

## **PENGEMBANGAN APLIKASI PENDAFTARAN AKUN EMAIL MICROSOFT 365 DAN GOOGLE WORKSPACE BERBASIS LARAVEL DAN REACT DI POLITEKNIK PIKSI INPUT SERANG**

**Muhammad Khaidir Fahram<sup>1</sup>, Vina Vijaya Kusuma<sup>2</sup>**  
Rekayasa Keamanan Siber, Politeknik Piksi Input Serang  
Serang, Indonesia  
e-mail : [1khaidir@fahram.com](mailto:khaidir@fahram.com) , [2vinavijaya@gmail.com](mailto:vinavijaya@gmail.com)

### ***Abstract***

*The creation of institutional email accounts such as Microsoft 365 and Google Workspace is often manually handled by administrators, resulting in inefficiencies and potential data errors. This study aims to design and develop a secure and integrated web-based email registration application. The application is built using Laravel 12 for the backend and React (TypeScript) with Inertia.js and radix-ui for the frontend. The system integrates with the Microsoft Graph API and Google Admin SDK for automated account creation, and with the WhatsApp API for user notifications. In terms of security, the system employs middleware for access control, conditional validation based on user roles (NIM for students, NIDN for lecturers), temporary password encryption using Laravel Crypt, form protection via reCAPTCHA, and rate limiting to prevent abuse. Testing results show that the system improves efficiency, accuracy, and security in the institutional email registration process at Politeknik Piksi Input Serang..*

**Keywords:** *Application Security, Google Workspace, Institutional Email, Laravel, Microsoft 365.*

### **PENDAHULUAN**

Perkembangan teknologi informasi yang semakin pesat telah membawa perubahan signifikan dalam dunia pendidikan, khususnya dalam pemanfaatan layanan digital untuk mendukung proses administrasi dan komunikasi akademik. Salah satu layanan yang sangat penting dalam ekosistem perguruan tinggi adalah email institusional. Layanan *email* ini tidak hanya berfungsi sebagai sarana komunikasi resmi, tetapi juga menjadi kunci akses terhadap berbagai platform pembelajaran daring, penyimpanan *cloud*, dan layanan kolaborasi seperti Microsoft 365 dan Google Workspace (Documentation, 2025). Email institusional adalah sarana komunikasi formal yang diberikan kepada mahasiswa, dosen, dan tenaga kependidikan. Akun ini biasanya diarahkan melalui platform seperti Microsoft 365 atau Google Workspace, sehingga memungkinkan akses yang mulus ke layanan pembelajaran daring, penyimpanan *cloud*, dan kolaborasi dokumen secara real-time. Banyak kampus kini mengonsolidasikan email dan produktivitas ke dalam satu sistem—terutama Microsoft 365—untuk meningkatkan efisiensi, keamanan, dan kemudahan manajemen TI, serta menyediakan pengalaman pengguna yang lebih konsisten dan terpadu (EdTech Magazine 2025b; Illinois Institute of Technology 2024; William & Mary 2024). Selain itu, dengan hadirnya solusi AI seperti Copilot di Microsoft 365 dan Gemini di Google Workspace, pengguna dapat memanfaatkan kecerdasan buatan secara lebih terintegrasi tanpa harus menggunakan aplikasi tambahan (EdTech Magazine 2025a).

Di lingkungan Politeknik Piksi Input Serang, proses pembuatan akun *email* institusional masih dilakukan secara manual oleh *administrator*. Setiap permintaan akun ditangani satu per satu melalui formulir atau komunikasi langsung, yang berpotensi menimbulkan inefisiensi, kesalahan *input* data, dan kendala dalam pelacakan serta dokumentasi permohonan. Seiring

bertambahnya jumlah mahasiswa dan dosen, metode manual ini tidak lagi mampu mengakomodasi kebutuhan dengan efektif. Hasil observasi internal menunjukkan bahwa proses manual dapat memakan waktu rata-rata 2–3 hari untuk satu akun dan memiliki potensi kesalahan input sebesar 12% per bulan, terutama pada penulisan alamat email dan pencatatan NIM/NIDN.

Beberapa penelitian telah membahas otomatisasi pembuatan email institusional di perguruan tinggi. (Shaik 2022) memanfaatkan Office 365 Graph API untuk mengotomatisasi pembuatan dan pengelolaan akun email secara aman, sementara (Mishra 2022) mengimplementasikan pengiriman email terprogram menggunakan Graph API yang dapat diadaptasi untuk proses pembuatan akun massal. Di sisi lain, (Yulianto 2021) menunjukkan penerapan WhatsApp API untuk mengirim notifikasi otomatis kepada pengguna, yang membuka peluang integrasi notifikasi real-time dalam sistem email institusional.

Beberapa penelitian telah mengulas otomatisasi pembuatan email institusional di perguruan tinggi. Kong (2023) membahas desain sistem *bulk email* yang lebih efektif dan personal di lingkungan universitas—dengan meningkatkan tingkat keterbacaan dan optimasi pesan (Kong 2023). Di sisi lain, penerapan WhatsApp API untuk notifikasi real-time dalam konteks pendidikan sudah terealisasi, seperti sistem notifikasi akademik berbasis WhatsApp yang dirancang untuk efisiensi layanan di administrasi akademik (Satria 2023) Namun demikian, belum ada sistem yang secara terpadu menggabungkan keamanan multi-lapis, integrasi lintas platform, dan notifikasi WhatsApp real-time dalam satu solusi komprehensif—menjadikannya sebagai celah riset yang menjanjikan ke depannya.

