

IMPLEMENTASI METODE *DEVOPS* PADA PENGEMBANGAN APLIKASI PENGAJIAN KEBUTUHAN PASCA BENCANA BERBASIS WEB

Muhammad Fahmi¹⁾, Ahmad Abul Khair²⁾, dan Dewi Mustika Sari³⁾

TI¹, TI², STMIK Widya Cipta Dharma

TI³, STMIK Widya Cipta Dharma

Jl. M. Yamin No. 25, Samarinda, 75123

E-mail: mfahmi@wicida.ac.id¹⁾, abul@wicida.ac.id²⁾, dewimustika15@gmail.com³⁾

ABSTRAK

DevOps merupakan tren baru yang muncul dari benturan dua tren lama yaitu *Agile Systems* dan *Agile Operations*. Metode ini merupakan kolaborasi antara staf pengembang dan staf operasi di setiap tahap siklus DevOps. Dev atau *development* berarti pengembang yang terlibat dalam fase produksi, Ops atau *operation* adalah istilah yang digunakan untuk administrator sistem, pakar keamanan dan berbagai disiplin ilmu serta posisi lainnya. DevOps adalah praktik operasi dan pengembang yang berpartisipasi bersama dalam setiap fase siklus hidup.

Selama ini BPBD Kota Samarinda masih melakukan Pengkajian Kebutuhan Pasca Bencana dengan cara manual yaitu dengan mengumpulkan informasi melalui form dari pihak-pihak yang terdampak. Dalam menghadapi permasalahan tersebut maka perlu memanfaatkan sebuah aplikasi tersendiri yang merupakan pengembangan dari aplikasi jitungasna yang di kelola oleh BNPB yang mampu mengelola data-data dari informasi pihak-pihak yang terdampak serta melakukan perhitungan secara otomatis sehingga penyelenggaraan kegiatan rehabilitasi dan rekonstruksi dapat terlaksana tepat waktu.

Pengembangan aplikasi jitungasna ini dibuat dengan menggunakan metode pengembangan sistem *DevOps*. Metode *DevOps* sangat adaptasi terhadap pengembangan perangkat lunak sehingga dapat dengan cepat diimplementasikan serta mendapatkan perubahan sesuai dengan kebutuhan sehingga dapat dilakukan *continuous improvement* pada aplikasi.

Berdasarkan hasil penelitian yang dilakukan, disimpulkan bahwa Pengembangan Aplikasi Pengkajian Kebutuhan Pasca Bencana Berbasis Web telah berhasil di bangun atau diimplementasikan dengan baik, sehingga dapat menjadi bahan pertimbangan bagi instansi BPBD Kota Samarinda dalam menerapkan teknologi informasi di dalam lingkup kerjanya.

Kata Kunci: *DEVOPS, OOAD, UML, WEBAPP, BPBD*

1. PENDAHULUAN

Dalam pengembangan suatu perangkat lunak khususnya aplikasi terdapat beberapa hal yang harus diperhatikan yaitu, sumber daya manusia yang terkait dalam pengembangan perangkat lunak, estimasi biaya serta metode yang diterapkan dalam proses pengembangan aplikasi tersebut. Terdapat banyak metode pendekatan pengembangan perangkat lunak salah satunya adalah DevOps.

DevOps merupakan tren baru yang muncul dari benturan dua tren lama yaitu *Agile Systems* dan *Agile Operations*. Metode ini merupakan kolaborasi antara staf pengembang dan staf operasi di setiap tahap siklus DevOps. Dev atau *development* berarti pengembang yang terlibat dalam fase produksi, Ops atau *operation* adalah istilah yang digunakan untuk administrator sistem, pakar keamanan dan berbagai disiplin ilmu serta posisi lainnya. DevOps adalah praktik operasi dan pengembang yang berpartisipasi bersama dalam setiap fase siklus hidup.

Bencana adalah peristiwa atau rangkaian peristiwa yang mengancam dan mengganggu kehidupan dan penghidupan masyarakat yang disebabkan baik oleh faktor alam atau faktor non alam maupun faktor manusia sehingga

mengakibatkan timbulnya korban jiwa, kerusakan lingkungan, kerugian harta benda dan dampak psikologis. Sedangkan rehabilitasi dan rekonstruksi sebagai bagian dari penyelenggaraan penanggulangan bencana memerlukan proses penilaian atas kerusakan dan kerugian serta kebutuhan yang bersifat komprehensif baik aspek fisik maupun aspek kemanusiaan. Semuanya dilakukan dengan prinsip dasar membangun yang lebih baik dan pengurangan resiko bencana dan diwujudkan dalam bentuk rencana aksi rehabilitasi dan rekonstruksi pascabencana.

