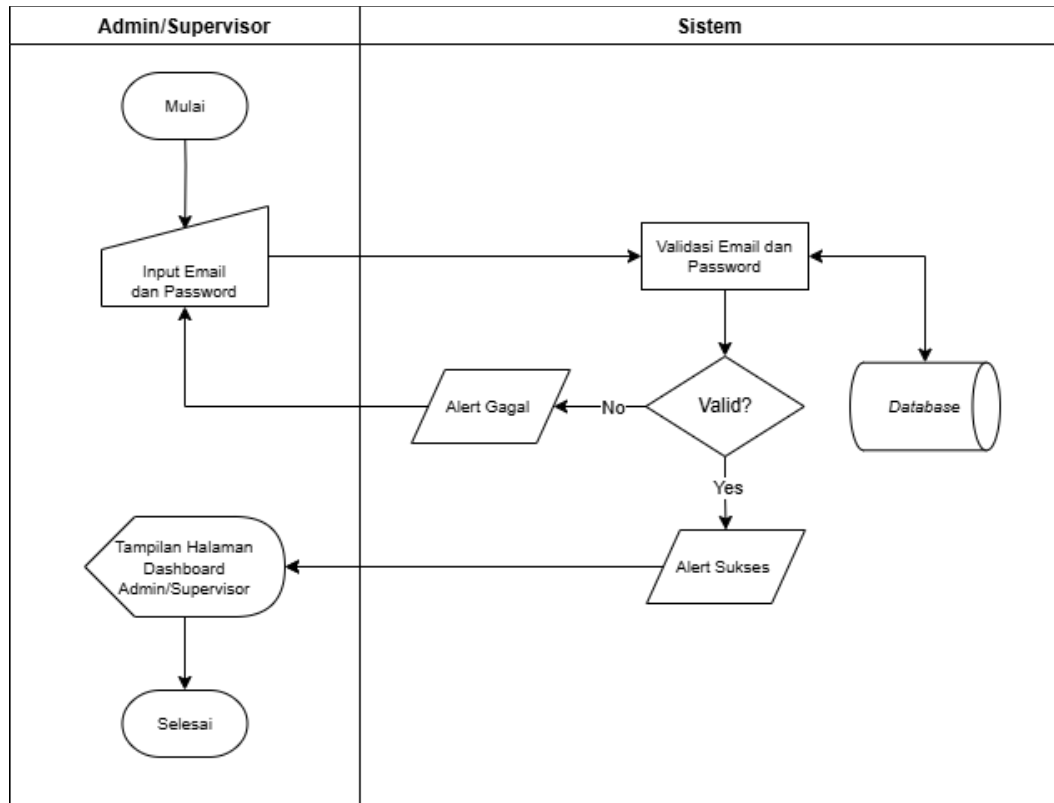
- Fitur pengiriman *email*: Fitur yang memungkinkan pengguna melakukan pengiriman *email* dengan memasukkan langsung melalui *form* maupun mengirim banyak *email* sekaligus menggunakan format excel.
 - Fitur hapus data riwayat pengiriman *email*: Fitur yang memungkinkan pengguna menghapus data riwayat *email* satu persatu ataupun dalam jumlah banyak sekaligus.
 - Fitur detail *email*: Fitur yang memungkinkan pengguna melihat *detail* dari data email yang berhasil terkirim maupun gagal terkirim.
 - Fitur kirim ulang *email*: Fitur yang memungkinkan pengguna melakukan pengiriman ulang *email* yang gagal terkirim.
 - Fitur filter berdasarkan tanggal: Fitur yang memungkinkan pengguna melakukan pengurutan data berdasarkan tanggal.
3. Menu Manajemen Aplikasi: Halaman yang menampilkan data seluruh aplikasi yang menggunakan sistem.
- Fitur pencarian aplikasi: Fitur yang memungkinkan pengguna untuk melakukan pencarian data aplikasi.
 - Fitur tambah aplikasi: Fitur yang memungkinkan pengguna untuk melakukan penambahan aplikasi baru.
 - Fitur hapus data aplikasi: Fitur yang memungkinkan pengguna untuk menghapus data aplikasi.
 - Fitur detail aplikasi: Fitur yang memungkinkan pengguna dapat melihat informasi detail dari aplikasi
 - Fitur ubah *secret key*: Fitur yang memungkinkan aplikasi untuk mendapatkan *secret key* baru.
 - Fitur ubah status aplikasi: Fitur yang dapat mengubah status aplikasi menjadi *disabled* atau *enabled*.
 - Fitur filter berdasarkan tanggal: Fitur yang memungkinkan pengguna melakukan pengurutan data berdasarkan tanggal.
 - Fitur filter berdasarkan abjad: Fitur yang memungkinkan pengguna melakukan pengurutan data berdasarkan abjad.
4. Menu Profil: Halaman yang menampilkan informasi pengguna.