Selain tantangan efisiensi, aspek keamanan juga menjadi perhatian utama dalam pengelolaan data pengguna. Data pribadi seperti NIM, NIDN, dan alamat *email* bersifat sensitif dan perlu dijaga kerahasiaannya. Sistem pendaftaran yang baik harus mampu memberikan kontrol akses yang tepat, validasi data yang akurat, dan perlindungan terhadap potensi penyalahgunaan oleh pihak yang tidak bertanggung jawab.

Untuk menjawab permasalahan tersebut, dibutuhkan sebuah sistem informasi yang mampu mengotomatisasi proses pendaftaran akun email institusional secara terintegrasi dan aman. Penelitian ini bertujuan untuk merancang dan membangun sebuah aplikasi berbasis web menggunakan Laravel 12 sebagai backend dan React (*TypeScript*) dengan Inertia.js serta radix-ui sebagai *frontend*. Sistem ini dirancang tidak hanya untuk mengurangi waktu pembuatan akun menjadi kurang dari 10 menit per akun, tetapi juga untuk meminimalkan kesalahan input melalui validasi otomatis dan mencegah akses ilegal dengan kombinasi *rate limiting*, middleware autentikasi, enkripsi kata sandi sementara, serta verifikasi *reCAPTCHA* (Laravel 2025; Wahyudi 2023) Aplikasi ini dilengkapi dengan integrasi ke berbagai layanan eksternal seperti Microsoft Graph API, Google Admin SDK, dan WhatsApp API untuk menunjang otomatisasi pembuatan akun dan pengiriman notifikasi. Dari sisi keamanan, sistem didukung oleh *middleware* autentikasi, validasi kondisional berdasarkan peran pengguna, enkripsi *password* sementara dan pembatasan permintaan (*rate limiting*) juga didasarkan pada rekomendasi dokumentasi resmi Laravel (Documentation, 2025). Sistem juga menerapkan fitur keamanan seperti *reCAPTCHA* untuk mencegah akses bot atau spam pada formulir pendaftaran, yang terbukti efektif meningkatkan keamanan aplikasi web (Wahyudi, 2023).

Dengan adanya sistem ini, diharapkan proses pendaftaran akun email institusional di Politeknik Piksi Input Serang dapat berlangsung lebih efisien, aman, dan terdokumentasi dengan baik, serta memberikan kontribusi baru dalam implementasi sistem otomatisasi email institusional yang terintegrasi multi-platform dan dilengkapi pengamanan multi-lapis, yang belum banyak diterapkan di penelitian sebelumnya.

## **METODE PENELITIAN**

Penelitian ini menggunakan pendekatan *Research and Development (R&D)* yang bertujuan untuk menghasilkan produk berupa aplikasi *web* pendaftaran akun email institusional secara otomatis dan terintegrasi. Dalam proses pengembangan sistem, digunakan model *Software*

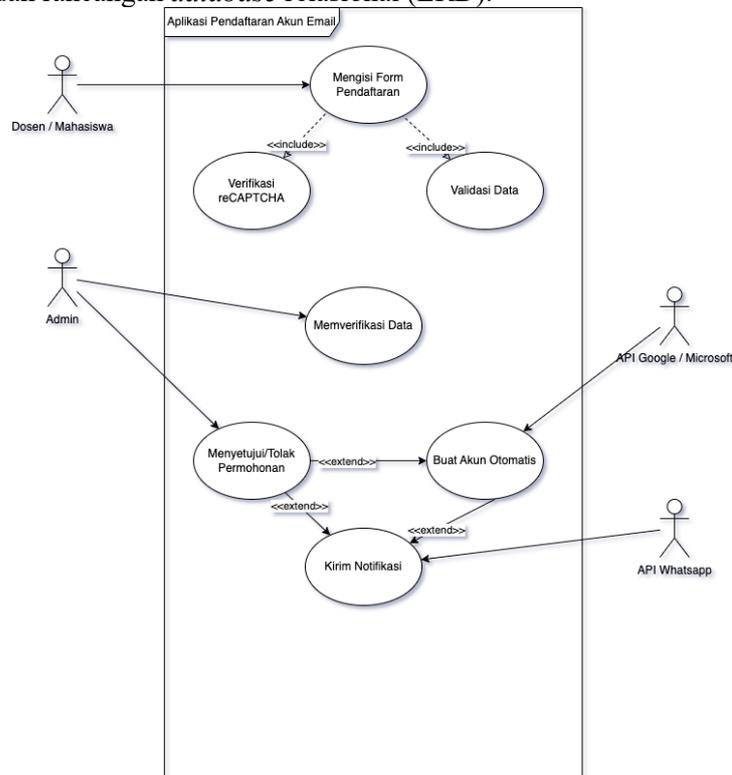
Development Life Cycle (SDLC) dengan pendekatan *Waterfall* sebagaimana juga digunakan pada penelitian pengembangan perangkat lunak sebelumnya (Mulyani, 2018) yang terdiri dari lima tahap utama: analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pemeliharaan.