Pengkajian Kebutuhan Pascabencana yang selanjutnya disebut Jitungasna adalah suatu rangkaian kegiatan pengkajian dan penilaian akibat, analisi dampak, perkiraan kebutuhan dan rekomendasi awal terhadap strategi pemulihan yang menjadi dasar penyusunan Rencana Rehabilitasi dan Rekonstruksi Pascabencana. Pengkajian Kebutuhan Pascabencana atau disebut Jitungasna merupakan instrument dalam menyusun kebijakan program dan kegiatan rehabilitasi dan rekonstruksi yang berlandaskan pada informasi yang akurat dari pihak yang terdampak bencana.

Pada aplikasi mobile inaRisk terdapat fitur Jitungasna yaitu fitur yang dapat digunakan untuk melaporkan kegiatan

atau informasi terkait pengurangan resiko bencana atau informasi kebencanaan oleh pemerintah, masyarakat, relawan dan masyarakat lain, data yang dapat dilaporkan pada fitur Jitupasna dapat berupa kondisi lingkungan sekitar, kegiatan-kegiatan positif ataupun kondisi lingkungan yang berpotensi menimbulkan bencana. Namun data yang terkumpul hanya dapat dianalisa dan digunakan oleh para pemangku kepentingan dalam mengambil keputusan. Sedangkan pada pedoman BPBD dalam melakukan Jitupasna terdapat beberapa tahapan yaitu, tahap persiapan, tahap pengumpulan data, tahap analisis dan tahap pelaporan.

Dalam menghadapi permasalahan tersebut maka perlu dilakukan pengembangan dari fitur Jitupasna dalam aplikasi inaRisk menjadi aplikasi tersendiri yang mampu mengelola data-data dari informasi pihak-pihak yang terdampak serta melakukan perhitungan secara otomatis sesuai dengan tahapan-tahapan Jitupasna yang terdapat pada pedoman BPBD. Untuk mengembangkan aplikasi tersebut maka digunakan metode DevOps, metode ini sangat adaptasi terhadap pengembangan perangkat lunak sehingga dapat dengan cepat diimplementasikan serta mendapatkan perubahan sesuai dengan kebutuhan dan dapat dilakukan *continuous improvement* pada Aplikasi Pengkajian Kebutuhan Pascabencana nantinya.

2. RUANG LINGKUP

Permasalahan difokuskan pada:

1. Metode pengembangan sistem yang digunakan dalam penelitian ini yaitu metode pengembangan *DevOps* dan alat bantu pengembangan sistem *UML* serta *git repository* untuk meninjau keseluruhan versi dan update dalam proses pembangunan dan pengembangan aplikasi.
2. Aplikasi ini dibangun dan dirancang meliputi proses input data lapangan yang meliputi data lokasi, akibat, dan dampak bencana.
3. Aplikasi yang dibangun berbasis web app yang di bagi menjadi *frontend* dan *backend* dimana *backend* akan menyediakan API untuk *frontend*.
4. Aplikasi ini akan menghasilkan pengkajian akibat, dampak dan perhitungan kebutuhan pascabencana.

3. BAHAN DAN METODE

3.1 DEVOPS

Menurut Frank Millstein (2020), istilah DevOps muncul sebagai kombinasi dari operasi dan pengembangan yang menjadi salah satu titik awal dalam menciptakan definisi, praktik, metodologi dan nilai-nilai DevOps. DevOps dapat didefinisikan sebagai metodologi, budaya dan filosofi untuk menciptakan pola pikir yang sama sekali baru dimana tim operasi dan pengembang perangkat lunak bekerja sama untuk mencapai tujuan bersama. DevOps terkadang membingungkan dengan peran atau alat dalam pengembangan perangkat lunak. Namun DevOps bukanlah sebuah alat meskipun faktanya ia menggunakan alat tertentu untuk mengotomatiskan deployment software dan delivery software dimana pengembang dan tim operasi diharapkan untuk mengetahui peran dan tanggung jawab mereka setiap saat.

