

# Rancang Bangun Sistem Informasi Manajemen Penanganan Kebencanaan

Richard Asmarakandi, Siti Rochimah, dan Agus Budi Raharjo  
Departemen Teknik Informatika, Institut Teknologi Sepuluh Nopember (ITS)  
*e-mail: siti@its.ac.id*

**Abstrak**—Indonesia merupakan salah satu negara yang secara geografis terletak di kawasan Ring of Fire atau Cincin Api Pasifik, dan karenanya Indonesia merupakan salah satu negara yang sering dilanda bencana alam seperti gempa bumi, letusan gunung berapi hingga tsunami. Untuk penanggulangan, pemerintah berupaya membangun berbagai fasilitas pelayanan publik seperti membentuk pos penyimpanan logistik kebutuhan, pos pengungsian, hunian sementara, masjid darurat dan beberapa fasilitas publik lainnya. Agar persebaran fasilitas pelayanan publik ini merata, pemerintah seringkali menempatkan tersebar di beberapa titik. Untuk penyesuaian kebutuhan setiap pos, disediakan catatan untuk mencatat data tentang pos yang sedang berjalan, seperti daftar penyintas, daftar relawan, daftar barang-barang logistik dan keuangan pos. Peningkatan efektifitas pencatatan menjadi landasan utama dalam tugas akhir ini dalam bentuk sebuah perangkat lunak yang digunakan untuk mencatat informasi penanganan kebencanaan. Hasil dari tugas ini adalah sebuah perangkat lunak yang secara efektif dapat mencatat segala keperluan penanganan kebencanaan dan kemudahan akses informasi bagi seluruh partisipan penanganan kebencanaan. Sejalan dengan peningkatan efektifitas pencatatan data, diharapkan dengan dibuatnya tugas akhir ini, tingkat pemerataan kebutuhan juga ikut meningkat dan dapat mempercepat pemulihan pasca kebencanaan.

**Kata Kunci**— Bencana Alam, Fasilitas Pelayanan Publik, Pencatatan Data, Perangkat Lunak, Pemulihan Pasca Kebencanaan.

## I. PENDAHULUAN

INDONESIA merupakan negara kepulauan dengan tingkat risiko bencana yang sangat tinggi akibat kondisi geografis, demografis, sosiologis, dan historisnya. Bencana alam seperti letusan gunung berapi, banjir, dan tanah longsor telah menjadi ancaman yang sering terjadi. Sebagai contoh, letusan Gunung Semeru pada Desember 2021 dan letusan Gunung Merapi di Yogyakarta yang telah terjadi lebih dari 80 kali sejak tahun 1600 hingga 2010. Selain itu, banjir besar juga sering terjadi di wilayah DKI Jakarta, seperti yang tercatat pada tahun 1621, 1654, 1918, 1942, 1976, 1996, dan 2002. Dengan risiko yang demikian besar, berbagai upaya telah dilakukan oleh Badan Nasional Penanggulangan Bencana (BNPB), seperti penyediaan bantuan logistik, pengelolaan tim penanganan bencana, serta pengembangan perangkat lunak untuk mendukung penanggulangan bencana.

BNPB telah meluncurkan berbagai aplikasi seperti InaRISK, InaWARE, InaSAFE, dan H-MHEWS. Aplikasi InaRISK, misalnya, membantu pemerintah dan masyarakat dalam memantau indeks risiko bencana serta memberikan informasi terkait ancaman bencana, kerentanan, kapasitas, dan risiko. Sementara itu, InaWARE digunakan oleh institusi terkait untuk mengelola data kejadian bencana, InaSAFE menyediakan skenario dampak bencana untuk perencanaan yang lebih baik, dan H-MHEWS memantau kejadian bencana

secara operasional. Meskipun aplikasi-aplikasi ini memberikan banyak manfaat, proses penanganan bencana di Indonesia masih menghadapi berbagai tantangan, termasuk pendokumentasian alokasi sumber daya yang masih manual, sehingga rawan kesalahan dan ketidakefisienan.

Dalam era digitalisasi, pengelolaan informasi kebencanaan dapat ditingkatkan melalui sistem berbasis teknologi. Oleh karena itu, tugas akhir ini bertujuan untuk merancang sebuah aplikasi Sistem Informasi Manajemen Penanganan Bencana berbasis Next.js. Aplikasi ini akan menyimpan informasi kebutuhan logistik dan relawan, memudahkan pengalokasian sumber daya secara efektif, serta meningkatkan efisiensi dalam penanganan bencana.

Rumusan masalah yang menjadi fokus penelitian ini meliputi bagaimana menggali kebutuhan penanganan kebencanaan dan pemetaan logistik yang tepat, bagaimana membangun Sistem Informasi Manajemen Penanganan Bencana, serta bagaimana menguji kesesuaian informasi yang dibangun dengan kebutuhan penanganan kebencanaan. Penelitian ini memiliki beberapa batasan untuk memastikan fokus dan pencapaian tujuan, yaitu sistem dikembangkan menggunakan kerangka kerja Next.js, basis data yang digunakan adalah PostgreSQL, fitur pemetaan diimplementasikan menggunakan Google Maps API, dan sistem mengasumsikan bahwa pengguna memiliki akses internet.

Tujuan dari penelitian ini adalah membangun aplikasi Sistem Informasi Manajemen Penanganan Bencana yang mampu menyediakan informasi kebutuhan logistik dan relawan secara terstruktur, sehingga dapat dimanfaatkan oleh pihak terkait untuk meningkatkan efisiensi penanganan bencana. Penelitian ini diharapkan dapat memberikan manfaat berupa membantu pengoordinasian logistik dan relawan secara lebih terstruktur, mempermudah manajemen sumber daya selama proses penanganan bencana, serta meningkatkan efektivitas dan efisiensi dalam penanganan kebencanaan, sehingga dapat meminimalkan dampak negatif bagi masyarakat terdampak.