- Fitur ubah *password*: Fitur yang memungkinkan pengguna untuk melakukan perubahan *password*.
 - Fitur edit informasi pribadi: Fitur yang memungkinkan pengguna melakukan perubahan nama dan *email*.
5. Menu Approval Aplikasi: Halaman yang hanya dapat diakses oleh supervisor dan menampilkan data aplikasi yang memerlukan *approval*.
- Fitur *approval*: Fitur yang memungkinkan supervisor agar dapat menyetujui suatu aplikasi ataupun menolak aplikasi untuk menggunakan sistem.
6. Fitur autentikasi: Fitur yang memungkinkan pengguna (admin dan supervisor) *login*, dan *logout* dari sistem.

1.3.2 Flowmap Sistem Email Service

Berikut merupakan *flowmap* dari setiap proses yang ada pada rancangan sistem *email service*:

1) Flowmap Login

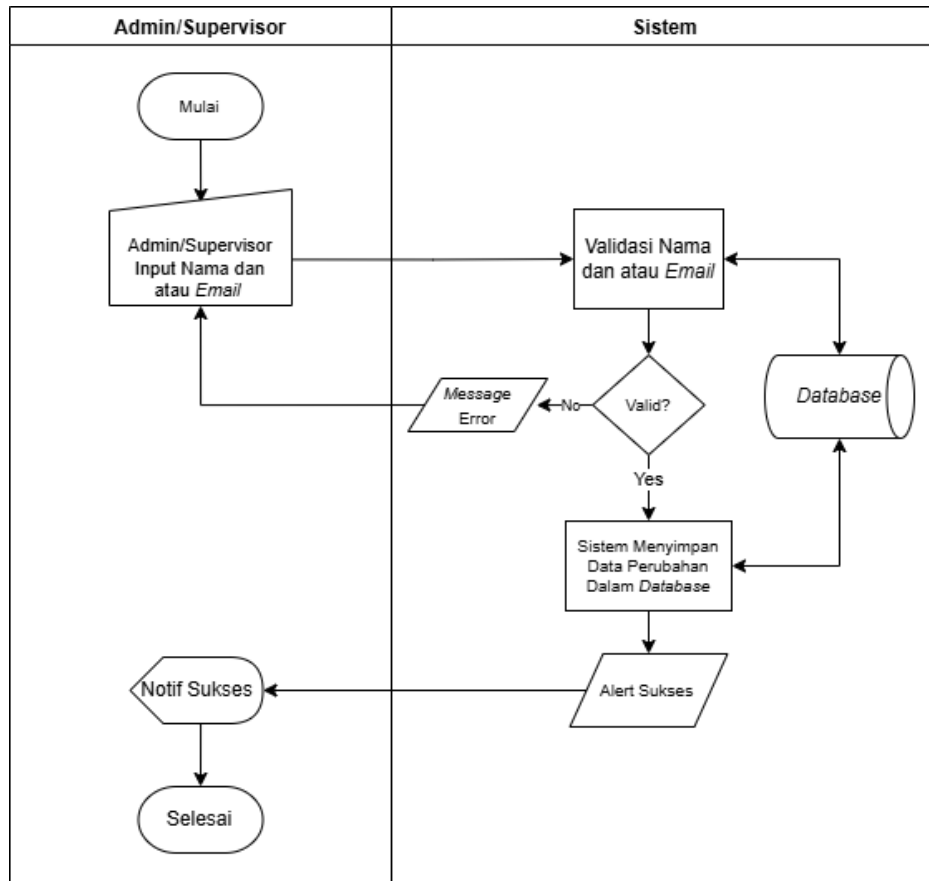


Gambar 1.1 Flowmap Login

Penjelasan *flowmap login*:

- Pengguna memasukkan *username* dan *password*.
- Sistem menerima input tersebut dan melakukan proses validasi dengan mencocokkannya terhadap data yang tersimpan di *database*.
- Jika data *username* dan *password* tidak ditemukan atau tidak cocok, maka data dianggap tidak valid, akan muncul pesan kesalahan *username* atau *password* dan pengguna akan kembali ke tampilan *login* untuk mencoba lagi.
- Jika data *username* dan *password* sesuai dengan yang ada di *database*, maka data dianggap valid. Sistem akan menampilkan halaman *dashboard* sesuai peran pengguna (admin atau supervisor).
- Pengguna berhasil *login*, melihat tampilan *dashboard*, dan proses *login* selesai.