Menurut Michael Joakson Gulo (2021), DevOps adalah singkatan dari Development dan Operations. Ini adalah teknologi atau metode pengembangan dengan memanfaatkan kolaborasi antara pengembang perangkat

lunak dan professional TI lainnya. Tujuan utama metode DevOps adalah untuk mengirimkan perangkat lunak lebih cepat, dan strategi praktis untuk memperoleh software yang efisien, bagian atau cara kerja metode DevOps antara lain sebagai berikut:

1. *Plan*

Tahapan ini meliputi keseluruhan perencanaan dan perancangan perangkat lunak yang akan dikembangkan. Semua aturan, persyaratan dan umpan balik dikumpulkan dan digunakan untuk membuat peta jalan proyek.

2. *Code*

Pada tahap ini pengembang mulai menulis kode dari perancangan yang sudah dibuat pada tahap *plan*, yang selanjutnya kode akan di *push* atau proses pemeriksaan kode ke dalam *repository*.

3. *Build*

Pada tahap ini dilakukan kompilasi *resource* dari kode yang sudah dibuat pada tahapan sebelumnya. Biasanya pada tahap ini sebelum *source* kode di bangun, pengembang akan mengadakan diskusi untuk mendapatkan *feedback* dan *review* kode, setelah pembahasan selesai maka *source* kode akan di-*build* secara utuh.

4. *Test*

Pada tahapan ini *source* kode yang sudah dibangun menjadi perangkat lunak diuji, apakah perangkat lunak yang dibangun atau dihasilkan telah memenuhi kriteria dan berfungsi dengan baik. Jika tidak sesuai makan pengembangan akan berhenti pada titik ini untuk meningkatkan perbaikan, namun jika sesuai maka tahapan selanjutnya adalah *release*.

5. *Release*

Pada tahap ini perangkat lunak yang sudah dibangun diberi label atau versi, kapan perangkat lunak dibangun, perubahan apa yang dilakukan, dan kapan perangkat lunak akan dirilis sebelum akhirnya perangkat lunak di-*deploy*.

6. *Deploy*

Deploy adalah proses implementasi atau penerapan perangkat lunak yang dibangun dan akhirnya dapat diakses oleh pengguna.

7. *Operate*

Selama tahap *operate*, tim operasi memastikan bahwa perangkat lunak dan infrastruktur berjalan dengan baik. Pengguna juga dapat memberikan umpan balik jika ditemukan kesalahan atau *bug*. Hal ini akan menjadi *benchmark* untuk pengembangan perangkat lunak nantinya.

8. *Monitor*

Monitor adalah fase terakhir dari siklus hidup *DevOps*. Dari fase sebelumnya dikumpulkan data kinerja, *bug* atau umpan balik. Data ini dapat digunakan untuk melakukan *self-assessment* atau evaluasi terhadap perangkat lunak yang sedang dikembangkan. Fase ini juga dapat memantau alur yang dibuat untuk hambatan di masa mendatang yang dapat memengaruhi produktivitas pengembangan perangkat lunak tersebut.

3.2 Pengkajian Kebutuhan Pascabencana

Menurut Perka No 6 Badan Nasional Penanggulangan Bencana (BNPB) Pasal 1 (2017), pengkajian kebutuhan pasca bencana yang selanjutnya disebut jitupasna adalah suatu rangkaian kegiatan dan penilaian akibat, analisis

dampak, perkiraan kebutuhan, dan rekomendasi awal terhadap strategi pemulihan yang menjadi dasar penyusunan rencana rehabilitasi dan rekonstruksi pasca bencana.

3.3 UML

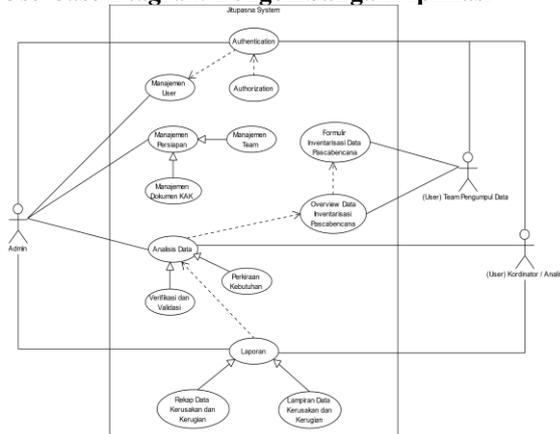
Menurut Rosa dan Shalahuddin (2018), UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. Penggunaan UML bergantung pada level abstraksi penggunaannya. Jadi belum tentu pandangan yang berbeda dalam penggunaan UML adalah suatu yang salah, tapi perlu ditelaah dimanakah UML digunakan dan hal apa yang ingin divisualkan. Secara analogi jika dengan bahasa yang kita gunakan sehari-hari, belum tentu penyampaian bahasa dengan puisi adalah hal yang salah.

