

Pengembangan Aplikasi Web *TrustNet* untuk Deteksi Gangguan Informasi Menggunakan Metode *Scrum*

Muhamad Rosidin^{*1}, Maela Khoirul Ummah², Panggah Widiandana³, Gilar Adi Pratama⁴, Wicaksono Yuli Sulisty⁵

^{1,4,5}Program Studi Sistem Informasi, Universitas Siber Muhammadiyah, Yogyakarta, Indonesia

²Program Studi Hukum, Universitas Siber Muhammadiyah, Yogyakarta, Indonesia

³Program Studi Informatika, Universitas Islam Mulia, Yogyakarta, Indonesia

E-mail: ^{*1}rosidin@sibermu.ac.id, ²maela.ummah@sibermu.ac.id ³panggah.widiandana@uim-yogya.ac.id, ⁴gilar20230200040@sibermu.ac.id, ⁵wicaksono@sibermu.ac.id

Abstrak

Gangguan informasi berupa misinformasi, disinformasi, dan malinformasi menyebar jauh lebih cepat daripada kemampuan verifikasi manual, sehingga dibutuhkan aplikasi yang mampu menelusuri bukti dan menyajikan tingkat kepercayaan suatu klaim secara terbuka. Sebagian besar penelitian sistem deteksi di Indonesia berhenti pada perbandingan algoritma dan, ketika sampai pada tahap pembangunan aplikasi, mengandalkan model Waterfall yang mengasumsikan kebutuhan bersifat tetap. Asumsi tersebut sulit dipenuhi pada sistem berbasis pembelajaran mesin, karena kebutuhan baru justru muncul setelah model dievaluasi. Penelitian ini mengembangkan aplikasi web *TrustNet* untuk deteksi gangguan informasi menggunakan metode *Scrum*, dengan pekerjaan disusun sebagai *product backlog* dan dikerjakan melalui enam *sprint*. Setiap *sprint* menghasilkan *increment* yang dapat dijalankan, mulai dari akuisisi berita, penelusuran pengetahuan, validasi bukti, kalibrasi probabilitas, evolusi keyakinan, hingga dasbor intelijen dan otomatis terjadwal. Pengujian fungsional dilakukan menggunakan *black box testing* terhadap 30 skenario uji yang mencakup sepuluh modul aplikasi. Hasil pengujian menunjukkan 28 skenario berjalan sesuai dengan yang diharapkan, sehingga diperoleh tingkat kelulusan sebesar 93,3%. Aplikasi berhasil dioperasikan pada aliran berita nyata dengan 2.004 artikel terhimpun, 3.079 isu terekstraksi, dan 333 isu tervalidasi bukti. Penerapan *Scrum* terbukti mengakomodasi dua perubahan kebutuhan yang tidak teridentifikasi pada perencanaan awal, yaitu penyisipan lapisan validasi bukti dan pengulangan pelatihan model hingga versi ketiga belas. Temuan ini menunjukkan bahwa pendekatan iteratif lebih sesuai untuk pengembangan aplikasi deteksi berbasis pembelajaran mesin dibanding pendekatan sekuensial.

Kata kunci - *Blackbox Testing*, Gangguan Informasi, Pengembangan Perangkat Lunak, *Scrum*, *TrustNet*

1. PENDAHULUAN

Penyebaran informasi melalui kanal digital berlangsung jauh lebih cepat dibanding proses verifikasinya. Berita yang belum terkonfirmasi dapat menjangkau ribuan pembaca dalam hitungan menit dan menimbulkan kerugian nyata, mulai dari menurunnya kepercayaan terhadap program kesehatan publik [1] hingga terganggunya proses politik menjelang pemilihan umum [2]. Kemunculan kecerdasan buatan generatif memperberat persoalan tersebut karena konten palsu kini dapat diproduksi dalam jumlah besar dengan kualitas bahasa yang menyerupai tulisan manusia [3]. Kondisi ini menuntut tersedianya aplikasi yang tidak sekadar memberi label benar atau salah, melainkan juga menampilkan bukti pendukung dan tingkat kepercayaan yang dapat ditelusuri pengguna.