## II. TINJAUAN PUSTAKA

### A. Next JS

NextJS adalah sebuah kerangka kerja React.js yang dikembangkan oleh Vercel dan kerangka kerja yang populer dikarenakan salah satu fiturnya yang dapat melakukan server-side rendering. Tiga keunggulan utama NextJS adalah peningkatan pengalaman pengguna, performa luar biasa dan pengembangan yang sederhana dan tidak terlalu rumit. Para pengembang React seringkali menggunakan NextJS untuk membantu meringankan pekerjaan mereka, karena NextJS telah menyediakan berbagai fitur penting yang tidak disediakan oleh ReactJS dan harus dikembangkan sendiri.

Salah satu contohnya adalah perihal routing atau pemetaan [1]. Tak hanya itu, pada laman resmi NextJS dijelaskan bahwa kerangka kerja NextJS mempunyai kemampuan untuk melakukan pengambilan data sesuai dengan kebutuhan, karena dapat dilakukan melalui klien maupun dari server. Pengambilan data melalui klien biasa disingkat CSR atau Client-Side Rendering dilakukan menggunakan `useEffect` pada React Hooks, lalu data dari API akan diambil setiap kali pengguna melakukan request halaman kepada server. Sedangkan pengambilan data melalui server biasa disingkat SSR atau Server-Side Rendering menggunakan fungsi khusus bernama `getServerSideProps` untuk melakukan pengambilan data dari sisi server. Fungsi `getServerSideProps` hanya berjalan pada sisi server dan tidak akan pernah berjalan pada sisi klien. Pengambilan data secara SSR cocok digunakan ketika aplikasi membutuhkan data yang dibutuhkan ketika klien melakukan request ke server.

### B. PostgreSQL

PostgreSQL atau postgres adalah sebuah sistem manajemen basis data yang bersifat open source yang bermula sebagai proyek penelitian University of California di Berkeley. PostgreSQL awalnya dirilis dibawah lisensi BSD, namun sekarang telah menggunakan The PostgreSQL License (TPL). PostgreSQL memiliki berbagai macam fitur dengan kelas Perusahaan, seperti SQL Windowing Functions, Common Table dan Recursive Common Table Expressions hingga Streaming Replication. Fitur-fitur tersebut jarang ditemukan pada basis data open source lainnya, tetapi sangat umum pada versi terbaru dari basis data pada Perusahaan tertentu seperti Oracle, SQL Server dan DB2. Salah satu keunggulan utama PostgreSQL dibandingkan dengan basis data lain, adalah kemampuannya untuk diperluas dengan mudah tanpa memerlukan kompilasi kode. PostgreSQL tidak hanya dilengkapi dengan fitur-fitur canggih, tetapi juga mampu menjalankan semua fitur tersebut dengan sangat efisien. Bahkan, dalam berbagai jenis beban kerja, PostgreSQL sering kali menunjukkan kinerja yang lebih baik dibandingkan banyak basis data lainnya.

PostgreSQL tidak hanya sekadar sebuah basis data, tetapi juga merupakan platform aplikasi yang memberi pengguna kemampuan untuk menulis prosedur tersimpan dan fungsi dalam berbagai bahasa pemrograman. Baik bahasa bawaan maupun yang diaktifkan melalui ekstensi, seperti PL/Perl, PL/Python, dan PL/JavaScript, semuanya dapat digunakan. Fitur ini memungkinkan penyelesaian masalah dengan pendekatan yang lebih spesifik dan fungsional. Selain itu, PostgreSQL mendukung penulisan fungsi agregat dan tipe data kustom, yang dapat diterapkan dalam kolom tabel atau pernyataan SQL. Kemampuan ini memadukan kekuatan SQL dengan bahasa lain, menawarkan fleksibilitas dan kinerja yang lebih baik dibandingkan banyak basis data lainnya. Dengan dukungan untuk tipe kustom dan ekstensi pihak ketiga, PostgreSQL semakin meningkatkan performa dan efisiensi, menjadikannya unggul dalam berbagai tugas pengolahan data.

### C. API Pemetaan

API atau Application Programming Interface adalah sebuah antarmuka yang disediakan oleh sebuah program perangkat lunak kepada perangkat lunak yang lain, kepada manusia serta dalam kasus pemrograman API website kepada

sdunia melalui internet. API dirancang untuk dapat berfungsi dengan program lain, sebagian besar dirancang agar dapat dipahami serta digunakan oleh manusia yang tengah membuat perangkat lunak lain [2]. API memiliki berbagai implementasi, dan salah satu yang umum digunakan adalah API untuk peta seperti Google Maps.

Google Maps merupakan sebuah layanan gratis milik google yang sudah digunakan secara luas. Fitur google map dapat dimasukkan ke dalam website yang dibuat menggunakan Google Maps API yang merupakan sebuah library javascript. Menggunakan Google Maps API akan mempermudah dalam membangun aplikasi peta digital, sehingga pengembang aplikasi dapat lebih fokus terhadap data-data yang akan dikelola. Untuk data peta dunia akan disediakan oleh Google melalui Google Maps API itu sendiri. Namun ketika ingin mengimplementasikan Google Maps API ke dalam perangkat lunak lain yang dibuat, diperlukan biaya sesuai dengan fitur yang disediakan. Hal tersebut bisa dilihat melalui website resmi Google Maps.