1. Analisis Kebutuhan

Pada tahap ini dilakukan pengumpulan data melalui wawancara dengan pihak administrator TI dan dokumentasi internal institusi terkait proses pembuatan akun email saat ini. Tujuannya adalah untuk mengidentifikasi kebutuhan fungsional dan non-fungsional sistem, termasuk validasi data, proses *approval*, integrasi API, dan aspek keamanan yang diperlukan.

2. Perancangan Sistem

Tahap ini mencakup desain arsitektur aplikasi, alur proses pendaftaran akun, dan desain antarmuka pengguna (UI). Perancangan dilakukan dengan mempertimbangkan kemudahan penggunaan, skalabilitas, serta keamanan. Tool yang digunakan mencakup *use case diagram*, *sequence diagram* dan rancangan *database* relasional (ERD).



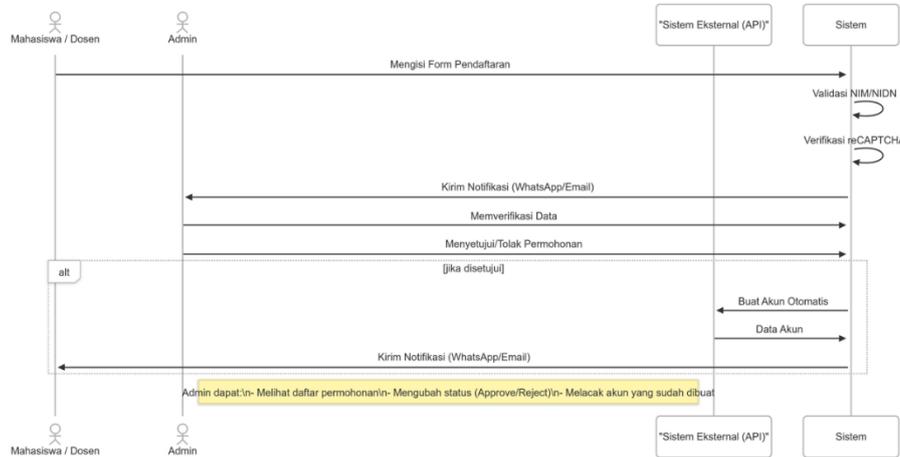
Gambar 1. Use Case diagram

Use Case Diagram ini menggambarkan alur interaksi antara pengguna (dosen/mahasiswa), admin, serta integrasi dengan API eksternal (Google/Microsoft dan WhatsApp):

1. Dosen/Mahasiswa
  - a) Mengisi form pendaftaran akun email.
  - b) Formulir yang diisi akan melalui proses verifikasi *reCAPTCHA* dan validasi data untuk memastikan keamanan dan keabsahan data.
2. Admin
  - a) Memverifikasi data pendaftar.
  - b) Menyetujui atau menolak permohonan pendaftaran.
  - c) Jika disetujui, proses dilanjutkan ke pembuatan akun secara otomatis menggunakan API Google/Microsoft.
  - d) Setelah akun berhasil dibuat, sistem akan mengirim notifikasi (misalnya via WhatsApp) kepada pemohon.

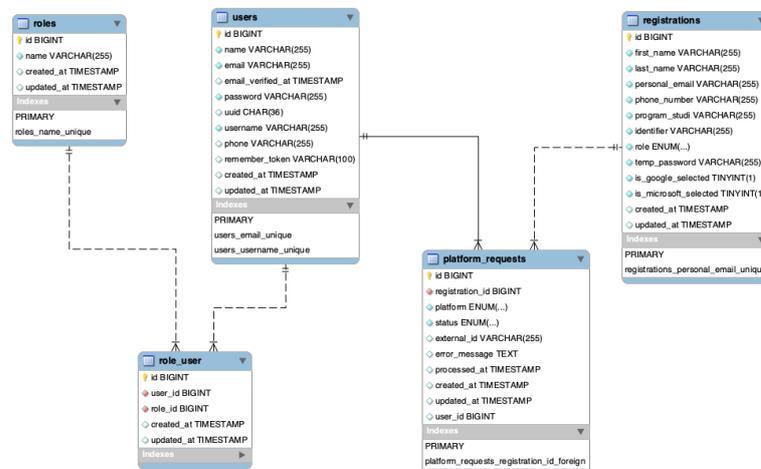
### 3. API Eksternal

- a) API Google/Microsoft: Digunakan untuk membuat akun email institusional secara otomatis.
- b) API WhatsApp: Digunakan untuk mengirim notifikasi otomatis kepada pengguna mengenai status pendaftaran mereka.



