BAB 1
PENDAHULUAN

1.1 Latar Belakang Masalah

Di era revolusi industri 4.0, penipuan email menjadi ancaman serius terhadap keamanan digital. Oknum yang menyalahgunakan ilmu teknologi semakin banyak beraksi, salah satu taktik yang sering digunakan adalah *phishing*. Dengan berkedok email palsu, oknum dapat mengelabui korban untuk mendapatkan informasi pribadi, kredensial login, atau bahkan informasi finansial dari korban [2]. Kasus *phishing* yang tiada henti dan taktik yang terus diperbaharui, menjadikan kebutuhan akan solusi efektif sangat diperlukan bagi masyarakat [9,12].

Tak hanya satu, berbagai macam *platform* pengelola email sering menjadi fokus utama sebagai target kegiatan *phishing* oleh oknum yang tidak bertanggung jawab. Pada kuesioner yang penulis buat dengan target responden merupakan karyawan dari perusahaan *startup* dalam bidang website *development*, terlihat bahwa 60 dari 64 atau 93% responden merasa terbantu jika aplikasi yang akan dibuat dapat berjalan dengan baik di lebih dari satu *platform* pengelola E-mail, sedangkan *anti-phishing* yang tersedia saat ini hanya dapat digunakan pada *Google Mail* saja [2]. Hal ini dapat menyebabkan pengguna baik disengaja maupun tidak disengaja dapat mengakses email *phishing* dari platform email lain tanpa peringatan apapun dan berpotensi menjadi korban dari email phishing karena kurangnya kompatibilitas dalam sistem anti-*phishing* yang saat ini tersedia.

Persoalan lain yang menjadi inti pengembangan aplikasi ini merupakan kurang nya transparansi dari sistem keamanan pada *platform* email tertentu. Salah satu nya adalah *platform* pengelola email yang sudah tidak asing bagi masyarakat seperti *Gmail* (*Google Mail*). Hingga saat penulis membuat skripsi ini, pengguna tidak menerima informasi dalam bentuk *visual* seperti *real-time alert* dan *click prevention* dari proses keamanan yang diterapkan oleh *platform* email yang sedang digunakan [1]. ini merupakan masalah penting bagi pengguna, mengingat transparansi dan kepercayaan kepada pengguna merupakan hal yang patut menjadi prioritas [5]. Hal ini juga berpotensi meningkatkan peluang pengguna mengabaikan ancaman yang ada tanpa sepengetahuan pengguna [4]. Dengan menerapkan *real-*

time alert dan *click prevention,* dapat membantu pengguna agar tetap berhati-hati. Dan terlihat pada kuesioner yang penulis buat, 62 dari 64 atau 96% responden menyatakan terbantu jika aplikasi yang akan dibuat memiliki kemampuan untuk memberikan *real-time alert* dan *click prevention* untuk mencegah pengguna meng-akses url atau file yang ter-indikasi berbahaya. Maka dari itu, *platform* email saat ini masih belum dapat mencegah tindakan yang dilakukan oleh pengguna seperti mengklik tautan atau membuka lampiran, sehingga pengguna bisa mendapat peringatan sebelum di arahkan oleh sistem menuju tautan atau file tertentu.

Masalah lain yang tidak kalah penting berkaitan dengan kendala dalam keamanan platform email yang masih memungkinkan *file attachment* berbahaya untuk lolos dari sistem deteksi yang ada. Saat ini, seringkali sistem keamanan pada *platform* email tidak dapat secara optimal mengidentifikasi dan mengatasi ancaman yang terkandung dalam *file attachment* [2]. Kelemahan ini memberikan peluang bagi penipu untuk menyusupkan *payload* berbahaya melalui email, mengakibatkan peningkatan risiko terhadap serangan *phishing* dan peretasan data. Selain itu, melalui file attachment seringkali penipu menyusupkan skema penipuan yang lebih kompleks, seperti *ransomware* atau *malware* berbahaya lainnya, yang dapat merugikan pengguna [3]. Dan terlihat dari kuesioner yang penulis buat, 64 dari 64 atau 100% responden menyatakan terbantu jika aplikasi yang akan dibuat memiliki kemampuan untuk meng-analisa *file attachment* dari ada nya ancaman *malware*. Oleh karena itu, diperlukan upaya lebih lanjut untuk meningkatkan kemampuan deteksi terhadap file attachment yang berpotensi berbahaya sehingga dapat mencegah ancaman ini lebih efektif dan melindungi pengguna dari potensi risiko keamanan yang serius.