2) Flowmap Edit Profil Admin atau Supervisor

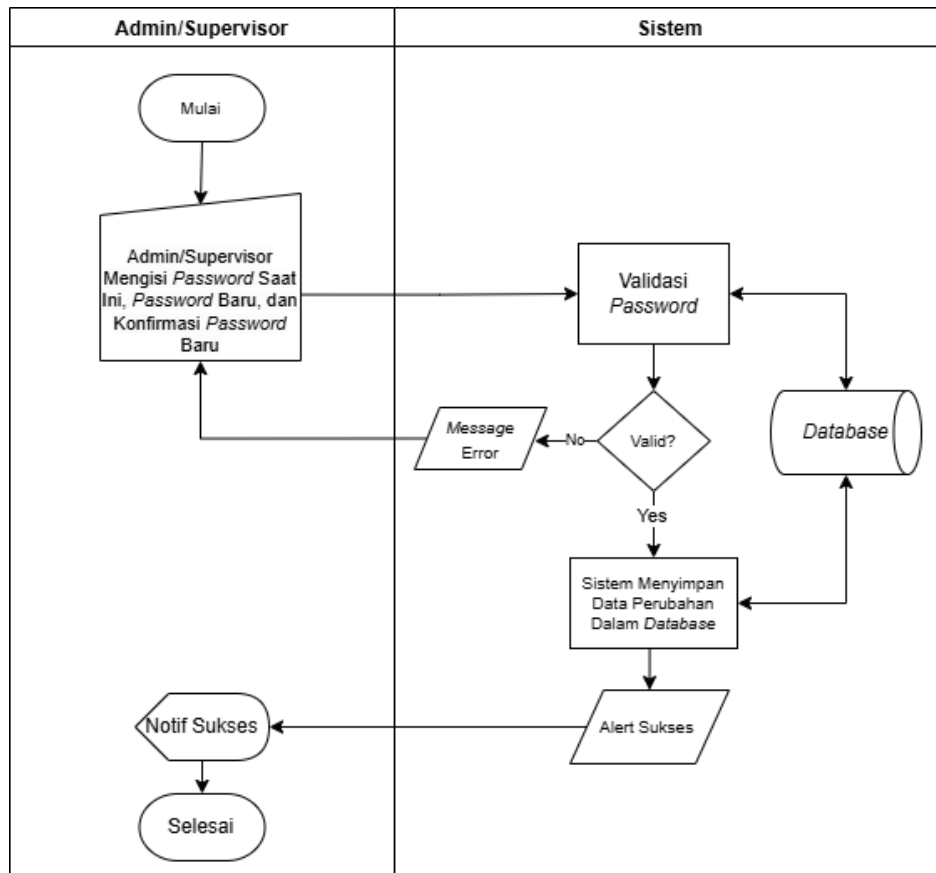


Gambar 1.2 Flowmap Edit Profil Admin atau Supervisor

Penjelasan *flowmap* edit profil admin atau supervisor:

- Admin/supervisor memasukkan nama dan atau *email*.
- Sistem akan memvalidasi input nama dan/atau email yang dimasukkan.
- Jika tidak valid, sistem menampilkan pesan *error*, dan admin/supervisor memasukkan kembali nama dan atau *email* ke dalam *form*.
- Jika valid maka pembaruan data akan disimpan pada *database*.
- Sistem akan memberikan pesan pembaruan data berhasil.
- Proses selesai.

3) Flowmap Edit Password

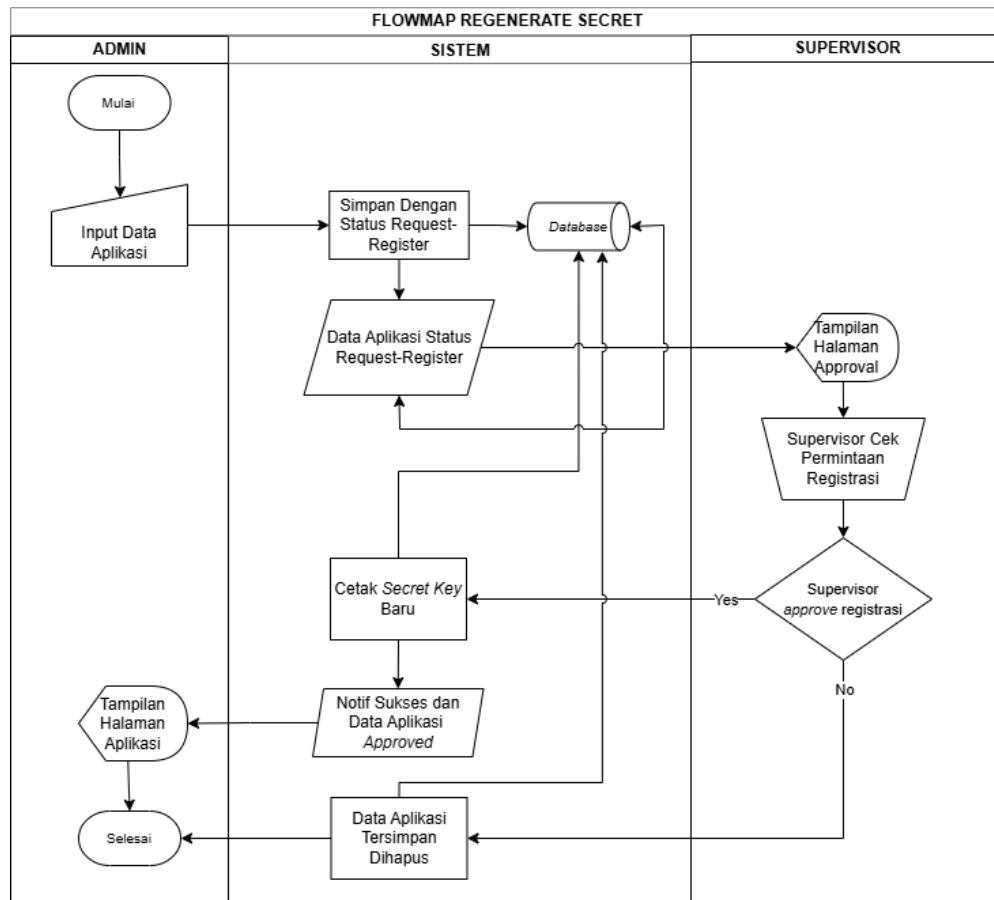


Gambar 1.3 Flowmap Edit Password

Penjelasan *flowmap* edit *password*:

- Admin/supervisor mengisi *password* saat ini, *password* baru, dan konfirmasi *password* baru ke dalam *form* yang tersedia.
- Sistem akan melakukan validasi terhadap data *password* yang dimasukkan.
- Jika tidak valid, sistem akan menampilkan pesan *error*, sehingga admin/supervisor kembali memasukkan data *password* dalam *form*.
- Jika valid, perubahan akan disimpan ke dalam *database*.
- Sistem akan memberikan pesan pembaruan data berhasil.
- Proses selesai.

4) Flowmap Pendaftaran Aplikasi

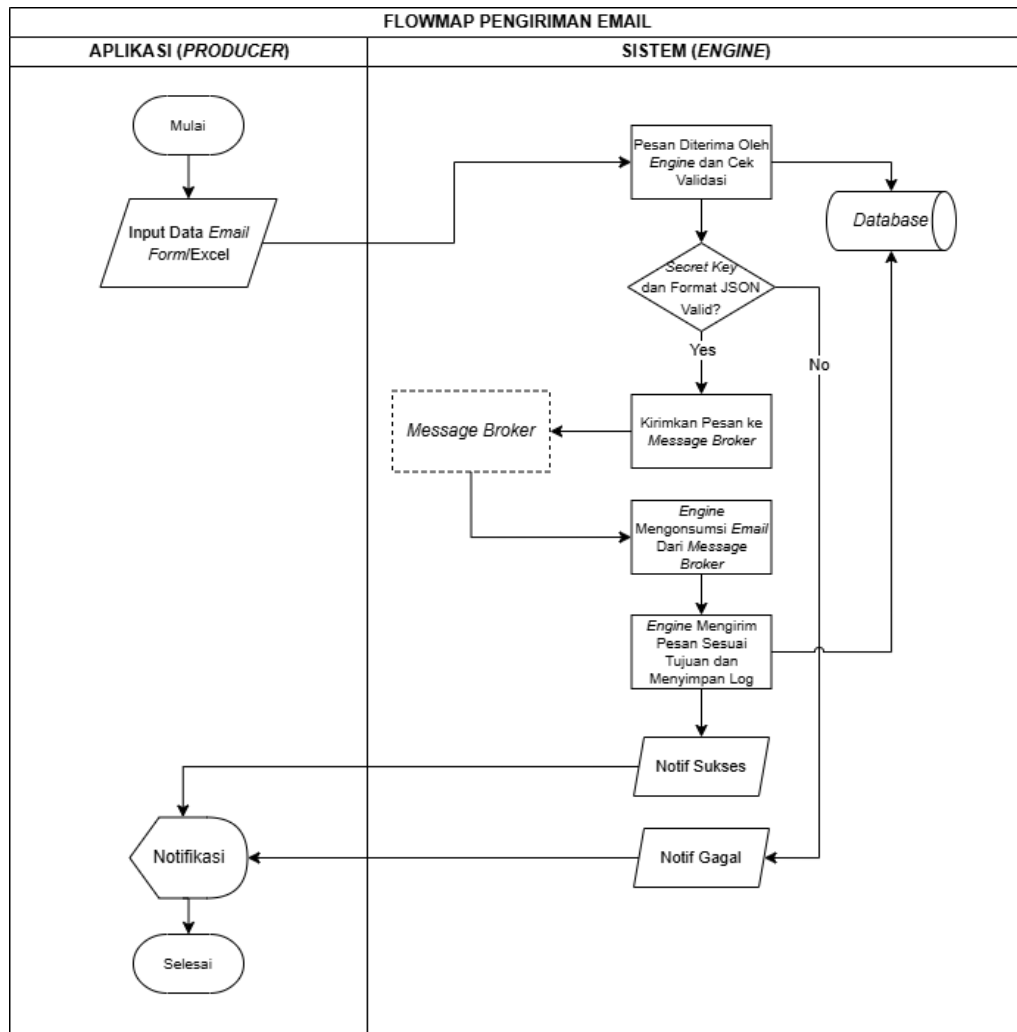


Gambar 1.4 Flowmap Pendaftaran Aplikasi

Penjelasan *flowmap* pendaftaran aplikasi:

- Admin melakukan input data aplikasi yang ingin didaftarkan.
- Sistem akan menyimpan data aplikasi ke dalam *database* dengan isian status “request-register”.
- Supervisor mendapatkan data aplikasi dengan status “request-register” yang membutuhkan persetujuan pada halaman *approval*.
- Jika permohonan disetujui, maka data aplikasi akan disimpan dan sistem akan mencetak *secret key* bagi aplikasi tersebut.
- Jika permohonan ditolak, maka sistem akan menghapus data aplikasi.
- Proses selesai.