### D. Siklus Hidup Pengembangan Perangkat Lunak

SDLC merupakan singkatan dari System Development Life Cycle atau yang dalam Bahasa Indonesia diartikan sebagai siklus hidup pengembangan perangkat lunak. SDLC adalah siklus pengembangan sistem yang memiliki fungsi menggambarkan tahapan-tahapan utama dan langkah-langkah dari setiap tahapan yang terbagi menjadi lima secara garis besar, yaitu analisis, desain, implementasi, pengujian dan pemeliharaan. Kegiatan dalam SDLC dapat dijelaskan dalam tujuan dan hasil kegiatannya. Terdapat beberapa jenis SDLC, tiga diantaranya adalah waterfall, throwaway prototyping, dan SCRUM.

Pendekatan tradisional pada pengembangan perangkat lunak dapat diilustrasikan melalui model waterfall yang selalu di tes pada setiap waktu dan termasuk model yang mudah dipahami. Model waterfall adalah sebuah model statis dan model yang menggunakan pendekatan pengembangan sistem secara linier dan berurutan, menyelesaikan satu aktivitas sebelum melakukan aktivitas lainnya. Model waterfall memiliki beberapa tahapan, yaitu: analisis kebutuhan, desain, implementasi, pengujian, serta operasi dan pemeliharaan. Model waterfall biasanya memiliki tujuan yang berbeda pada setiap tahapan pengembangannya. Setiap kali sebuah tahapan telah selesai dikembangkan, pengembangan akan lanjut ke tahapan berikutnya dan tidak ada kesempatan untuk mundur dan kembali ke tahapan sebelumnya.

Systematic Customer Resolution Unraveling Meeting atau yang biasa disebut dengan SCRUM merupakan metode agile yang paling sering digunakan saat melakukan pengembangan perangkat lunak. Metode agile adalah sebuah metodologi yang terus menerus melakukan iterasi dan uji coba selama proses pengembangan. Pada dasarnya, SCRUM adalah sebuah kerangka kerja yang ringan, yang menyediakan langkah-langkah untuk mengelola dan mengontrol perangkat lunak dan proses pengembangan produk. Alur kerja SCRUM terdiri dari kolaborasi erat antara tim SCRUM dan SCRUM master dengan pemilik produk melalui iterasi berkelanjutan dari pengembangan sebuah perangkat lunak. Peran utama SCRUM master adalah menghilangkan kesulitan. Untuk tim SCRUM sendiri terdiri dari tim pengembang, penguji dan ahli

lain dari berbagai bidang yang mengarah pada produk akhir yang serbaguna dan inovatif sesuai dengan kepuasan pengguna. SCRUM fokus pada produktivitas melalui komunikasi dan perencanaan yang menyediakan kebebasan pada tim untuk menemukan solusi pada engineer. Sprint merupakan blok SCRUM terkecil yang memiliki tim kecil yang mengerjakan tugas yang diberikan. Sprint berlangsung dari satu hingga tiga minggu. Tugas pada sprint ditentukan oleh sprint backlog. Sprint backlog adalah dokumentasi dari semua kebutuhan pada sprint untuk dikerjakan.

Product backlog adalah sebuah daftar kebutuhan dimana ditentukan oleh pemilik produk dan disebut dengan user stories. Product backlog dipecah menjadi sprint backlog, bersamaan dengan sprint planning dimana mengandung metode untuk menyelesaikan sprint. Pada setiap penghujung hari, akan diadakan daily SCRUM yang ditujukan pada perkembangan setiap tugas yang diberikan pada hari itu. Tujuan dari setiap sprint adalah untuk menghasilkan produk berpotensi yang dapat dikirim. Di akhir setiap sprint, terdapat tinjauan sprint yang berlangsung dengan pemilik produk untuk menunjukkan peningkatan produk yang dapat dikirim. Tujuan dari sprint tidak dapat diubah ketika sebuah sprint berjalan, namun dengan setiap peningkatan, pemilik produk dapat menambahkan fitur baru ke proyek yang tidak ditentukan sebelumnya [3].

Model prototyping adalah pengembangan perangkat lunak yang diawali dengan penggalan kebutuhan sistem, lalu dilanjutkan dengan membuat prototipe dan evaluasi dari pengguna. Terdapat dua jenis pendekatan pada model prototyping, yaitu throw-away prototyping atau rapid prototyping dan evolutionary prototyping. Pada proses evolutionary prototyping, sistem dikembangkan tanpa mengetahui spesifikasi sistem yang benar pada awal pengembangan mengakibatkan verifikasi tidak dapat dilakukan karena tidak adanya spesifikasi sistem. Sedangkan pada proses throw-away prototyping, spesifikasi awal dari sistem diketahui di awal pengembangan, sehingga proses ini ditujukan untuk mengurangi risiko kebutuhan yang tidak terpenuhi.

Tahapan dalam proses throw-away prototyping dimulai dengan menentukan kebutuhan pengguna dengan cara mewawancarai pengguna untuk mendapatkan ide tentang apa yang diinginkan oleh pengguna dari sistem yang dikembangkan. Tahap selanjutnya adalah pembuatan prototipe. Kemudian setelah prototipe dibuat, berikutnya adalah tahap evaluasi. Pada tahap ini pengguna diperkenalkan prototipe yang telah dibuat guna mendapatkan pendapat dari pengguna. Apabila prototipe diterima, maka lanjut ke tahap berikutnya, sedangkan apabila terdapat perbaikan maka kembali lagi ke tahapan menggali kebutuhan pengguna dan begitu seterusnya hingga prototipe dapat diterima oleh pengguna. Lalu tahap terakhir merupakan implementasi sistem, yaitu para pemrogram melakukan pemrograman dan memanfaatkan prototipe sebagai pedoman untuk mengembangkan sistem yang operasional [4].