3.4 Version Control System

Menurut Husni Fakhri (2021), *Version Control System* adalah sistem yang mengelola suatu perubahan pada file dokumen, *source code*, atau kumpulan informasi lainnya. VCS mencatat setiap perubahan pada file yang dikerjakan oleh seseorang. Hampir semua detail perubahan dicatat, seperti ID Perubahan, Identitas seseorang yang melakukan perubahan, waktu perubahan, judul perubahan, dan deskripsi perubahan. Biasanya hal tersebut kita temui saat kita membuat dokumen, di dalam *Microsoft Word* kita bisa menemukan hal yang serupa dengan *Version Control System*. *Version Control System* juga memungkinkan kita untuk kembali ke perubahan sebelumnya. Pada intinya, *version control system* bisa melakukan tugas seperti melihat riwayat perubahan, mengelola setiap perubahan (*commit, push, checkout, pull*), menuntaskan bagian-bagian berkas secara langsung tanpa menunggu pengerjaan yang lain selesai, dan memungkinkan kita kembali ke perubahan sebelumnya.

4. RANCANGAN APLIKASI

1. Use Case Diagram Pengembangan Aplikasi



Gambar 1. Use Case Diagram Aplikasi

Pada *Use Case Diagram* gambar 1 terdapat 3 aktor yang memiliki hak akses ke dalam aplikasi yaitu admin, team pengumpul data dan kordinator. Dimana aktor admin dapat melakukan akses terhadap manajemen user, manajemen

persiapan, analisis data dan laporan. Pada aktor team pengumpul data memiliki hak akses yaitu melakukan akses terhadap formulir inventarisasi dan *overview data* inventarisasi user team pengumpul data dibatasi oleh daftar team yang sudah diinput pada manajemen persiapan. Pada aktor kordinator memiliki hak akses yaitu melakukan akses terhadap analisis data dan laporan, dimana pada analisis data user kordinator dapat melakukan verifikasi dan validasi data yang sudah dikumpulkan oleh team pengumpul data serta dapat melihat perkiraan kebutuhan dari data tersebut, sedangkan pada laporan aktor kordinator dapat melakukan *export data* yang sudah valid.

2. Persyaratan Sistem

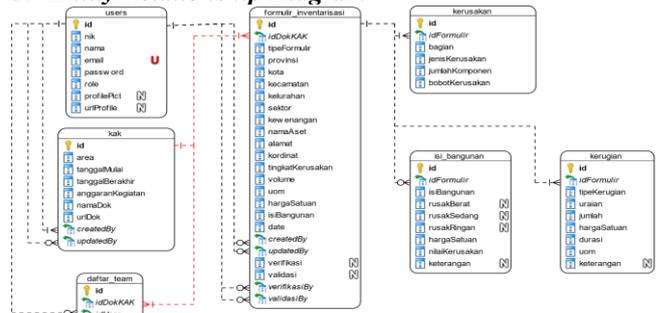
Persyaratan sistem merupakan spesifikasi dari perangkat keras maupun perangkat lunak yang digunakan untuk menjalankan sistem ketika digunakan. Persyaratan sistem dalam pengembangan aplikasi pengkajian kebutuhan pascabencana dapat dilihat pada tabel 1 berikut.

Tabel 1. Persyaratan Sistem

No	Tipe	Spesifikasi
1	Server	Cloud atau Hosting
2	Memori	RAM 3GB, Disk 12GB
3	DBMS	NoSql / Mongoddb
4	Web Server	Node JS
5	Framework	Express JS, Next JS

Aplikasi jitungasna nantinya akan di *hosting* dengan menggunakan spesifikasi server yaitu ram sebesar 3GB dan penyimpanan sebesar 12GB, kapasitas ram sebesar 3GB diharapkan cukup untuk menjalankan *web server* dan penyimpanan sebesar 12GB diharapkan cukup untuk menyimpan file yang di upload kedalam aplikasi. Untuk DBMS menggunakan *database cloud* NoSql Mongoddb yang terpisah dengan server *hosting* yang diharapkan tidak menambah beban pada ram dan penyimpanan pada server *hosting* nantinya. Sedangkan untuk *framework* akan menggunakan Express JS untuk *backend* dan Next Js untuk *frontend*, kedua *framework* tersebut nantinya akan dijalankan pada *web server* Node JS pada server *hosting*.