Gambar 2. Sequence Diagram

Proses pendaftaran akun email institusional dimulai ketika mahasiswa atau dosen mengisi formulir pendaftaran pada sistem. Setelah itu, sistem akan melakukan validasi terhadap NIM atau NIDN serta verifikasi *reCAPTCHA* untuk memastikan data yang masuk valid dan bukan berasal dari bot. Jika proses validasi berhasil, sistem secara otomatis mengirimkan notifikasi awal kepada admin, baik melalui WhatsApp maupun email. Admin kemudian memverifikasi data yang masuk dan menentukan status permohonan, apakah akan disetujui (*approve*) atau ditolak (*reject*). Jika permohonan disetujui, sistem akan terhubung dengan API eksternal (seperti Google atau Microsoft) untuk melakukan pembuatan akun email secara otomatis. Setelah akun berhasil dibuat, sistem mengirimkan notifikasi akhir berisi informasi akun kepada pemohon melalui WhatsApp atau email. Selain itu, admin memiliki fitur untuk melacak status permohonan dan akun-akun yang sudah berhasil dibuat, sehingga seluruh proses dapat dipantau dan dikelola secara terpusat serta terdokumentasi dengan baik.



### Gambar 3. ERD

Pada rancangan basis data untuk aplikasi pendaftaran akun email institusional. Terdapat lima tabel utama, yaitu *roles*, *users*, *role\_user*, *registrations*, dan *platform\_requests*. Tabel *roles* menyimpan informasi tentang jenis peran yang ada di sistem, seperti admin atau pengguna biasa. Tabel *users* merekam data setiap pengguna aplikasi, termasuk nama, email, dan username.

Relasi antara pengguna dan peran diatur melalui tabel *role\_user*, sehingga satu pengguna dapat memiliki lebih dari satu peran. Selanjutnya, tabel *registrations* digunakan untuk menyimpan data pendaftaran akun email yang dilakukan oleh mahasiswa atau dosen, mulai dari data diri hingga pilihan platform email yang diinginkan.

Terakhir, tabel *platform\_requests* mencatat seluruh permintaan pembuatan akun ke platform eksternal (Google atau Microsoft), termasuk status proses dan informasi terkait hasil permintaan. Dengan desain ini, basis data mampu mendukung pengelolaan pengguna, pendaftaran, dan otomatisasi proses pembuatan akun email secara terstruktur, konsisten, dan aman.

#### 3. Implementasi

Implementasi dilakukan menggunakan Laravel 12 di sisi backend untuk menangani logika bisnis, komunikasi API eksternal, serta otorisasi pengguna. Untuk frontend digunakan React (*TypeScript*) yang diintegrasikan dengan Inertia.js dan radix-ui untuk membangun antarmuka yang responsif dan modern. Sistem ini terhubung dengan MySQL sebagai basis data. Integrasi dengan layanan eksternal dilakukan melalui:

- a. Microsoft Graph API untuk pembuatan akun Microsoft 365.
- b. Google Admin SDK untuk pembuatan akun Google Workspace.
- c. WhatsApp API untuk pengiriman notifikasi otomatis.
- d. Recaptcha API mencegah *spam* atau *abuse* pada *form* pendaftaran akun.

#### 4. Pengujian Sistem

Sistem diuji menggunakan metode *Black Box Testing* untuk memastikan bahwa seluruh fitur berjalan sesuai dengan fungsinya tanpa melihat struktur internal program. Selain itu, dilakukan juga pengujian keamanan terhadap *form input*, autentikasi pengguna, serta integritas data melalui simulasi berbagai skenario serangan umum (misalnya *brute force* dan *spam request*) (Wahyudi, 2023).

#### 5. Pemeliharaan

Tahap ini mencakup perbaikan terhadap *bug* yang ditemukan saat uji coba, serta peningkatan fitur dan keamanan berdasarkan umpan balik dari pengguna. Proses pemeliharaan dilakukan secara berkelanjutan guna memastikan sistem tetap relevan dan aman seiring waktu.

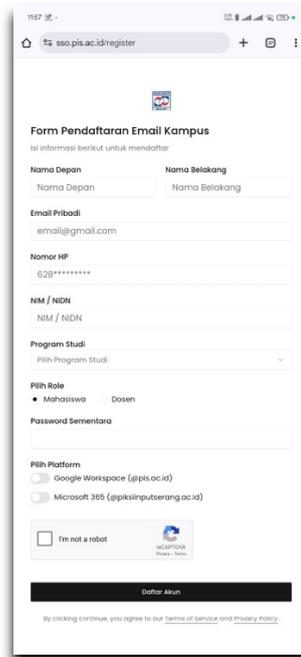
Dengan pendekatan metode ini, sistem yang dikembangkan diharapkan dapat memenuhi kebutuhan institusi dalam menyediakan layanan pendaftaran akun *email* yang cepat, aman, dan terotomatisasi secara penuh.

## HASIL DAN PEMBAHASAN

Setelah proses perancangan dan pengembangan dilakukan, diperoleh hasil berupa aplikasi web pendaftaran akun email institusional berbasis *Laravel* dan *React* yang telah berhasil diimplementasikan di lingkungan Politeknik Piksi Input Serang. Aplikasi ini memiliki antarmuka yang intuitif serta mendukung proses validasi, integrasi API eksternal, dan keamanan sistem.

#### 1. Formulir Pendaftaran Akun

Gambar 4 menunjukkan tampilan halaman formulir pendaftaran akun email kampus, di mana pengguna diminta untuk mengisi informasi seperti nama, email pribadi, nomor HP, NIM/NIDN, program studi, peran (mahasiswa atau dosen), serta memilih platform email yang diinginkan (Google Workspace atau Microsoft 365). Sistem juga dilengkapi dengan proteksi Google reCAPTCHA untuk mencegah *spam* atau akses otomatis oleh *bot*.