Penelitian mengenai deteksi berita palsu di Indonesia telah berkembang cukup luas pada aras algoritma. Algoritma Naive Bayes banyak digunakan sebagai basis klasifikasi teks dan menunjukkan akurasi yang tinggi pada korpus terbatas [4], sementara perbandingan dengan Support Vector Machine memberikan gambaran mengenai karakteristik masing-masing pendekatan [5]. Model berbasis transformer seperti IndoBERT dilaporkan lebih tangguh dibanding pendekatan statistik ketika dihadapkan pada kalimat jebakan [6], dan representasi kata terdistribusi juga dimanfaatkan untuk klasifikasi teks berbahasa Indonesia [7]. Pendekatan penggalian teks maupun pembelajaran mendalam turut diterapkan pada domain berita kesehatan dan perbandingan antarsitus berita [8], [9].

Sebagian penelitian melangkah lebih jauh dengan mewujudkan hasil klasifikasi menjadi aplikasi yang dapat digunakan publik. Sistem deteksi hoaks berbasis web telah dibangun dengan memadukan pemrosesan bahasa alami dan klasifikasi Naive Bayes [10], sedangkan pengembangan sistem deteksi informasi palsu berbasis BERT dilaporkan mencapai tingkat kelulusan pengujian black box sebesar 95,5% [11]. Pemanfaatan model bahasa besar yang dipadukan dengan Retrieval-Augmented Generation juga mulai diterapkan untuk menghadirkan jawaban yang bersandar pada dokumen sumber [12]. Perkembangan ini menunjukkan bahwa persoalan deteksi gangguan informasi telah bergeser dari pertanyaan algoritma apa yang paling akurat menuju pertanyaan bagaimana kemampuan tersebut diwujudkan menjadi perangkat lunak yang layak dioperasikan.

Pada aras rekayasa perangkat lunak, penelusuran terhadap publikasi pengembangan sistem deteksi di Indonesia memperlihatkan dominasi pendekatan sekuensial. Model Waterfall digunakan pada pembangunan sistem informasi berbasis web [13], sistem pemantauan kesehatan berbasis Android [14], maupun sistem pendukung deteksi objek [15]. Pendekatan berorientasi objek melalui Object-Oriented Analysis and Design dipilih untuk menjamin arsitektur yang modular [16], sedangkan metode prototyping diterapkan ketika kebutuhan pengguna belum sepenuhnya jelas di awal [17], [18]. Pengembangan sistem pakar untuk deteksi dini kondisi kesehatan pun disusun mengikuti alur bertahap yang serupa dan dievaluasi setelah seluruh tahap perancangan selesai [19]. Seluruh pendekatan tersebut menempatkan penetapan kebutuhan sebagai aktivitas yang tuntas sebelum implementasi dimulai.

Asumsi kebutuhan yang bersifat tetap sulit dipertahankan pada aplikasi yang intinya adalah model pembelajaran mesin. Kebutuhan baru justru banyak bermunculan setelah model dilatih dan dievaluasi, misalnya ketika keluaran model ternyata memerlukan penyaring bukti tambahan atau ketika distribusi kelas menuntut pelatihan ulang. Kesenjangan inilah yang menjadi titik tolak penelitian ini: belum ditemukan publikasi pengembangan aplikasi deteksi gangguan informasi di Indonesia yang secara eksplisit menerapkan kerangka kerja Scrum beserta artefak dan seremoninya. Penelitian ini bertujuan mengembangkan aplikasi web TrustNet untuk deteksi gangguan informasi menggunakan metode Scrum, mendokumentasikan bagaimana perubahan kebutuhan ditangani melalui mekanisme sprint, serta mengukur kesesuaian fungsional aplikasi melalui pengujian black box. Kontribusi yang diharapkan mencakup tersedianya rujukan penerapan Scrum pada pengembangan aplikasi deteksi berbasis pembelajaran mesin, sekaligus aplikasi TrustNet yang dapat diakses publik.