3. Entity Relationship Diagram

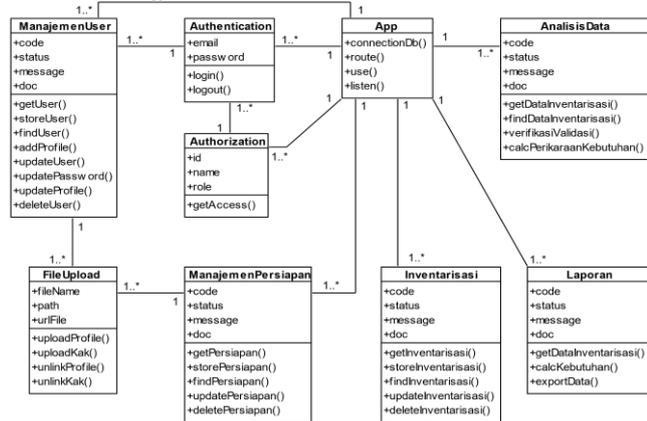


Gambar 2. Entity Relationship Diagram

Dapat dilihat pada gambar 4.7, terdapat 7 tabel yang saling berelasi. Tabel *users* memiliki relasi *one to many* dengan tabel *kak*, tabel *daftar_team* dan tabel *formulir_inventarisasi*, dimana id user akan digunakan pada masing-masing tabel relasinya sebagai *foreign key* untuk mereferensi data pada tabel *users*. Tabel *kak* memiliki relasi

one to many terhadap tabel daftar_team dan tabel formulir_inventarisasi, dimana id kak akan digunakan pada masing-masing tabel relasinya sebagai *foreign key* untuk mereferensi data pada tabel kak. Tabel formulir_inventarisasi memiliki relasi *one to many* dengan tabel kerusakan, isi_bangunan dan kerugian dimana table formulir_inventarisasi digunakan sebagai induk data, id formulir_inventarisasi akan digunakan pada masing-masing tabel relasinya sebagai *foreign key* untuk mereferensi data pada tabel formulir_inventarisasi. Pada kolom id akan menggunakan uuid untuk menghindari *duplicated* data atau data ganda. Kolom dengan tanda U mengindikasikan bahwa kolom tersebut bersifat *unique* yang artinya kolom tersebut hanya boleh diisi dengan *value* unik atau tidak boleh sama, sedangkan kolom dengan tanda N mengindikasikan bahwa kolom tersebut bersifat *null* yang artinya kolom tersebut boleh diisi atau dikosongkan.

4. Class Diagram



Gambar 3. Class Diagram

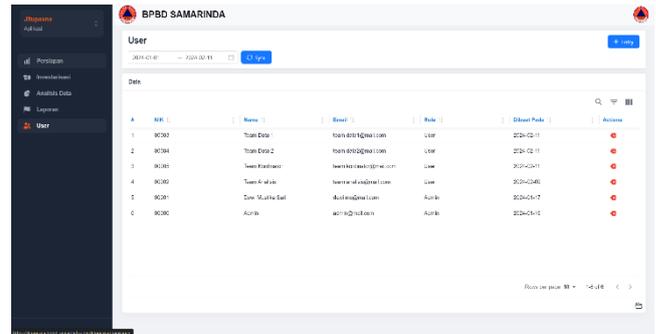
Pada *Class Diagram* yang dapat dilihat pada gambar 4.8 terdapat 9 *Class* dimana *Class App* adalah main *Class* yang akan diakses pertama kali, adapun *Class App* bertugas untuk menangani koneksi ke *database*, mengatur *routing* aplikasi, melakukan akses dan menggunakan *library* yang terpasang, serta mengaktifkan server pada port yang sudah ditentukan. *Class App* memiliki relasi *one to many* terhadap 7 *Class* lainnya dimana *operation route* akan menentukan *Class* yang akan diakses oleh *Class App* berdasarkan *request* dari *front end*. Sedangkan *Class FileUpload* bertugas untuk melakukan upload file dimana *Class* ini digunakan atau berelasi dengan *Class ManajemenUser* dan *ManajemenPersiapan*

5. IMPEMENTASI

Berikut adalah tampilan aplikasi jitupasna yang berhasil dikembangkan dengan menggunakan metode DevOps:

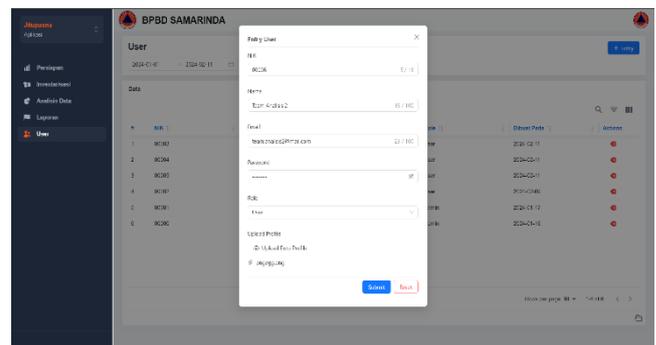
1. Manajemen User

Fitur ini memungkinkan admin untuk membuat atau menambahkan akun user yang dapat melakukan akses atau login kedalam aplikasi pengkajian kebutuhan pascabencana. Selain menambahkan akun user pada fitur ini admin juga dapat merubah foto profil dari user, merubah data nik, nama, email, dan role dari user yang bersangkutan, pada fitur ini juga admin dapat melakukan perubahan password pada user yang diinginkan dan juga menghapus data user.



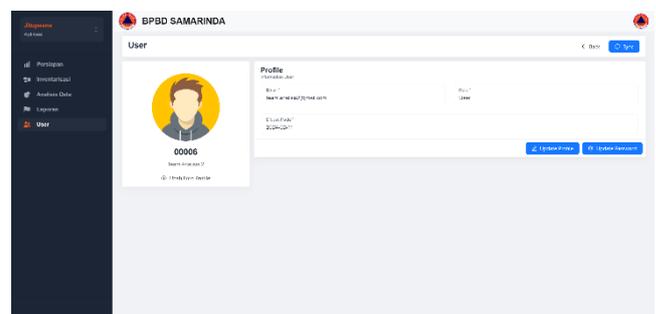
Gambar 4. Manajemen User

Untuk menambahkan user baru, admin dapat menekan tombol entry dan mengisikan data-data seperti nik, nama, email, password, role dan foto profil untuk user yang akan dibuat. Jika data telah sesuai dan benar admin dapat langsung menekan tombol submit untuk menyimpan data user tersebut. Berikut adalah gambar form entry user yang dapat dilihat pada gambar 5 berikut.



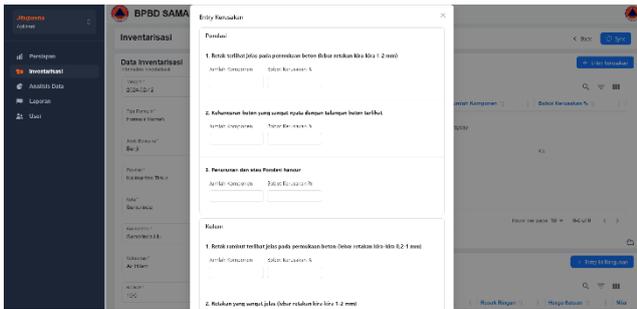
Gambar 5. Entry User

Untuk melihat detail data dari user yang sudah dibuat, admin dapat melakukan klik 2 kali pada data user yang diinginkan, kemudian sistem akan menampilkan halaman detail dari user tersebut, sedangkan untuk menghapus user yang sudah dibuat admin dapat menekan tombol action yaitu delete dan melakukan konfirmasi penghapusan data, untuk halaman detail user dapat dilihat pada gambar 6 berikut.



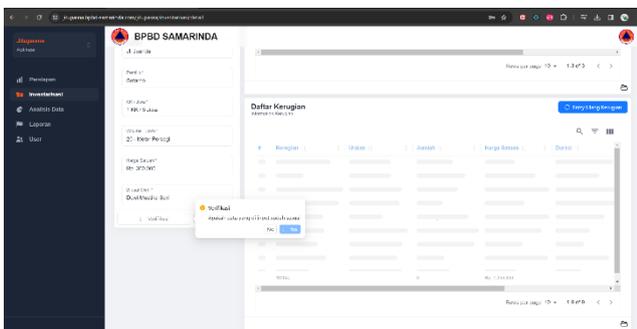
Gambar 6. Detail User

Halaman detail user menampilkan foto profil dari user, nik, dan nama serta menampilkan data-data lain seperti email, role dan tanggal kapan akun tersebut dibuat. Pada halaman detail user admin dapat merubah foto profil user dengan menekan tombol ubah foto profil. Sedangkan untuk merubah atau melakukan update profil, admin dapat menekan tombol update profile, sedangkan untuk merubah password admin dapat menekan tombol update password dan mengisikan password baru.