### III. METODOLOGI

#### A. Metode Perancangan

Tahap pertama dari alur pengembangan dimulai dengan identifikasi kebutuhan awal. Pada tahap ini, penulis mencoba

mengumpulkan segala informasi tentang kebutuhan apa saja yang sekiranya perlu diimplementasikan dalam Sistem Informasi Manajemen Penanganan Kebencanaan. Informasi yang dikumpulkan oleh penulis berupa data pos kebencanaan selama penulis melaksanakan kegiatan KKN. Dari data tersebut, penulis menjabarkan kebutuhan fungsionalitas yang kemungkinan akan dibutuhkan sebagai kebutuhan sistem ketika sistem dibangun, seperti jenis data yang akan dicatat pada sistem sesuai dengan data yang telah didapatkan, beserta semua fungsi penunjangnya dapat dilihat pada Gambar 1.

Setelah memperkirakan kebutuhan yang diperlukan, selanjutnya adalah perancangan dan pembuatan prototipe. Tujuan perancangan dan pembuatan prototipe ini adalah memperjelas kebutuhan dan mencoba menggambarkan ketidakpastian di dalam sistem. Prototipe yang dibuat dilakukan dengan pendekatan sederhana yang memungkinkan untuk pembuatan prototipe secara cepat. Aspek teknis dan estetika yang membutuhkan waktu yang lama dihindari pada tahap ini, karena pada akhirnya prototipe akan dibuang, mengingat prototipe disini hanya bertujuan untuk memperjelas kebutuhan dan menggambarkan ketidakpastian di dalam sistem, bukan sebagai landasan untuk dilanjutkan ke fase pengembangan.

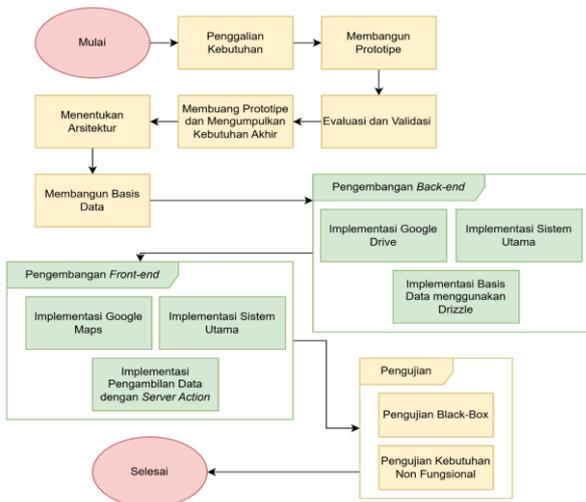
Tahap berikutnya adalah pengumpulan feedback dari pengguna atau stakeholder mengenai prototipe yang telah dibuat. Stakeholder pada Sistem Informasi Manajemen Penanganan Kebencanaan ini adalah badah MKPI ITS. Tahap ini dilakukan untuk mengembangkan ide yang telah dituangkan ke dalam prototipe yang telah dibuat sebelumnya. Di samping itu, tahap pengumpulan feedback ini juga dilakukan untuk meminimalisir kesalahan yang mungkin terjadi pada prototipe yang telah dikembangkan.

Setelah semua feedback telah dikumpulkan, selanjutnya adalah memperbaiki dan memperjelas kembali kebutuhan sistem, hingga akhirnya didapatkan kebutuhan akhir dari sistem yang akan dikembangkan. Kebutuhan akhir ini merupakan hasil kebutuhan yang paling akurat karena telah melalui beberapa proses identifikasi dan pengumpulan feedback sebelumnya, dan oleh karenanya dilakukan dokumentasi kebutuhan akhir pada tahap ini. Pada pengerjaan tugas akhir ini, dokumentasi disimpan dalam bentuk use case scenario. Setelah kebutuhan akhir terbentuk dan didokumentasikan, maka prototipe yang tadinya dikembangkan selanjutnya dibuang dan tidak digunakan sebagai dasar pengembangan produk akhir, karena prototipe yang dibangun tidak memiliki kualitas yang memadai.

Setelah semua tahap selesai dilakukan, maka tahap utama pengembangan dapat dimulai berdasarkan use case scenario yang telah didokumentasikan pada tahap sebelumnya. Setelah pengembangan dilakukan, akan dilakukan pengujian fungsionalitas sistem dengan metode Blackbox testing. Metode ini memungkinkan pengujian dilakukan tanpa melihat kode sumber yang telah dikembangkan, melainkan hanya melakukan proses uji berdasarkan masukan dan keluaran yang diharapkan pada sistem. Selanjutnya dilakukan pengujian untuk kebutuhan non-fungsional sistem menggunakan bantuan perangkat lunak Apache JMeter, ZAPProxy, Qualys SSL Labs dan htop.

#### B. Bahan dan Peralatan yang Digunakan

Spesifikasi perangkat keras yang digunakan dalam



Gambar 1. Alur pengembangan sistem informasi manajemen penanganan kebencanaan.