5) Flowmap Pengiriman Email



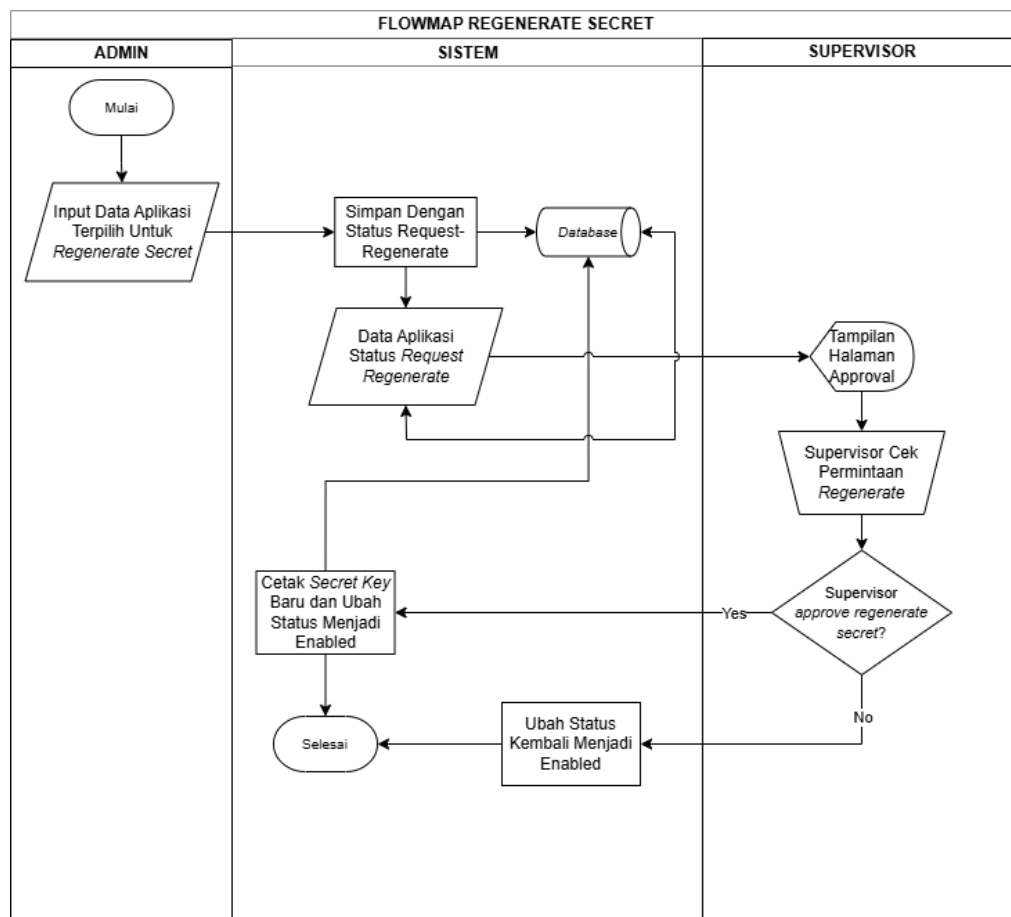
Gambar 1.5 Flowmap Pengiriman Email

Penjelasan *flowmap* pengiriman *email*:

- Proses pengiriman *email* dimulai dari sisi aplikasi (*producer*) yang terdaftar dimana aplikasi memasukkan data pesan *email*, baik dalam format *form* atau *excel*.
- Setelah itu, data dikirimkan ke sistem *engine* dan dilakukan validasi format pengiriman.
- Dalam sistem, dilakukan pemeriksaan dari *secret key* beserta format JSON dari masing-masing pesan yang dikirimkan. Jika valid, pesan akan dikirimkan ke dalam *message broker* RabbitMQ. Jika tidak, sistem menampilkan notifikasi gagal, dan proses akan berakhir.

- d. Kemudian, pesan akan masuk ke dalam *message broker* RabbitMQ dan ditentukan prioritas dari masing-masing pesan, dan disimpan ke dalam antrian (*queue*).
- e. Proses berlanjut ke *engine* yang melakukan konsumsi (*consume*) dari pesan-pesan yang tersimpan sesuai dengan antrian (*queue*) di RabbitMQ.
- f. *Engine* kemudian mengirimkan setiap pesan *email* sesuai tujuan. Masing-masing *log* dari pesan kemudian disimpan ke dalam *database*.
- g. Sistem menampilkan notifikasi sukses.
- h. Proses selesai.