Gambar 12. Entry Kerusakan

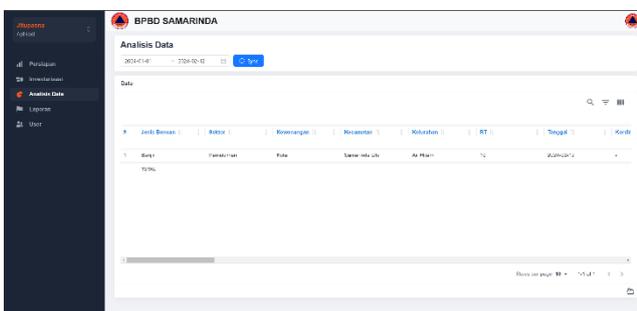
Semua form pada entry data kerusakan wajib diisi, apabila kenyataan pada lapangan kerusakan tersebut tidak ada maka user dapat mengisi data tersebut dengan nilai 0. Hal ini juga berlaku bagi data isi bangunan dan data kerugian, setelah data kerusakan, isi bangunan dan data kerugian berhasil dibuat, tombol entry akan berubah menjadi tombol entry ulang, hal ini bertujuan agar user dapat melakukan entry ulang apabila data yang diinput sebelumnya terdapat kesalahan. Apabila semua data telah sesuai user dapat melakukan verifikasi dengan menekan tombol verifikasi, dan data yang sudah diverifikasi tidak dapat dirubah kembali. Berikut adalah tampilan verifikasi data yang dapat dilihat pada gambar 13.



Gambar 13. Verifikasi Data Inventarisasi

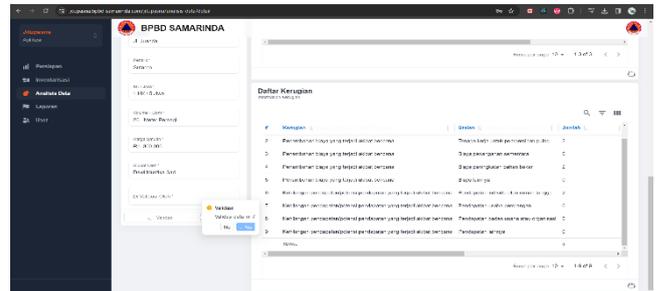
4. Analisis Data

Fitur ini ditujukan bagi user analisis data untuk melakukan analisis data dan melihat detail data yang sudah di buat oleh team pengumpul data. Pada halaman analisis perkiraan nilai dihitung secara otomatis oleh sistem dan team analisis dapat melakukan validasi dengan menekan tombol action yaitu validasi dan melakukan konfirmasi validasi. User analisis data juga dapat melihat detail data sebelum melakukan validasi dengan melakukan klik 2 kali pada data yang diinginkan dan dapat melakukan validasi pada halaman detail data. Berikut adalah tampilan halaman analisis data yang dapat dilihat pada gambar 14.



Gambar 14. Analisis Data

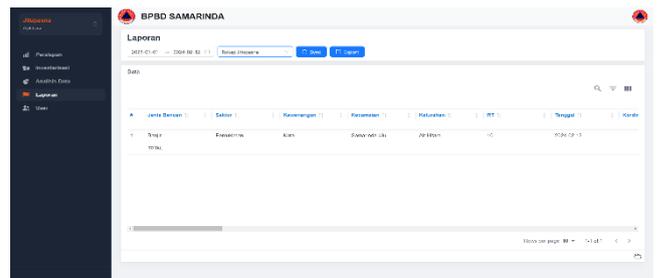
Pada halaman detail data, user analisis data dapat melihat detail data kerusakan, isi bangunan dan data kerugian, serta dapat melihat foto dari tempat kejadian. Berikut adalah tampilan proses validasi data yang dapat dilihat pada gambar 15.