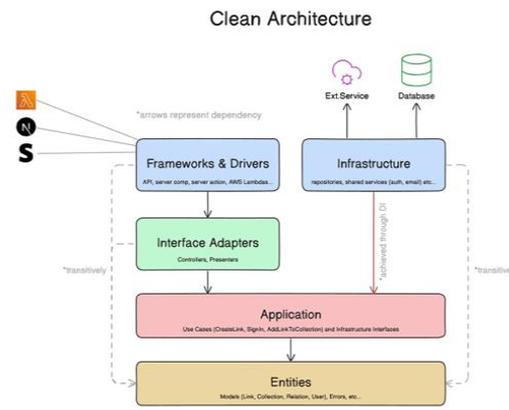
membangun Sistem Informasi Manajemen Penanganan Kebencanaan meliputi CPU 11th Gen Intel(R) Core(TM) i3-1115G4 @ 3.00GHz dengan kecepatan 2995 Mhz, 2 Core, dan 4 Logical Processor, GPU Intel(R) UHD Graphics, penyimpanan sebesar 256GB menggunakan NVMe SSD PCIe Gen3, memori RAM 8GB DDR4-3200 onboard, dan monitor berukuran 14 inci FHD (1920x1080) dengan teknologi IPS-Level. Perangkat keras ini dipilih untuk mendukung kebutuhan komputasi dalam pengembangan sistem secara optimal.

Sedangkan spesifikasi perangkat lunak yang digunakan meliputi sistem operasi Microsoft Windows 11 Home Single 23H2, lingkungan pengembangan berbasis Node.js v.20.10.0, serta PostgreSQL sebagai basis data. Bahasa pemrograman yang digunakan adalah JavaScript dengan dukungan berbagai pustaka seperti Next.js, Zod, SWR, React Hook Form, React Hot Toast, Google APIs Drive, Google APIs OAuth2, Lucia Auth, Drizzle, VisGL React Google Maps, Inversify, Moment, Node PG, React Icons, React Select, Sweet Alert 2, dan Zustand. Kombinasi perangkat lunak ini mendukung pengembangan sistem yang modern, efisien, dan fleksibel dapat dilihat pada Gambar 2.

### C. Implementasi

Arsitektur perangkat lunak yang digunakan adalah Clean Architecture, yang memisahkan logika bisnis inti dari dependensi eksternal. Pendekatan ini mengadopsi lima lapisan, yaitu Framework and Drivers, Interface Adapters, Application, Entities, dan Infrastructure, dengan Next.js sebagai teknologi utama. Lapisan Framework and Drivers berfungsi sebagai pintu masuk data dan pengelolaan tampilan UI, sementara Interface Adapters bertugas menghubungkan framework dengan logika aplikasi melalui komponen seperti Controller dan Presenter. Lapisan Application mengelola logika kasus penggunaan aplikasi, seperti validasi, otorisasi, dan eksekusi kebutuhan bisnis. Lapisan Entities berfungsi memodelkan aturan bisnis dengan entitas seperti User dan Item, sedangkan Infrastructure mendukung kebutuhan teknis seperti pengelolaan basis data menggunakan PostgreSQL dan Drizzle ORM.

Struktur hierarki proyek diatur dengan mengikuti pola penempatan berkas bawaan dari Next.js. Pada tingkat direktori paling luar, terdapat folder seperti “.next” untuk



Gambar 2. Diagram clean architecture.

menyimpan halaman statis, berkas “.env” untuk konfigurasi environment, dan “next.config.mjs” untuk pengaturan utama proyek. Penambahan struktur hierarki detail dilakukan untuk mempermudah pembacaan dan navigasi komponen proyek, sehingga mendukung pengembangan yang lebih terstruktur dan efisien.

## IV. HASIL DAN PEMBAHASAN

Sebagai hasil dari proses perancangan dan pengembangan sistem, aplikasi Sistem Informasi Manajemen Penanganan Kebencanaan telah berhasil diimplementasikan dan dapat diakses melalui tautan <https://simp-kebencanaan.ichadas.com>. Selanjutnya, pengujian dilakukan untuk memastikan bahwa sistem ini telah memenuhi kebutuhan fungsional dan non-fungsional yang telah ditentukan sebelumnya. Pengujian ini meliputi berbagai aspek seperti performa, keandalan, kompatibilitas, dan kegunaan. Pada sub-bab berikut, akan dibahas secara rinci hasil dari berbagai jenis pengujian yang telah dilakukan serta analisis terkait keberhasilan sistem dalam memenuhi kebutuhan.

### A. Pengujian Fungsionalitas

Pengujian fungsionalitas dilakukan dengan metode *Black-box Testing* untuk setiap kebutuhan fungsional sistem berupa penambahan, pembaruan dan penghapusan data. Uji coba dilakukan dengan berbagai macam *test case* dan konfigurasi akses. Hasil pengujian dapat dilihat pada Tabel 1.

### B. Pengujian Kebutuhan Non-Fungsional

Pengujian kebutuhan non-fungsional yang dilakukan mencakup efisiensi kinerja, keandalan, keamanan, kompatibilitas, kegunaan dan portabilitas. Metrik yang digunakan untuk pengujian efisiensi mencakup waktu respon, throughput dan penggunaan sumber daya. Metrik yang digunakan untuk mengukur keandalan mencakup uptime dan waktu pemulihan. Metrik yang digunakan untuk mengukur keamanan mencakup otorisasi dan perlindungan data. Metrik yang digunakan untuk mengukur kompatibilitas mencakup interoperabilitas. Metrik yang digunakan untuk mengukur kegunaan mencakup dan kemudahan pemahaman fitur dan kemudahan penggunaan. Metrik yang digunakan untuk mengukur portabilitas mencakup adaptabilitas. Hasil pengujian kebutuhan non-fungsional dijabarkan sebagai berikut:

Tabel 2.  
Hasil pengujian kebutuhan fungsional sistem informasi manajemen penanganan kebencanaan.