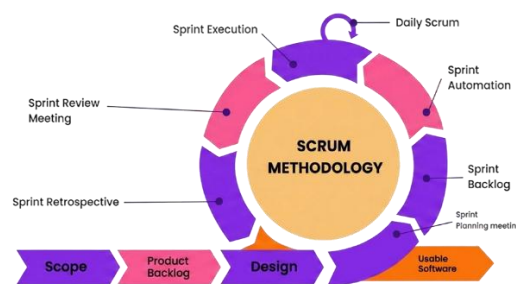
2. METODE PENELITIAN

Penelitian ini merupakan penelitian rekayasa perangkat lunak yang menghasilkan aplikasi web sebagai luaran utama. Objek penelitian adalah aplikasi TrustNet, sebuah platform media intelligence untuk deteksi gangguan informasi pada berita daring yang dapat diakses pada <https://trustnet.sibermu.my.id/>. TrustNet merupakan nama riset dan nama domain teknis platform ini, sedangkan pada antarmuka publik yang diakses pengguna sehari-hari aplikasi yang sama ditampilkan dengan nama merek KemenWaras (Kementerian Orang Waras); kedua nama tersebut merujuk pada satu sistem yang identik, sehingga seluruh tangkapan layar pada bagian hasil penelitian yang menampilkan branding KemenWaras adalah tampilan aktual aplikasi TrustNet.

Pengembangan dilakukan menggunakan kerangka kerja Scrum, sedangkan kesesuaian fungsional aplikasi diukur menggunakan black box testing.

2.1 Kerangka Kerja Scrum

Metode Scrum dipilih karena spesifikasi kebutuhan aplikasi tidak dapat ditetapkan secara menyeluruh pada tahap awal pengembangan. Komponen inti aplikasi, yakni model klasifikasi dan penelusuran bukti, baru dapat dinilai kelayakannya setelah diuji menggunakan data nyata. Oleh karena itu, hasil evaluasi dari satu tahap akan menjadi masukan bagi penentuan kebutuhan di tahap selanjutnya. Karakteristik ini berbeda dengan pendekatan sekuensial yang umum digunakan pada sebagian besar penelitian sejenis, namun sejalan dengan prinsip evaluasi iteratif pada metode prototyping. Secara visual, alur kerja dan tahapan pelaksanaan pengembangan perangkat lunak menggunakan metodologi *Scrum* dapat dilihat secara lebih terperinci pada Gambar 1.



Gambar 1 Metodologi Scrum

Gambar 1 merupakan alur metodologi *Scrum* dimulai dengan pendefinisian ruang lingkup proyek (*Scope*) yang kemudian dijabarkan ke dalam daftar kebutuhan utama sistem atau *Product Backlog*. Setelah melalui tahap perancangan awal (*Design*), proses pengembangan memasuki siklus iteratif utama. Setiap iterasi diawali dengan *Sprint Planning meeting* guna menentukan *Sprint Backlog*, yaitu kumpulan tugas yang akan diselesaikan dalam satu periode *sprint*. Selama tahap pengerjaan (*Sprint Execution* dan *Sprint Automation*), tim pengembang melakukan koordinasi harian melalui seremoni *Daily Scrum*. Pada akhir setiap siklus, tim menyelenggarakan *Sprint Review Meeting* untuk mendemonstrasikan hasil kerja kepada pemangku kepentingan, serta *Sprint Retrospective* untuk mengevaluasi dan memperbaiki proses kerja internal tim. Rangkaian siklus ini akan terus berulang dan secara bertahap menghasilkan penambahan fitur aplikasi berupa perangkat lunak yang berfungsi dan siap digunakan (*Usable Software*).