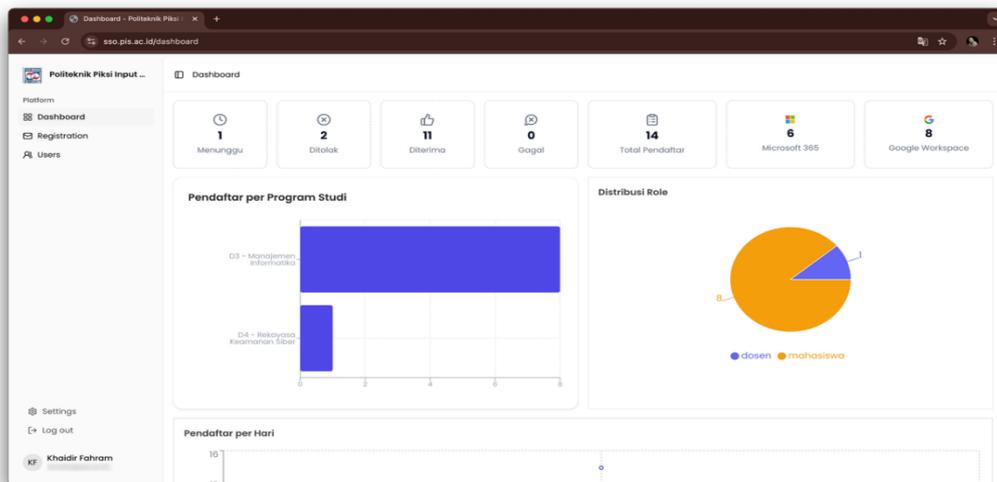


Gambar 4. Tampilan Form Pendaftaran Akun Email Kampus

Validasi kondisional diterapkan pada kolom NIM/NIDN, di mana sistem akan memverifikasi apakah pemohon adalah mahasiswa atau dosen dan memastikan input sesuai.

## 2. Dashboard Administrator

Gambar 5 memperlihatkan tampilan *dashboard admin* yang menampilkan statistik *real-time*, seperti jumlah pendaftar yang menunggu, ditolak, diterima, maupun gagal. Statistik juga mencakup distribusi berdasarkan *platform email* yang dipilih serta program studi dan peran pengguna. Hal ini memudahkan pihak *administrator* dalam melakukan pemantauan dan pengambilan keputusan. Dashboard administrator yang menampilkan statistik pendaftaran dan fitur pelaporan, sejalan dengan best practice dalam pengelolaan sistem informasi sebagaimana dibahas pada penelitian-penelitian terdahulu (Kusuma, 2020).

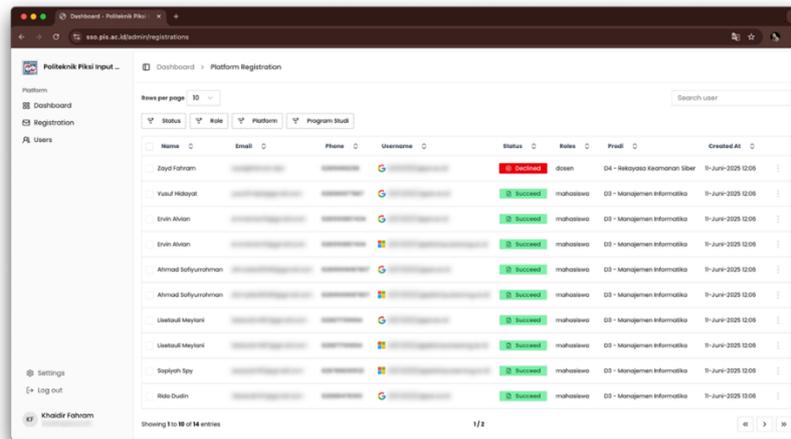


Gambar 5. Tampilan Dashboard Admin – Statistik Pendaftaran

Dashboard juga menyajikan grafik distribusi pendaftar berdasarkan program studi dan *role* (mahasiswa/dosen), serta data pendaftar per hari, sehingga memudahkan analisis dan pelaporan internal.

### 3. Manajemen Permohonan Pendaftaran

Gambar 6 menampilkan halaman manajemen permohonan pendaftaran, di mana *admin* dapat melihat detail setiap pendaftar, termasuk nama, email, platform yang dipilih, serta status permohonan. *Admin* dapat menyetujui atau menolak permintaan langsung dari panel ini, yang secara otomatis memicu pembuatan akun melalui integrasi API dan mengirimkan notifikasi ke pengguna.