Uji Fitur	Jumlah Skenario Pengujian	Jumlah Skenario Berhasil
Masuk ke Akun	5	5
Dafar Akun	7	7
Keluar Akun	1	1
Menambahkan Kebencanaan	9	9
Menghapus Kebencanaan	5	5
Memperbarui Deskripsi Kebencanaan	11	11
Menambahkan Dokumentasi Kebencanaan	6	6
Memperbarui Dokumentasi Kebencanaan	8	8
Menghapus Dokumentasi Kebencanaan	6	6
Menambahkan Pos Kebencanaan	8	8
Menghapus Pos Kebencanaan	6	6
Menambahkan Akses Pengguna	6	6
Memperbarui Akses Pengguna	8	8
Menghapus Akses Pengguna	5	5
Memperbarui Deskripsi Pos Kebencanaan	9	9
Menambahkan Dokumentasi Pos Kebencanaan	6	6
Memperbarui Dokumentasi Pos Kebencanaan	8	8
Menghapus Dokumentasi Pos Kebencanaan	6	6
Menambahkan Pengumuman Pos Kebencanaan	8	8
Memperbarui Pengumuman Pos Kebencanaan	9	9
Menghapus Pengumuman Pos Kebencanaan	6	6
Menambahkan Acara Pos Kebencanaan	8	8
Memperbarui Acara Pos Kebencanaan	9	9

(Lanjutan)

Tabel 1.  
Hasil pengujian kebutuhan fungsional sistem informasi manajemen penanganan kebencanaan.

Uji Fitur	Jumlah Skenario Pengujian	Jumlah Skenario Berhasil
Menghapus Acara Pos Kebencanaan	6	6
Menambahkan Aktivitas Pos Kebencanaan	9	9
Memperbarui Aktivitas Pos Kebencanaan	9	9
Menghapus Aktivitas Pos Kebencanaan	8	8
Menambahkan Partisipan Pos Kebencanaan	9	9
Memperbarui Partisipan Pos Kebencanaan	9	9
Menghapus Partisipan Pos Kebencanaan	6	6
Menambahkan Permintaan Barang Pos Kebencanaan	10	10
Memperbarui Permintaan Barang Pos Kebencanaan	11	11
Menghapus Permintaan Barang Pos Kebencanaan	9	9
Meninjau Permintaan Barang Pos Kebencanaan	12	12
Mendaftar Pos Kebencanaan	9	9
Memperbarui Pendaftaran Pos Kebencanaan	9	9
Menghapus Pendaftaran Pos Kebencanaan	8	8
Meninjau Pendaftaran Pos Kebencanaan	12	12
Menambahkan Pergerakan Barang Pos Kebencanaan	8	8
Menambahkan Stok Transfer Pos Kebencanaan	8	8
Memperbarui Pergerakan Barang Pos Kebencanaan	9	9
Membatalkan Pergerakan Barang Pos Kebencanaan	8	8
Menambahkan Pergerakan Keuangan Pos Kebencanaan	8	8
Menambahkan Transfer Keuangan Pos Kebencanaan	9	9
Memperbarui Pergerakan Keuangan Pos Kebencanaan	8	8
Membatalkan Pergerakan Keuangan Pos Kebencanaan	8	8

1) Pengujian efisiensi kerja,

Pengujian efisiensi kerja untuk metrik waktu respon dan *throughput* diuji menggunakan aplikasi JMeter. Parameter yang digunakan ketika pengujian adalah kombinasi banyak simulasi pengguna dan periode pengaksesan situs web Sistem Informasi Manajemen Penanganan Kebencanaan. Terdapat juga parameter tetap yaitu iterasi sebanyak 5 kali dan 5 *path* pengujian berupa halaman beranda, halaman daftar kebencanaan, halaman detail kebencanaan, halaman daftar pos kebencanaan dan halaman detail pos kebencanaan. Pengujian pertama dilakukan dengan banyak simulasi

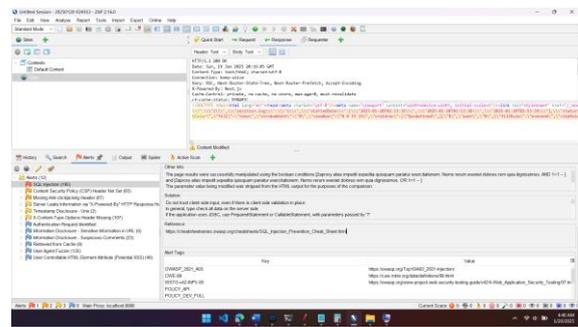
pengguna 10 dan periode *ramp-up period* 1 detik. Dengan kata lain, pengujian akan mengirimkan sebanyak 10 *request* atau permintaan pengguna ke situs web Sistem Informasi Manajemen Penanganan Kebencanaan dalam waktu 1 detik. Setiap simulasi permintaan pengguna, dilakukan berulang hingga 5 kali perulangan. Pengujian kedua dilakukan dengan banyak simulasi pengguna 25 dan *ramp-up period* 1 detik. Pengujian ketiga dilakukan dengan banyak simulasi pengguna 50 dan *ramp-up period* 1 detik. Peningkatan terus

Tabel 3.  
Hasil pengujian metrik *throughput* pada system.