6) *Flowmap Regenerate Secret Key*

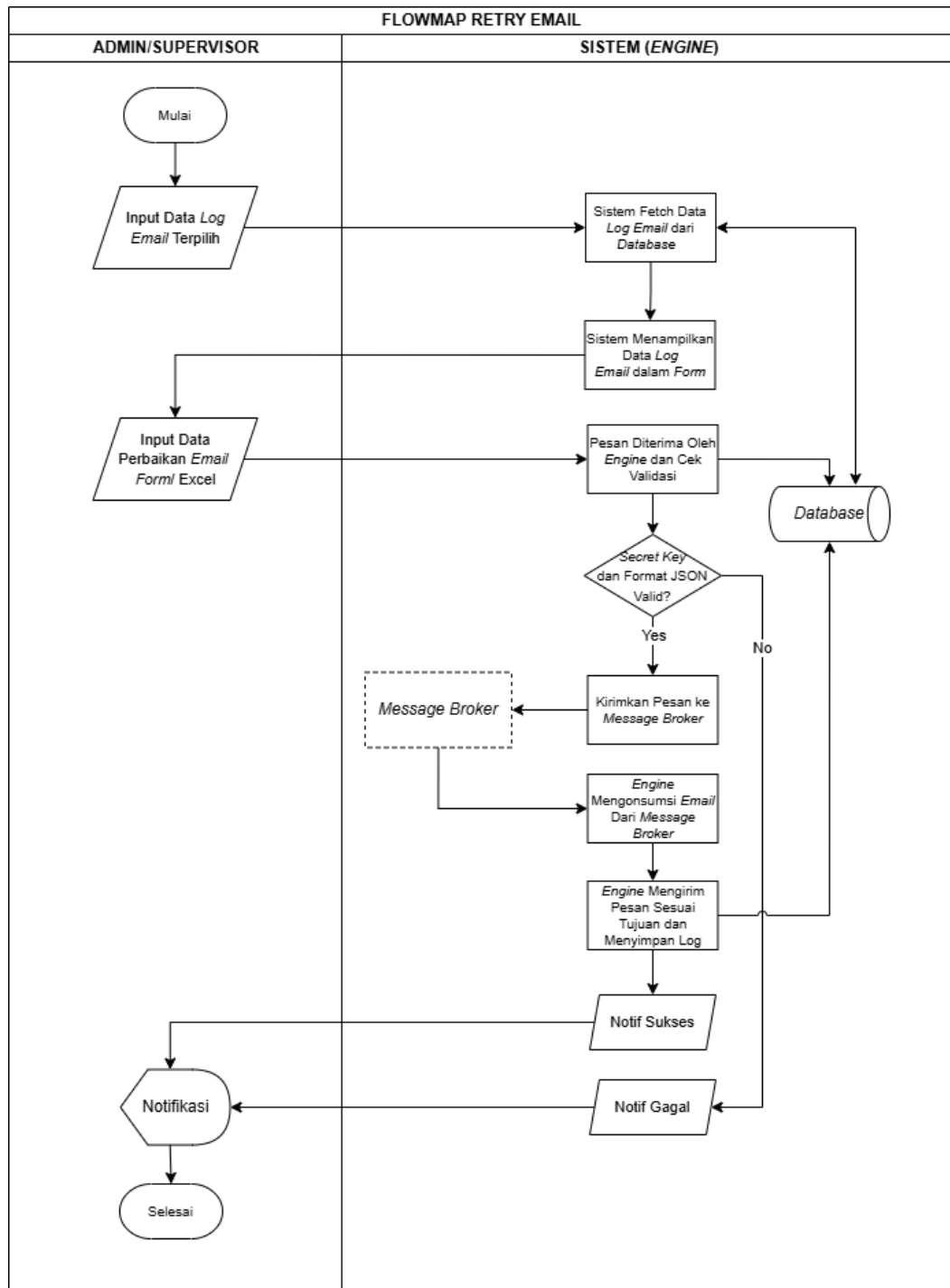


Gambar 1.6 Flowmap Regenerate Secret Key

Berikut penjelasan *flowmap regenerate secret key*:

- a. Admin menekan tombol *request regenerate secret key* pada aplikasi yang memerlukan *secret key* baru, sehingga permintaan data aplikasi tersebut masuk ke dalam sistem.
- b. Sistem akan memperbarui status aplikasi menjadi *request-regenerate*.
- c. Data aplikasi yang melakukan *regenerate secret* akan didapat oleh supervisor pada halaman *approval*.
- d. Jika supervisor menyetujui permintaan *regenerate secret key*, maka sistem akan mencetak *secret key* baru dan mengubah status aplikasi kembali menjadi *enabled*.
- e. Jika supervisor menolak permintaan, maka sistem mengubah status kembali seperti semula, dan *secret key* tidak diperbarui.
- f. Proses selesai.

7) *Flowmap Pengiriman Ulang Email*



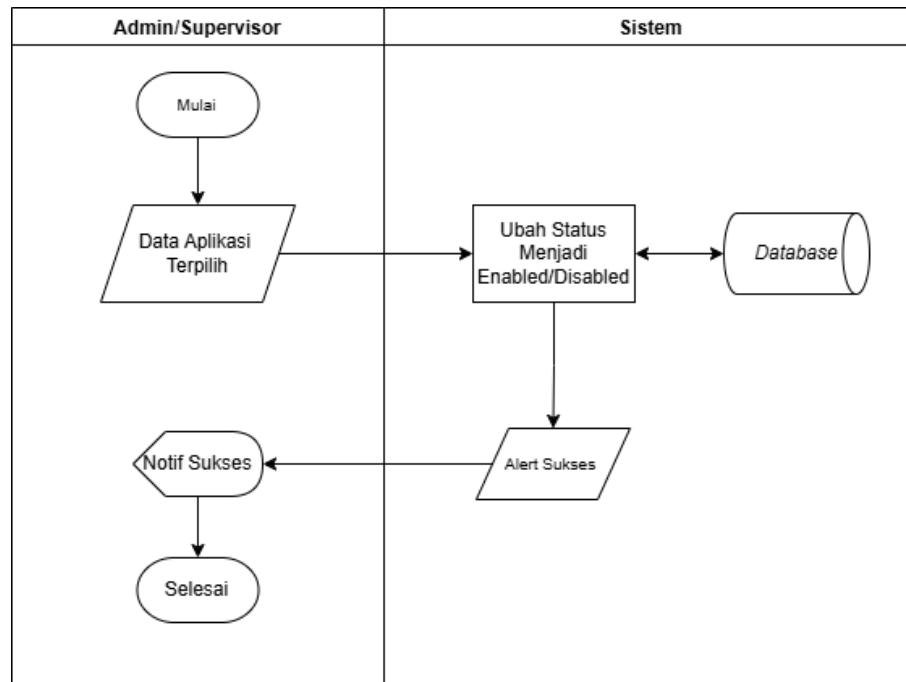
Gambar 1.7 Flowmap Pengiriman Ulang Email