Sebagai dasar dalam pengembangan aplikasi *chrome extension* untuk mendeteksi email penipuan, penulis berupaya untuk mengevaluasi perkembangan dari beberapa tahun lalu yang dilakukan oleh peneliti lain dalam keamanan email dan deteksi penipuan. Seperti pada jurnal dalam penelitian yang dilakukan oleh Pavan, Venkata Sai Mounika, Mahitha, dan Ameena begum [8]. Mereka menggunakan algoritma *Random Forest* yang digabungkan dengan algoritma *Decission Tree* untuk membuat prediksi dan melakukan *test* untuk meng-aplikasikan dataset lalu membuat aplikasi *chrome extension*. Mereka juga

memanfaatkan kemampuan dari *chrome extension* untuk memanipulasi DOM dari struktur email dan mendapatkan informasi seperti link URL. Dan model yang mereka buat digunakan untuk memverifikasi URL secara *real-time* termasuk domain, SSL (*Secure Sockets Layer*), web *traffic* dan hosting *provider*.

Selanjutnya, pada jurnal dalam penelitian yang dilakukan oleh Agus Fatkhurohman, dan Eli Pujastuti [7]. Mereka menggunakan algoritma *Naïve Bayes* sebagai dasar metode untuk meningkatkan keamanan dari *phishing*. Mereka menyebutkan terdapat 4 golongan utama dalam karakteristik phishing, yaitu:

1. *Address Bar Based Feature*
2. *Abnormal based Feature*
3. HTML dan Javascript
4. *Domain based Feature*

Dari setiap karakteristik tersebut, dapat digabungkan dengan penelitian yang dilakukan oleh Vishal Verma, dan Anurag Sinha [6]. Mereka menggunakan sentiment analisis dari 3 karateristik, yaitu *Address Bar based Feature*, *Abnormal based Feature,* dan HTML and Javascript.

1.2 Identifikasi Masalah

Berdasarkan latar belakang masalah diatas, dapat disimpulkan masalah sebagai berikut:

1. Kurangnya kompatibilitas dalam sistem *anti-phishing* yang saat ini tersedia.
2. Sulitnya pengguna untuk mengidentifikasi email yang beresiko penipuan.
3. Kurangnya kemampuan platform email saat ini dalam deteksi *file attachment* pada email yang berpotensi berbahaya

1.3 Maksud dan Tujuan

Maksud dari pembuatan aplikasi ini yaitu untuk mengembangkan dan mengimplementasikan aplikasi *chrome extension* yang inovatif dan berfokus pada deteksi email penipuan.

Adapun tujuan dari pengembangan aplikasi ini, yaitu:

1. Menambahkan kompatibilitas untuk platform pengelola email yang sering dipakai dan platform pengelola email private
2. Mempermudah pengguna mengidentifikasi email yang beresiko penipuan dengan memberikan *alert* dan *click prevention*
3. Menambahkan fitur deteksi terhadap *file attachment* yang berpotensi berbahaya sehingga dapat mencegah ancaman dan melindungi pengguna dari potensi risiko keamanan yang serius

1.4 Batasan Masalah

Aplikasi yang penulis kembangkan, tidak luput dari beberapa masalah yang menjadi batasan dalam pengembangan aplikasi ini, yaitu:

1. Aplikasi ini berbasis *Chrome extension.*
2. Aplikasi hanya bisa berjalan jika pengguna mengakses website dari platform pengelola email yang support, seperti RoundCube, dan Gmail.
3. Pengguna harus terkoneksi ke internet untuk bisa menggunakan aplikasi.
4. Aplikasi menggunakan HTML dan Javascript sebagai client-side, dan server-side yang terpisah menggunakan PHP dengan Laravel Framework.
5. Fokus utama pengembang yaitu scan email otomatis, pemberitahuan langsung, dan integrasi mudah. Terdapat juga fitur lain seperti Analisis tautan dan lampiran, pembaruan berkala, dan Panduan Keamanan.
6. Aplikasi hanya memberikan output berupa saran aksi, pencegahan pengguna untuk mengakses link atau attachment yang di indikasi berbahaya oleh aplikasi, dan beberapa informasi mengenai email seperti ikhtisar pengirim email, domain, security layer, dan lainnya.

1.5 Metodologi Penelitian

Jenis penelitian yang penulis gunakan yaitu survey karena jenis penelitian ini diaplikasikan guna mendapatkan informasi maupun data mengenai penggunaan email, pengalaman percobaan penipuan email, dan pengalaman menggunakan platform pengelola email. Dari data survey tersebut, penulis bisa jadikan patokan untuk pengembangan aplikasi dan juga untuk meningkatkan keamanan email disuatu platform pengelola email.