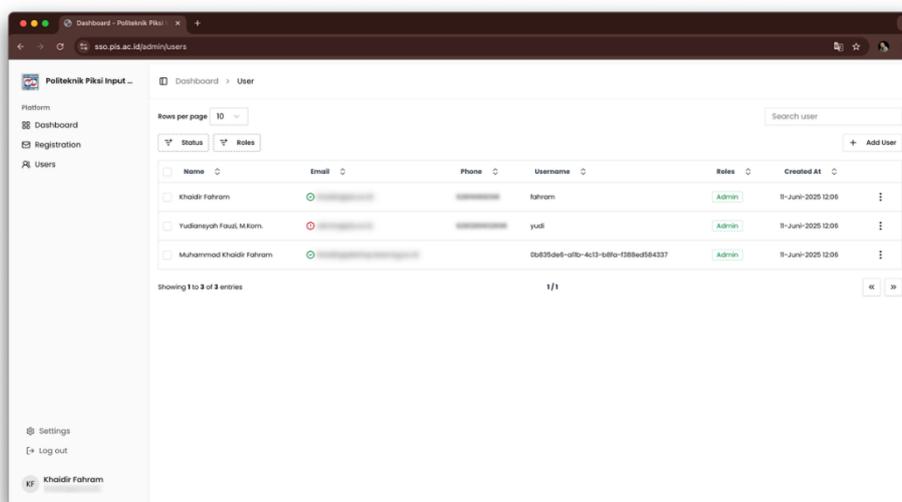


Gambar 6. Tampilan Data Permohonan Pendaftaran Akun

Akun yang berhasil dibuat akan diberi label “*Succeed*” dan dikaitkan dengan username institusional seperti @pis.ac.id atau @piksiinputserang.ac.id.

### 4. Manajemen Pengguna

Sebagaimana ditunjukkan dalam Gambar 7, sistem juga memiliki modul manajemen pengguna, di mana admin dapat melihat daftar *administrator* yang memiliki akses ke sistem. Setiap pengguna admin ditampilkan dengan *email*, nomor telepon, dan *username*, serta status akun dan waktu pembuatan.



Gambar 7. Tampilan Manajemen Pengguna (*Admin Panel*)

Modul ini penting untuk menjaga keamanan dan integritas sistem dengan membatasi hak akses hanya kepada pihak yang berwenang.

#### 5. Keberhasilan Integrasi API dan Otomasi

Dari sisi integrasi, sistem telah berhasil dikoneksikan dengan Microsoft Graph API, Google Admin SDK, WhatsApp API.

```
1 $createResponse = Http::withToken($this->accessToken)
2   ->post('https://graph.microsoft.com/v1.0/users', [
3     'accountEnabled' => true,
4     'displayName' => $fullname,
5     "surname" => $request->registration->first_name,
6     "givenName" => $request->registration->last_name,
7     'mailNickname' => explode('@', $email)[0],
8     'userPrincipalName' => $email,
9     "mobilePhone" => $request->registration->phone_number,
10    "preferredLanguage" => "id-ID",
11    "usageLocation" => "ID",
12    "otherMails" => [$request->registration->personal_email],
13    'passwordProfile' => [
14      'forceChangePasswordNextSignIn' => false,
15      'password' => $password,
16    ],
17  ]);
```

**Gambar 8.** Microsoft Graph API untuk pembuatan akun Microsoft 365 secara otomatis

Kode tersebut merupakan contoh implementasi pembuatan akun pengguna baru di Microsoft 365 secara otomatis menggunakan Microsoft Graph API. Proses ini dilakukan dengan cara mengirim permintaan *HTTP POST* ke endpoint *https://graph.microsoft.com/v1.0/users*, beserta data pengguna yang dibutuhkan.

Data yang dikirim meliputi status akun (*accountEnabled*), nama lengkap (*displayName*), nama depan dan belakang, *email*, nomor telepon, bahasa dan lokasi, serta *email* pribadi sebagai alternatif. Selain itu, terdapat pengaturan profil *password*, di mana *password* awal diatur dan pengguna tidak dipaksa mengubahnya pada saat login pertama (*forceChangePasswordNextSignIn* bernilai *false*).

Kode ini menunjukkan bagaimana sistem dapat mengotomatisasi proses pembuatan akun email institusional secara efisien, konsisten, dan aman, serta memastikan integrasi langsung antara aplikasi internal kampus dengan layanan Microsoft 365. Pendekatan ini sangat relevan untuk mendukung digitalisasi layanan administrasi pendidikan tinggi.

```
1
2 $client = self::getClient();
3 $service = new Directory($client);
4
5 $user = new Directory\User([
6     'name' => [
7         'givenName' => $data['givenName'],
8         'familyName' => $data['familyName'],
9     ],
10    'password' => $data['password'],
11    'primaryEmail' => $data['primaryEmail'],
12    'recoveryEmail' => $data['recoveryEmail'] ?? null,
13    'recoveryPhone' => $data['recoveryPhone'] ?? null,
14 ]);
15
16 $createdUser = $service->users->insert($user);
```