Penjelasan *flowmap* pengiriman ulang *email*:

- Proses pengiriman *email* dimulai dari admin/supervisor yang memilih data *log email* yang mengalami kegagalan.

- b. Setelah itu, sistem mengambil isi pesan dari *email* yang gagal dikirimkan, dan menampilkan data.
- c. Admin/supervisor melakukan perbaikan terhadap data yang menyebabkan gagalnya pengiriman *email*.
- d. Setelah itu, data dikirimkan ke sistem *engine* dan dilakukan validasi format pengiriman.
- e. Dalam sistem, dilakukan pemeriksaan dari *secret key* beserta format JSON dari masing-masing pesan yang dikirimkan. Jika valid, pesan akan dikirimkan ke dalam *message broker* RabbitMQ. Jika tidak, sistem menampilkan notifikasi gagal, dan proses akan berakhir.
- f. Kemudian, pesan akan masuk ke dalam *message broker* RabbitMQ dan ditentukan prioritas dari masing-masing pesan, dan disimpan ke dalam antrian (*queue*).
- g. Proses berlanjut ke *engine* yang melakukan konsumsi (*consume*) dari pesan-pesan yang tersimpan sesuai dengan antrian (*queue*) di RabbitMQ.
- h. *Engine* kemudian mengirimkan setiap pesan *email* sesuai tujuan. Masing-masing *log* dari pesan kemudian disimpan ke dalam *database*.
- i. Sistem menampilkan notifikasi sukses.
- j. Proses selesai.

8) *Flowmap* Ubah Status Aplikasi



Gambar 1.8 Ubah Status Aplikasi

Berikut penjelasan *flowmap* ubah status aplikasi:

- Admin atau supervisor memilih data aplikasi yang akan di *update* status aplikasinya.
- Sistem akan mengubah status aplikasi *enabled* atau *disabled* pada *database*.
- Setelah berhasil, maka muncul pesan bahwa perubahan status berhasil.
- Proses selesai.

1.4 Tujuan Proyek

Tujuan dari proyek “Pengembangan Aplikasi *Email Service* Berbasis *Web* Menggunakan RabbitMQ dan Elastic Stack untuk Pengiriman *Email* Tersentralisasi pada PT. Bank Pembangunan Daerah Bali” diuraikan sebagai berikut:

- Membangun Sistem Pengiriman *Email* yang Terpusat dan Terstruktur:
Untuk menggantikan metode pengiriman *email* yang sebelumnya bersifat terdesentralisasi dan manual, dengan sistem berbasis *web* yang mendukung integrasi lintas aplikasi milik PT. Bank Pembangunan Daerah Bali.

2. Mengoptimalkan Performa Pengiriman *Email* Melalui Teknologi RabbitMQ: Dengan menggunakan RabbitMQ sebagai *message broker*, sistem dapat mengelola antrian *email* secara efisien, memproses pengiriman dalam jumlah besar, serta meminimalisasi risiko kehilangan pesan.
3. Meningkatkan Visibilitas dan Monitoring Aktivitas *Email* Menggunakan Elastic Stack: Elastic Stack diterapkan untuk menyimpan, mengelola, dan menampilkan *log* aktivitas *email* secara *real-time* sehingga memudahkan proses pemantauan, analisis performa, dan *troubleshooting* jika terjadi kegagalan.

1.5 Keuntungan Proyek

Berikut adalah beberapa keuntungan yang diharapkan dari pengembangan aplikasi *email service* berbasis *web* menggunakan RabbitMQ dan Elastic Stack untuk pengiriman *email* tersentralisasi pada PT. Bank Pembangunan Daerah Bali:

1. Efisiensi dalam Pengelolaan *Email*: Sistem yang dikembangkan memungkinkan pengiriman *email* yang lebih terstruktur dan terpusat, mengurangi risiko *email* tidak terkirim atau tertunda tanpa sepengetahuan.
2. Peningkatan Performa dan Keamanan: Dengan implementasi RabbitMQ dan Elastic Stack, sistem akan lebih stabil, dapat menangani antrian pengiriman *email* dengan optimal, serta memberikan analisis data secara *real-time*.
3. Kemudahan dalam *Monitoring* dan *Troubleshooting*: Elastic Stack akan membantu dalam pemantauan sistem secara menyeluruh, memberikan visibilitas lebih baik terhadap *log* yang tersimpan.
4. Manfaat bagi Perusahaan: PT. Bank Pembangunan Daerah Bali akan mendapatkan solusi yang lebih efektif dalam pengelolaan *email*, meningkatkan produktivitas serta keandalan sistem komunikasi internal dan eksternal.