Peran di dalam tim pengembang terbagi menjadi tiga. Pertama, Product Owner bertanggung jawab dalam menyusun dan menetapkan prioritas product backlog berdasarkan kebutuhan analisis informasi. Kedua, Scrum Master bertugas memfasilitasi jalannya seremoni (Scrum events) serta mengatasi berbagai hambatan teknis. Ketiga, Development Team melaksanakan proses perancangan, pengodean, dan pengujian secara mandiri. Adapun seremoni yang dijalankan dalam siklus ini meliputi: Sprint Planning pada awal sprint, Daily Scrum sebagai wadah sinkronisasi harian, Sprint Review untuk mendemonstrasikan increment (hasil kerja), serta Sprint Retrospective yang bertujuan untuk mengevaluasi dan memperbaiki proses kerja pada sprint berikutnya.

2.2 Product Backlog dan Perencanaan Sprint

Kebutuhan aplikasi dirumuskan sebagai product backlog dalam bentuk user story yang diberi bobot menggunakan story point berskala Fibonacci. Prioritas ditetapkan berdasarkan ketergantungan antarmodul, yaitu modul yang menjadi masukan bagi modul lain dikerjakan lebih dahulu. Ringkasan product backlog beserta pemetaannya ke dalam sprint ditunjukkan pada Tabel 1.

Tabel 1 Product backlog dan pemetaan sprint aplikasi TrustNet

ID	User story	Story point	Sprint
PB-01	Sebagai sistem, saya menghimpun berita dari kanal media daring secara terjadwal	8	1
PB-02	Sebagai analis, saya melihat isu yang sedang menjadi tren dari berita terhimpun	5	1

ID	User story	Story point	Sprint
PB-03	Sebagai sistem, saya menelusuri dokumen pengetahuan yang relevan dengan suatu isu	13	2
PB-04	Sebagai analis, saya melihat bukti pendukung beserta skor kecocokan fakta	8	3
PB-05	Sebagai sistem, saya menghasilkan probabilitas terkalibrasi atas status suatu isu	13	4
PB-06	Sebagai analis, saya memantau perubahan tingkat keyakinan ketika bukti baru muncul	8	5
PB-07	Sebagai analis, saya membaca ringkasan asesmen intelijen per isu	5	5
PB-08	Sebagai analis, saya melihat dasbor ringkas kondisi nasional dan antrian kerja	8	6
PB-09	Sebagai administrator, saya menjadwalkan eksekusi rantai proses secara otomatis	5	6
PB-10	Sebagai administrator, saya mengelola akun dan peran pengguna aplikasi	3	6

Tabel 1 mencerminkan estimasi kerumitan relatif, bukan durasi pengerjaan. Butir PB-03 dan PB-05 memperoleh bobot tertinggi karena keduanya melibatkan integrasi model bahasa dan proses kalibrasi yang hasilnya tidak dapat dipastikan sejak awal.

2.3 Arsitektur Aplikasi

Aplikasi disusun sebagai rantai pemrosesan berlapis. Lapisan pertama melakukan akuisisi berita dan ekstraksi isu. Lapisan kedua menelusuri dokumen pengetahuan menggunakan pendekatan Retrieval-Augmented Generation. Lapisan ketiga menghasilkan probabilitas terkalibrasi atas status suatu isu melalui tiga tahap: pertama, sistem menghitung skor deteksi awal pola berdasarkan kemiripan isu dengan korpus berita yang telah diberi label; kedua, skor tersebut disesuaikan dengan konteks menggunakan bukti yang diperoleh dari lapisan penelusuran pengetahuan; dan ketiga, probabilitas akhir dikalibrasi menggunakan pendekatan inferensi Bayesian sehingga nilai keluaran mencerminkan kekuatan bukti yang ditemukan, bukan sekadar skor klasifikasi mentah. Lapisan keempat memperbarui tingkat keyakinan ketika bukti baru tersedia. Seluruh lapisan dikendalikan oleh penjadwal yang menjalankan rantai proses secara otomatis, dengan pola otomatisasi yang serupa dengan sistem pemantauan berkala pada domain lain [20].