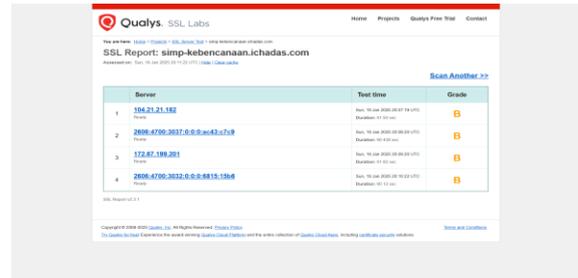
No	Konfigurasi		Repetisi	Throughput (Permintaan /detik)
	Jumlah Simulasi Pengguna	Periode Permintaan		
1	10	1	5	11.9
2	25	1	5	25.3
3	50	1	5	31.1
4	75	1	5	33
5	100	1	5	32.9
6	250	1	5	32.1

Tabel 4.  
Hasil pengujian metrik penggunaan sumber daya pada system.

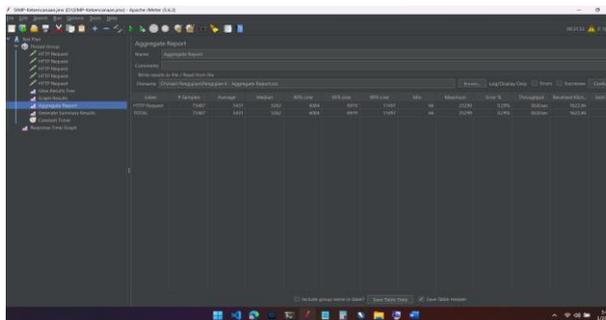
No	Konfigurasi			Penggunaan CPU (%)	Penggunaan Memori RAM (MB)
	Jumlah Simulasi Pengguna	Periode Permintaan	Repetisi		
1	10	1	5	36.93%	309
2	25	1	5	53%	316
3	50	1	5	60%	321
4	75	1	5	61%	322
5	100	1	5	63%	323
6	250	1	5	60%	332



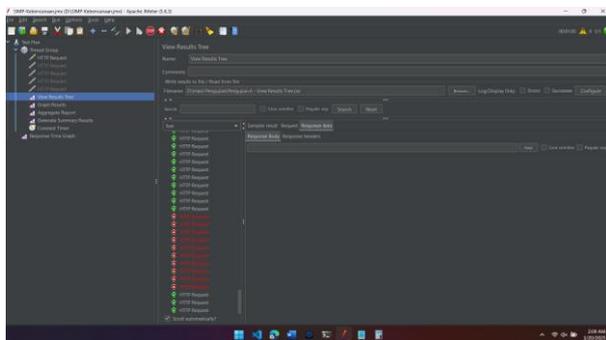
Gambar 3. Hasil pengujian metrik otorisasi menggunakan zap.



Gambar 4. Hasil pengujian metrik perlindungan data menggunakan Qualys SSL Labs.



Gambar 7. Hasil pengujian metrik waktu *uptime* pada sistem.



Gambar 8. Hasil pengujian metrik waktu pemulihan pada system.

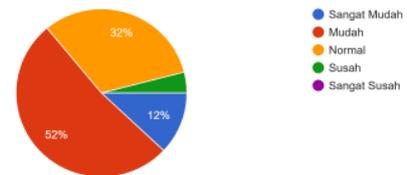
dilakukan pada banyak simulasi pengguna. Hasil pengujian dapat dilihat pada Tabel 2.

Lalu untuk pengujian metrik penggunaan sumber daya pada pengujian efisiensi kinerja, dilakukan dengan bantuan program htop bawaan dari sistem operasi ubuntu. Tabel hasil pengujian efisiensi kinerja dapat dilihat pada Tabel 3.

### 2) Pengujian Keandalan

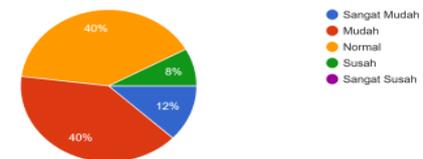
Pengujian keandalan mencakup metrik waktu uptime dan waktu pemulihan. Untuk pengujian metrik waktu uptime, dilakukan dengan bantuan JMeter. JMeter akan melakukan banyak permintaan ke server Sistem Informasi Manajemen Penanganan Kebencanaan secara berulang setiap detik selama 30 menit. Maka nilai uptime adalah banyaknya permintaan yang telah dikirimkan oleh JMeter dan berhasil di respon oleh server, dibandingkan dengan banyaknya

Seberapa mudah Anda menemukan fitur atau informasi yang Anda cari?  
25 responses



Gambar 5. Hasil pengujian metrik kemudahan penggunaan melalui kemudahan menemukan fitur.

Apakah antarmuka sistem mudah dimengerti?  
25 responses



Gambar 6. Hasil pengujian metrik kemudahan penggunaan melalui kemudahan pemahaman antarmuka system.

permintaan total yang dilakukan oleh JMeter. Skenario yang diterapkan untuk pengujian ini adalah, terdapat 150 pengguna yang secara bersamaan akan mengakses 5 url path yang berbeda pada Sistem Informasi Manajemen Penanganan Kebencanaan setiap detiknya. Hasil pengujian pada Gambar 3 menunjukkan bahwa selama 30 menit hanya terjadi 0.29% error, sehingga menciptakan persentase uptime sebesar 99.71%.

Untuk pengujian metrik waktu pemulihan, juga dibantu dengan aplikasi JMeter. JMeter akan mensimulasikan satu pengguna, melakukan satu permintaan setiap detiknya. Lalu proses yang menjalankan Sistem Informasi Manajemen Penanganan Kebencanaan akan dihentikan, lalu dinyalakan kembali. Setiap permintaan pada JMeter yang gagal dihitung sebagai lama waktu pemulihan, karena satu permintaan gagal merepresentasikan satu detik server tidak dapat melayani permintaan. Hasil pengujian pada Gambar 4 menunjukkan

bahwa butuh waktu 10 detik untuk pemulihan server setelah proses mati.