**Gambar 9.** Google Admin SDK untuk pembuatan akun Google Workspace secara langsung.

Kode tersebut adalah contoh implementasi pembuatan akun email Google Workspace secara otomatis melalui Google Admin SDK *Directory API*. Proses dimulai dengan mengambil *client API*, lalu membuat *instance service Directory*. Selanjutnya, sebuah objek user baru dibuat dengan data-data seperti nama depan, nama keluarga, password, email utama, email pemulihan, dan nomor telepon pemulihan. Setelah seluruh data pengguna diatur, proses pembuatan akun dieksekusi dengan memanggil fungsi *insert* pada *service users*.

```
1 public function handle()
2 {
3     $response = Http::asForm()->withHeaders([
4         'Accept' => 'application/json',
5         'Content-Type' => 'application/x-www-form-urlencoded',
6         'Authorization' => 'Bearer ' . env('WA_SECRET_KEY')
7     ])->post(env('WA_END_POINT'), [
8         'phone' => $this->to,
9         'receiver' => $this->to,
10        'token' => env('WA_SECRET_KEY'),
11        'device' => env('WA_DEVICE'),
12        'message' => $this->message,
13        'type' => 'chat',
14    ]);
15 }
```

**Gambar 10.** WhatsApp API untuk mengirimkan notifikasi hasil pendaftaran secara *real-time* kepada pemohon.

Setiap pendaftaran yang disetujui akan memicu pemanggilan API untuk mengirim pesan WhatsApp secara otomatis melalui API pihak ketiga dari aplikasi *backend*. Penjelasannya, fungsi

*handle()* akan membuat permintaan HTTP POST ke endpoint API WhatsApp yang sudah dikonfigurasi (*WA\_END\_POINT*). Permintaan ini dikirim dalam format form data dengan beberapa *header*, termasuk otorisasi *Bearer token* yang diambil dari *environment variable*. Data yang dikirim meliputi nomor telepon penerima (*phone* dan *receiver*), *token API*, *ID device WhatsApp*, isi pesan yang akan dikirim (*message*), dan tipe pesan (*type*, dalam hal ini "chat").

Dengan implementasi seperti ini, sistem dapat secara otomatis mengirim notifikasi atau informasi penting ke pengguna melalui WhatsApp setelah terjadi proses tertentu (misal: berhasil daftar email institusi), sehingga komunikasi dengan pengguna menjadi lebih efisien dan responsif.

## 6. Keamanan Sistem

Sistem menerapkan berbagai fitur keamanan, antara lain:

```
1 Route::middleware("role:admin")
2   ->prefix('admin')
3   ->group(function () {
4       Route::post('set-role', [UserController::class, 'setRole' ])->name('users.setRole');
5       Route::delete('/users/delete-multiple', [UserController::class, 'deleteMultiple'])
6         ->name('users.deleteMultiple');
7       Route::resource('users', UserController::class);
8       Route::delete('registrations/approve', [RegistrationController::class, 'approve'])
9         ->name('registrations.approve');
10      Route::delete('registrations/decline-multiple', [RegistrationController::class, 'declineMultiple'])
11        ->name('registrations.declineMultiple');
12      Route::resource('registrations', RegistrationController::class);
13  })
14  ->name('admin');
```

**Gambar 11.** *Middleware* auth dan admin untuk autentikasi dan pembatasan akses berdasarkan peran.

Gambar tersebut adalah rute (*route*) Laravel yang digunakan untuk mengatur akses fitur khusus admin pada aplikasi. seluruh rute dalam grup ini hanya dapat diakses oleh pengguna dengan peran admin, karena dilindungi oleh *middleware role:admin* dan memiliki *prefix URL admin/*.

```
1 Route::middleware('throttle:3,1')
2   ->post('/register', [RegistrationController::class, 'store']);
```

**Gambar 12.** Proteksi *form* dengan reCAPTCHA dan pembatasan akses menggunakan Laravel *ThrottleRequests*.

Pada Gambar 12 terdapat *middleware throttle:3,1* membatasi akses ke *endpoint /register* agar hanya boleh diakses maksimal 3 kali dalam 1 menit oleh setiap pengguna atau alamat IP. Hal ini bertujuan untuk mencegah spam atau penyalahgunaan form pendaftaran, serta melindungi aplikasi dari serangan otomatis (*brute force*). Dengan pendekatan ini, sistem menjadi lebih aman dan stabil dalam menangani permintaan pendaftaran akun baru.

Fitur-fitur ini menjadikan sistem tidak hanya efisien tetapi juga aman dalam menangani data pengguna dan permohonan akun yang bersifat sensitif.

## KESIMPULAN

Pengembangan aplikasi pendaftaran akun email institusional berbasis web ini berhasil memberikan solusi terhadap permasalahan proses manual yang selama ini dilakukan di Politeknik Piksi Input Serang. Sistem yang dibangun menggunakan Laravel 12 sebagai *backend* dan *React (TypeScript)* sebagai frontend mampu menyediakan antarmuka pengguna yang responsif, serta proses *backend* yang terstruktur dan aman. Aplikasi ini mendukung:

- a. Validasi data pengguna secara otomatis dan dinamis sesuai dengan peran (mahasiswa atau dosen),
- b. Persetujuan permintaan akun oleh *administrator*,
- c. Integrasi penuh dengan Microsoft Graph API dan Google Admin SDK untuk otomatisasi pembuatan akun *email* institusional,
- d. Pengiriman notifikasi melalui WhatsApp API dan Email.
- e. Implementasi berbagai mekanisme keamanan seperti *middleware* autentikasi, enkripsi data, reCAPTCHA, dan rate limiting.