Antarmuka aplikasi dibangun berbasis web dengan penyusunan tata letak yang mengacu pada prinsip user centered design agar informasi kepercayaan suatu isu dapat dibaca analis tanpa pelatihan khusus [21]. Hak akses dipisahkan ke dalam peran administrator dan author melalui mekanisme autentikasi dan middleware, sebagaimana diterapkan pada pengamanan sistem persuratan berbasis web [22], sedangkan pembatasan akses pada tiap modul mengikuti praktik access control pada pengamanan informasi website akademik [23]. Kredensial layanan eksternal disimpan sebagai variabel lingkungan dan tidak ditanam langsung pada kode sumber, mengikuti praktik pencegahan kebocoran kredensial pada repositori [24].

2.4 Metode Pengujian

Pengujian kesesuaian fungsional dilakukan menggunakan black box testing, yaitu pengujian yang menilai keluaran aplikasi terhadap masukan tertentu tanpa memeriksa struktur kode internalnya. Metode ini dipilih karena berfokus pada perilaku yang teramati oleh pengguna dan telah digunakan pada penelitian pengembangan sistem deteksi maupun evaluasi sistem informasi berbasis web sebagaimana diuraikan pada bagian pendahuluan. Konsep dasar dan alur kerja dari metode pengujian tersebut diilustrasikan pada Gambar 2.



Gambar 2 Alur black box testing

Gambar 2 metode *Black Box Testing* berfokus pada pengujian fungsionalitas sistem tanpa perlu mengetahui struktur internal maupun kode program dari aplikasi tersebut (direpresentasikan sebagai kotak hitam atau *Software under test*). Proses pengujian berawal dari dokumen spesifikasi kebutuhan (*Requirements Document*) yang digunakan sebagai acuan utama untuk merumuskan data masukan (*Inputs*). Data masukan tersebut kemudian diproses oleh sistem untuk

menghasilkan sebuah luaran (*Outputs*). Pada tahap akhir, luaran yang dihasilkan akan dievaluasi dan divalidasi (*Validate output*) guna memastikan bahwa respons sistem telah sepenuhnya sesuai dengan ekspektasi fungsional yang tertuang dalam dokumen spesifikasi awal.

Sebanyak 30 skenario uji disusun untuk mencakup sepuluh modul aplikasi sesuai butir product backlog. Setiap skenario memuat masukan, hasil yang diharapkan, dan hasil yang teramati. Skenario dinyatakan lulus apabila hasil yang teramati sama dengan hasil yang diharapkan. Tingkat kelulusan dihitung menggunakan persamaan (1).

$$\text{Tingkat kelulusan} = (\text{Jumlah skenario lulus} / \text{Jumlah seluruh skenario}) \times 100\% \quad (1)$$

Pengujian dijalankan pada lingkungan produksi menggunakan data berita nyata yang telah terhimpun, sehingga perilaku yang teramati mencerminkan kondisi operasional aplikasi dan bukan kondisi laboratorium.

3. HASIL DAN PEMBAHASAN

3.1 Pelaksanaan Sprint

Pengembangan diselesaikan dalam enam sprint. Setiap sprint ditutup dengan sprint review yang menghasilkan increment berupa modul yang dapat dijalankan, kemudian dilanjutkan sprint retrospective untuk menilai proses kerja. Ringkasan capaian tiap sprint ditunjukkan pada Tabel 2.