Gambar 15. Proses Validasi Data

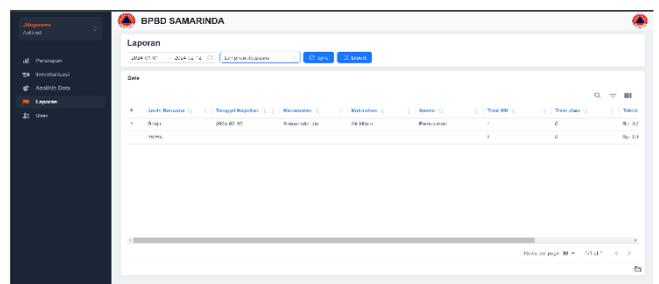
5. Laporan

Fitur ini digunakan untuk melihat data-data yang sudah dibuat berdasarkan tipe data yang sudah di tentukan yaitu laporan rekap jitupasna dan lampiran jitupasana. Laporan rekap jitupasna berisikan data-data inventarisasi yang sudah dihitung perkiraan nilainya dan data tersebut merupakan data real yang dibuat oleh team pengumpul data dan sudah divalidasi oleh team analisis. Sedangkan laporan lampiran jitupasana merukan pengelompokan dari rekap jitupasna berdasarkan tanggal kejadian bencana. Kedua laporan dapat diunduh kedalam data format excel yang nantinya dapat diolah kembali menjadi data-data yang dibutuhkan.



Gambar 16. Laporan Rekap Jitupasna

User dapat memilih rentang tanggal laporan yang ingin diunduh, dan dapat memilih tipe laporan dengan menekan tombol pilihan laporan, tombol sync digunakan untuk melakukan refresh data terbaru sebelum data tersebut diunduh. Untuk melakukan export atau mengunduh data excel dari laporan yang diinginkan user dapat menekan tombol export dan sistem akan otomatis melakukan pengunduhan data dengan format .xlsx, berikut adalah tampilan laporan lampiran jitupasna yang dapat dilihat pada gambar 17.



Gambar 17. Laporan Lampiran Jitupasna

6. KESIMPULAN

1. Metodologi DevOps tidak hanya mencakup fase pengembangan tetapi juga fase operational, dimana tahapan pada metode DevOps cukup komprehensif sehingga sangat kecil kemungkinan aplikasi gagal untuk diimplementasikan secara berkelanjutan.
2. Permasalahan development yang meliputi tahap plan, code, build, dan test juga dapat didukung oleh tools sehingga meminimalkan terjadinya kesalahan dalam setiap tahapan development.
3. Masalah operational seperti implementasi aplikasi pengkajian kebutuhan pascabencana dapat diautomatiskan dengan mudah dengan menggunakan tools seperti repository.
4. Berdasarkan hasil penelitian, pada aplikasi inRisk dengan fitur jituasna aplikasi ini merupakan aplikasi untuk melaporkan kegiatan atau informasi terkait pengurangan resiko bencana dimana data yang terkumpul dikelola oleh BNPB.
5. Dari hasil penelitian, BPBD masih melakukan pendataan secara manual sehingga dibutuhkan pengembangan dari fitur jituasna pada aplikasi inRisk mobile dimana data yang dihasilkan dapat dikelola langsung oleh BPBD.

7. SARAN

1. Dapat dilakukan pengembangan atau penambahan fitur yang dapat menjadi fokus pada area *Continuous Development* sebagai salah satu siklus penting dalam metode DevOps.
2. Hasil penelitian dapat menjadi bahan pertimbangan bagi instansi BPBD dalam menerapkan teknologi informasi didalam lingkup kerjanya.
3. Aplikasi yang dibangun dapat digunakan dengan baik sebagai solusi atas masalah yang dihadapi yaitu pengelolaan data inventarisasi pascabencana.
4. Dapat dilakukan pengembangan untuk aplikasi mobile agar team pengumpul data mampu melakukan pendataan langsung melalui mobile sehingga data dapat terdistribusi dengan cepat.
5. Diharapkan dapat dijadikan sebagai bahan acuan atau referensi bagi mahasiswa bidang informatika.

8. DAFTAR PUSTAKA

Buku:

Badan Nasional Penanggulangan Bencana. 2017. *Peraturan Badan Nasional Penanggulangan Bencana Republik Indonesia Nomor 6 Tahun 2017 Tentang Penyelenggaraan Rehabilitasi dan Rekonstruksi Pascabencana*. Jakarta: Badan Nasional Penanggulangan Bencana.

Buku:

Millstein, Frank. 2020. *DevOps Handbook And DevOps Adoption*. Google Books: Frank Millstein.

Buku:

S, Rosa, A. dan M. Shalahuddin. 2018. *Rekayasa Perangkat Lunak Terstruktur Dan Berorientasi Objek Informatika Edisi Revisi*. Bandung: Informatika.

Jurnal Ilmiah:

Gulo, Michael Joakson, 2021. *Analisis Penerapan Metode DevOps Pada Aplikasi Berbasis Android*. Batam: Universitas Putera Batam.

Jurnal Ilmiah:

Faqih, Husni. Pangestu Andika T. 2021. *Modul Dasar GIT dan Github*. Tegal: Universitas BSI.