### 3) *Pengujian Keamanan*

Pengujian keamanan mencakup metrik otorisasi dan perlindungan data. Otorisasi diuji menggunakan bantuan aplikasi bernama ZAP. Hasil dari pengujian ini dapat dilihat pada Gambar 5.

Menurut data hasil pengujian, otorisasi sistem dapat bekerja dengan baik, hanya saja perlu ditingkatkan lagi terkait pertahanan dari serangan SQL Injection. Untuk metrik perlindungan data, dilakukan dengan verifikasi enkripsi TLS dengan bantuan perangkat lunak Qualys SSL Labs yang dapat diakses melalui situs web dengan URL [www.ssllabs.com/ssltest](http://www.ssllabs.com/ssltest). Hasil verifikasi enkripsi TLS dapat dilihat pada Gambar 6 menunjukkan bahwa penilaian terhadap proses enkripsi ketika transmisi data bernilai B.

### 4) *Pengujian Ketergunaan*

Pengujian komparabilitas meliputi metrik interoptabilitas yang dilakukan untuk memastikan bahwa sistem dapat berfungsi dengan baik saat diintegrasikan dengan sistem lain, yang dalam implementasi Sistem Informasi Manajemen Penanganan Kebencanaan ini adalah Google Maps. Pengujian dilakukan secara manual, memastikan bahwa setiap fitur Google Maps seperti zoom in, zoom out, drag dan marker berjalan dengan baik pada Sistem Informasi Manajemen Penanganan Kebencanaan. Hasil pengujian menunjukkan bahwa semua fitur yang diuji dapat bekerja dengan baik.

### 5) *Pengujian Ketergunaan*

Pengujian ketergunaan diuji menggunakan metrik kemudahan penggunaan yang melibatkan pengguna secara langsung. Pengujian dilakukan dengan membagikan kuisioner tentang kemudahan penggunaan Sistem Informasi Manajemen Penanganan Kebencanaan. Pada kuisioner, diberikan beberapa poin berupa berbagai macam fitur yang dapat diakses oleh pengguna, sekaligus pengguna diharapkan melakukan eksplorasi terhadap fitur yang telah dijabarkan. Kemudian pengguna menilai tingkat kemudahan mengakses fitur yang mereka cari dan juga menilai tingkat kemudahan memahami tampilan antarmuka Sistem Informasi Manajemen Penanganan Kebencanaan. Hasil pengujian metrik ini dapat dilihat pada Gambar 7-8. Pada kedua pertanyaan apabila dijumlahkan, responden yang mengaku bahwa tampilan antarmuka pengguna dan kemudahan akses fitur terbilang normal dan mudah sebanyak 188% dari 200%, sehingga bisa dikatakan 97% orang mengaku bahwa mereka

tidak merasa kesulitan menggunakan Sistem Informasi Manajemen Penanganan Kebencanaan.

### 6) *Pengujian Portabilitas*

Pengujian portabilitas meliputi metrik adaptabilitas, yaitu meliputi kemampuan sistem untuk berjalan di berbagai macam sistem operasi. Pengujian ini dilakukan pada 3 sistem operasi, yaitu Windows, Linux dan juga Mac OS. Pengujian dilakukan secara manual dengan cara menguji berbagai fitur yang ada di Sistem Informasi Manajemen Penanganan Kebencanaan di beberapa perangkat dengan sistem operasi yang berbeda-beda. Hasilnya, Sistem Informasi Manajemen Penanganan Kebencanaan dapat bekerja di ketiga sistem operasi, yaitu Windows 11, Bodhi Linux dan juga Mac OS.

## V. KESIMPULAN

Berdasarkan hasil penelitian yang dilakukan, didapatkan Kesimpulan penggalan kebutuhan yang didasarkan pada data yang penulis dapatkan dari mengikuti kegiatan relawan kebencanaan Gunung Semeru pada tahun 2021 lalu secara langsung telah menghasilkan rincian kebutuhan sistem yang kemudian dapat diaplikasikan secara langsung terhadap data dan juga relevan dengan informasi yang didapatkan pada data. Pengujian kebutuhan fungsional sistem menggunakan metode black-box telah dilakukan dan keseluruhan fungsionalitas sistem telah diuji dan dapat bekerja dengan semestinya, sehingga sistem telah berhasil dikembangkan dengan baik. Pengujian kebutuhan non-fungsional sistem telah dilakukan dan telah memenuhi persyaratan kebutuhan sistem.

## DAFTAR PUSTAKA

- [1] V. K. R. Ballamudi, K. Lal, H. Desamsetti, and S. Dekkati, "Getting started modern web development with next.js: An indispensable react framework," *Digitalization & Sustainability Review*, vol. 1, no. 1, pp. 1–11, Mar. 2021.
- [2] B. Jin, S. Sahni, and A. Shevat, *Designing Web APIs: Building APIs That Developers Love*, 1st ed. America: O'Reilly Media, Inc., 2018.
- [3] A. Srivastava, S. Bhardwaj, and S. Saraswat, "SCRUM model for agile methodology," in *International Conference on Computing, Communication and Automation, ICCCA 2017*, Greater Noida, India: IEEE, Dec. 2017, pp. 864–869. doi: 10.1109/CCAA.2017.8229928.
- [4] A. Wibowo and A. Azimah, "Rancang bangun sistem informasi penjaminan mutu perguruan tinggi menggunakan metode throwaway prototyping development," in *Seminar Nasional Teknologi Informasi dan Multimedia*, Yogyakarta: STMIK AMIKOM, Feb. 2016, pp. 103–108.