Hasil pengujian sistem menunjukkan bahwa seluruh fitur berfungsi dengan baik dan sistem dapat meningkatkan efisiensi, akurasi, serta keamanan dalam proses pendaftaran akun email institusional. Aplikasi ini juga memudahkan administrator dalam memantau dan mengelola permohonan akun secara terpusat dan terdokumentasi dengan baik.

Dengan adanya sistem ini, proses digitalisasi layanan email di lingkungan kampus dapat berjalan lebih efektif dan profesional, mendukung visi institusi dalam meningkatkan pelayanan berbasis teknologi informasi (Kusuma, 2020). Studi ini juga melengkapi penelitian sebelumnya tentang aplikasi layanan publik berbasis teknologi di Provinsi Banten (Fahram, 2016), serta penelitian-penelitian lain yang membahas pengembangan aplikasi pelayanan publik maupun otomasi berbasis teknologi informasi di berbagai bidang (Santa, 2025), khususnya dalam hal otomasi layanan kampus.

## SARAN

Untuk mendukung keberlanjutan dan pengembangan lebih lanjut dari sistem pendaftaran akun email institusional yang telah dibangun, beberapa saran berikut dapat dipertimbangkan:

1. Integrasi *Single Sign-On (SSO)*  
Pengembangan selanjutnya dapat mengimplementasikan fitur SSO agar akun yang dibuat dapat langsung digunakan untuk mengakses berbagai layanan internal kampus seperti *e-learning*, sistem informasi akademik, dan portal mahasiswa secara terpusat. Sebagaimana tren transformasi digital layanan pendidikan (Maesaroh, 2021).
2. *Dashboard* Pemohon  
Perlu ditambahkan fitur *dashboard* khusus bagi mahasiswa atau dosen agar mereka dapat memantau status permohonan akun secara langsung tanpa harus menghubungi admin.
3. Manajemen Pengguna Lanjutan  
Pengelolaan hak akses admin dan pelaporan aktivitas pengguna dapat dikembangkan lebih lanjut guna meningkatkan akuntabilitas dan keamanan sistem.
4. *Audit Trail* dan *Logging*  
Disarankan untuk menambahkan fitur pencatatan *log* aktivitas sistem secara menyeluruh, terutama pada proses sensitif seperti pembuatan akun dan perubahan status permohonan.
5. *Pengujian Keamanan Lebih Lanjut*  
Untuk memastikan sistem aman terhadap potensi serangan seperti *SQL injection*, *XSS*, dan *CSRF*, pengujian keamanan lanjutan seperti *penetration testing* sebaiknya dilakukan secara berkala.

Dengan memperhatikan saran-saran tersebut, sistem diharapkan dapat terus dikembangkan untuk memenuhi kebutuhan institusi secara lebih komprehensif serta mendukung transformasi digital yang berkelanjutan di lingkungan Politeknik Piksi Input Serang.

#### DAFTAR PUSTAKA

- Fahram, M. K. (2016). *Aplikasi Fasilitas Umum Provinsi Banten Berbasis Lokasi Pada Platform Android*. *Jurnal Insan Unggul*, 4(2), 13–40. <https://doi.org/10.47926/insanunggul.2016.4.2.13-40>
- Google Developers. (n.d.). *Admin SDK Directory API*. Retrieved June 11, 2025, from <https://developers.google.com/admin-sdk>
- Kristofel Santa & Rafter Johannes Karouw (2025) *Implementasi Metode Extreme Programming pada Aplikasi Pelayanan Publik Kewaspadaan Nasional dan Penanganan Konflik Kesbangpol Kota Manado*. <https://doi.org/10.47080/iftech.v7i1.3647>
- Kusuma, V. V., Saifudin, A., & Rohmat, S. (2020). *Sistem Informasi Pengendalian Persediaan Bahan Produksi Berdasarkan Reorder Point Berbasis MRP pada PT. Tereos FKS Indonesia*. *Jurnal Insan Unggul*, 8(1), 1–22. <https://doi.org/10.47926/insanunggul.2020.8.1.1-22>
- Microsoft. (n.d.). *Overview of Microsoft Graph*. Retrieved June 11, 2025, from <https://learn.microsoft.com/en-us/graph/overview>
- Laravel Documentation. (2025). *Security - Laravel 12.x Documentation*. Retrieved from <https://laravel.com/docs/12.x/security>
- Mulyani, D., & Patrie, H. (2018). *Analisa dan Perancangan Sistem Informasi Administrasi Simpan Pinjam Koperasi pada SMP Negeri 264 Jakarta Barat dengan Metodologi Berorientasi Obyek*. *Jurnal Idealis*, 1(4), 190–197.
- Maesaroh, S., Andriani, D., & Agustiningrum, A. (2021). *Sistem Informasi Koperasi Simpan Pinjam pada PT Susanti Megah Berbasis Web*. *Journal SENSI*, 7(2), 140–152.
- Sahrul Ramdani Nugraha & Agus Ramdhani Nugraha (2025) *Sistem Substitusi Air pada Akuarium Berbasis Internet of Things (IoT)* <https://doi.org/10.47080/iftech.v7i1.3723>
- Wahyudi, S. (2023). *Penerapan reCAPTCHA untuk Meningkatkan Keamanan Formulir Web*. *Jurnal Teknologi Informasi dan Komputer*, 9(1), 55–63.