BAB IV

PENUTUP

4.1 Kesimpulan

Berdasarkan hasil analisis kebutuhan, perancangan sistem, serta implementasi sistem *email service* berbasis *web* menggunakan RabbitMQ dan Elastic Stack pada PT. Bank Pembangunan Daerah Bali, dapat disimpulkan bahwa sistem pengiriman email yang terpusat berhasil diimplementasikan dengan baik dan mampu menggantikan metode sebelumnya yang tidak terstruktur menjadi lebih terorganisir dan terdokumentasi. RabbitMQ terbukti efektif sebagai *message broker* dalam mengatur antrean dan prioritas pengiriman *email*, sehingga proses dapat berjalan lebih tertib dan efisien. Selain itu, penerapan Elastic Stack sebagai sistem manajemen *log* juga berjalan dengan baik, memberikan visibilitas secara *real-time* terhadap aktivitas pengiriman *email* melalui kolaborasi antar komponen seperti Beats, Logstash, Elasticsearch, dan Kibana yang memungkinkan analisis *log* dilakukan secara interaktif dan informatif.

4.2 Saran

Berdasarkan hasil pelaksanaan proyek ini, terdapat beberapa saran yang dapat diberikan untuk pengembangan dan pemanfaatan sistem ke depannya:

1. Perlu dilakukan pengujian secara berkelanjutan guna memastikan sistem tetap berjalan dengan baik dan dapat menyesuaikan diri terhadap perubahan kebutuhan operasional perusahaan.
2. Peningkatan fitur dan fungsionalitas sistem dapat terus dilakukan agar sistem *email service* ini semakin mudah digunakan serta mampu mendukung lebih banyak kebutuhan pengguna di masa depan.

DAFTAR PUSTAKA

- [1] D. Antoro, N. Anwar, M. Bahrul Ulum, A. Mulyo Widodo, dan N. Erzed, “Rancang Bangun Sistem Penggajian Karyawan Menggunakan Metode Rapid Application Development (RAD),” *ikraith-informatika*, vol. 7, no. 1, 2022, doi: 10.37817/ikraith-informatika.v7i1.2238.
- [2] A. Nur, I. Purwanto, dan N. Rasyid, “REST API and Message Broker RAbbitMQ for Integration of College Academic Information System,” vol. 16, no. 2, 2023.
- [3] A. Ćatović, N. Buzadija, dan S. Lemes, “Microservice development using RabbitMQ message broker,” *Science, Engineering and Technology*, vol. 2, no. 1, 2022, doi: 10.54327/set2022/v2.i1.19.
- [4] A. A. Wahid, “Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi,” *Jurnal Ilmu-ilmu Informatika dan Manajemen STMIK*, vol. 1, no. 1, 2020.
- [5] S. Sandfreni, M. B. Ulum, dan A. H. Azizah, “Analisis Perancangan Sistem Informasi Pusat Studi Pada Fakultas Ilmu Komputer Universitas Esa Unggul,” *Sebatik*, vol. 25, no. 2, 2021, doi: 10.46984/sebatik.v25i2.1587.
- [6] A. N. Ichtiarto, M. I. Akbar, dan Y. D. Cahyo, “Perancangan Active Database Dalam Sistem Database Toko Online Program Studi Teknik Informatika UWKS,” *Melek IT Information Technology Journal*, vol. 6, no. 1, 2020, doi: 10.30742/melek-it.v6i1.307.
- [7] W. Sholihah, S. Pripambudi, dan A. Mardiyono, “Log Event Management Server Menggunakan Elastic Search Logstash Kibana (ELK Stack),” *JTIM : Jurnal Teknologi Informasi dan Multimedia*, vol. 2, no. 1, 2020, doi: 10.35746/jtim.v2i1.79.
- [8] A. Setiyawan, A. Pinandito, dan W. Purnomo, “Pengembangan Sistem Informasi Log Management Server Monitoring Menggunakan ELK (Elastic Search, Logstash dan Kibana) Stack pada Aplikasi Padichain di PT. Bank Rakyat Indonesia,” 2023. [Daring]. Tersedia pada: <http://j-ptiik.ub.ac.id>
- [9] P. Napoleon dan K. Bayu, “Implementasi Server Log Monitoring System menggunakan Elastic Stack,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 6, no. 4, 2022.
- [10] H. Ilham, B. Wijayanto, dan S. P. Rahayu, “Analysis And Design Of User Interface/User Experience With The Design Thinking Method In The Academic Information System Of Jenderal Soedirman University,” *Jurnal Teknik Informatika (Jutif)*, vol. 2, no. 1, hlm. 17–26, Jan 2021, doi: 10.20884/1.jutif.2021.2.1.30.