1.5.1 Metode Pengumpulan Data

1. Sumber Data

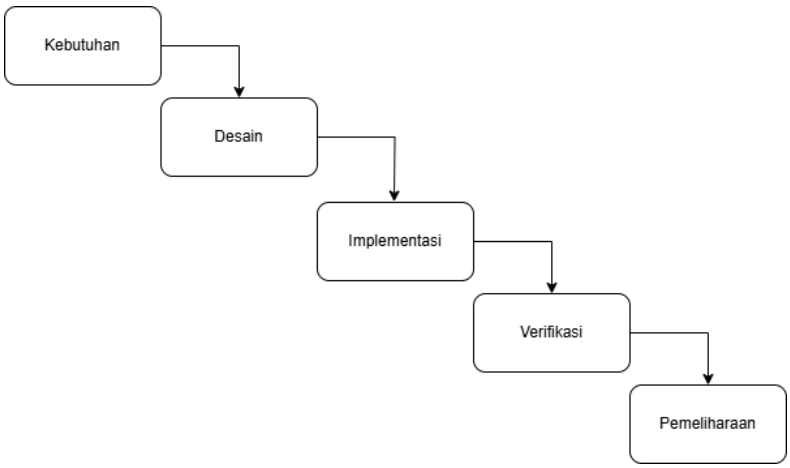
   Data yang digunakan untuk dianalisis bersumber dari *DOM query* yang di lakukan oleh aplikasi menggunakan library Chrome Extension API. Data yang diperoleh merupakan link (jika ada), domain, security layer, text (jika ada), file attactment (jika ada), dll.

2. Kuesioner

   Data kuesioner berupa nama, email, rentang usia, pekerjaan, dan beberapa pertanyaan terkait untuk menunjang pembangunan aplikasi. Data email kemudian digunakan untuk mengirim aplikasi versi alpha kepada test user.

1.5.2 Metode Pembangunan Perangkat Lunak

Dalam Pembangunan aplikasi, penulis menggunakan metode *waterfall*. Alur dari metode ini dapat dilihat pada gambar 1.1.



Gambar 1.1 Metode waterfall

Berikut adalah proses dari aplikasi yang menggunakan metode *waterfall*:

1. Kebutuhan

   Pemahaman mendalam terhadap kebutuhan pengguna dan kebutuhan sistem. Mengumpulkan data fungsional dan non-fungsional hingga dokumen yang menyongsong pembuatan skripsi ini.

2. Desain

   Rancangan antarmuka yang didesain sedemikian rupa agar memenuhi kebutuhan pengguna dengan proporsi dan color palate yang cukup. Hal ini dapat membantu pengembang dalam menentukan tampilan aplikasi.

3. Implementasi

   Dalam fase ini, source code mulai dibuat. Code menyesuaikan kebutuhan dan seminimalis mungkin. Ini berdampak terhadap efisiensi dan kecepatan respons yang mana menjadi prioritas utama untuk pengguna.

4. Verifikasi

   Verifikasi dilakukan untuk mengetahui tingkat kesiapan dari aplikasi. Jika kemudian kesalahan pada logika aplikasi ditemukan pada fase ini, maka tindakan perbaikan akan segera dilakukan. Pengujian dilakukan pada setiap fungsi yang terdapat pada source code aplikasi.

5. Pemeliharaan

   Setelah aplikasi rampung dan siap untuk digunakan, pemeliharan sangat lah penting untuk terus dilakukan dalam jangka panjang. Mengingat pembaruan berkala merupakan prioritas bagi pengguna. Tak hanya berfokus pada pemeliharaan client-side, server-side juga sangatlah penting.

1.6 Sistematika Penulisan

Sebagai acuan bagi penulis agar penulisan skripsi ini dapat terarah dan tersusum sesuai dengan yang penulis harapkan, maka akan disusun sistematika penulisan sebagai berikut :

BAB 1 PENDAHULUAN

BAB 1 berisi uraian latar belakang masalah, identifikasi masalah, maksud dan tujuan, batasan masalah, metodologi penelitian, tahap pengumpulan data, model pengembangan perangkat lunak dan sistematika penulisan.

BAB 2 TINJAUAN PUSTAKA

BAB 2 akan membahas berbagai konsep konsep dasar dan teori-teori pendukung yang berhubungan dengan pembangunan sistem.

BAB 3 ANALISIS DAN PERANCANGAN SISTEM

BAB 3 akan membahas tentang deskripsi sistem, analisis kebutuhan dalam pembangunan sistem serta perancangan sistem.

BAB 4 IMPLEMENTASI DAN PENGUJIAN SISTEM

BAB 4 ini berisi hasil implementasi analisi dari bab sebelumnya dan perancangan sistem yang dilakukan, serta hasil pengujian sistem untuk mengetahui apakah sistem yang dibangun sudah memenuhi kebutuhan.

BAB 5 KESIMPULAN DAN SARAN

BAB 5 berisi kesimpulan yang diperoleh dari hasil pengujian sistem, serta saran untuk pengembangan sistem yang telah dirancang.