Tabel 2 Capaian increment dan hasil sprint review

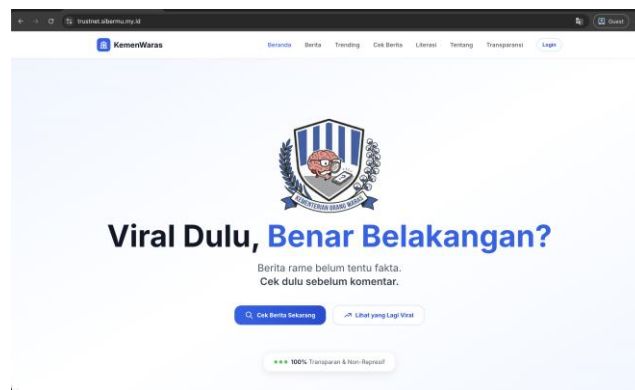
Sprint	Butir backlog	Increment yang dihasilkan	Hasil review
1	PB-01, PB-02	Modul akuisisi berita terjadwal dan halaman tren isu	Diterima
2	PB-03	Modul penelusuran pengetahuan berbasis RAG	Diterima dengan catatan
3	PB-04	Modul validasi bukti beserta skor kepercayaan dan kecocokan fakta	Diterima
4	PB-05	Modul kalibrasi probabilitas dan panel kesehatan model	Diterima dengan catatan
5	PB-06, PB-07	Modul evolusi keyakinan dan asesmen intelijen	Diterima
6	PB-08, PB-09, PB-10	Dasbor operasional, penjadwal proses, dan manajemen peran pengguna	Diterima

Dua sprint ditutup dengan status diterima dengan catatan. Pada sprint kedua, review menemukan bahwa keluaran penelusuran pengetahuan belum dapat langsung dipercaya karena dokumen yang terambil tidak selalu memuat pernyataan yang benar-benar menguji klaim. Product owner kemudian menambahkan butir PB-04 sebagai lapisan validasi bukti yang disisipkan di antara penelusuran pengetahuan dan kalibrasi probabilitas. Butir ini tidak terdapat pada product backlog awal dan merupakan hasil langsung dari mekanisme review.

Pada sprint keempat, review menemukan bahwa probabilitas keluaran model belum stabil sehingga pelatihan ulang perlu dilakukan berkali-kali seiring bertambahnya data berlabel. Pelatihan ulang tersebut dikelola sebagai pekerjaan berulang lintas sprint dan menghasilkan model aktif versi ketiga belas pada akhir pengembangan. Kedua temuan ini memperlihatkan bahwa kebutuhan penting baru terungkap setelah increment dijalankan pada data nyata, suatu kondisi yang sulit diakomodasi apabila kebutuhan dibekukan sejak tahap analisis.

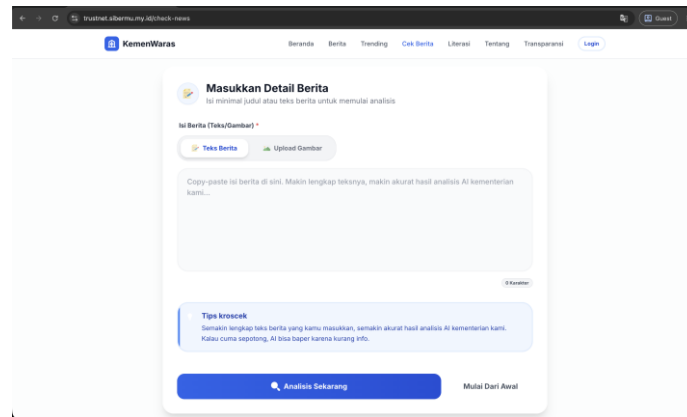
3.2 Implementasi Antarmuka Aplikasi

Increment dari keenam sprint terintegrasi menjadi satu aplikasi web yang dapat diakses publik bernama TrustNet, yang pada antarmuka publiknya ditampilkan dengan nama merek KemenWaras sebagaimana terlihat pada Gambar 3 hingga Gambar 7. Antarmuka aplikasi disajikan secara berurutan mulai dari halaman beranda, form pemasukan berita yang akan diverifikasi, hingga halaman keluaran hasil analisis, untuk memperlihatkan alur penggunaan sistem oleh pengguna. Tampilan halaman beranda aplikasi ditunjukkan pada Gambar 3.



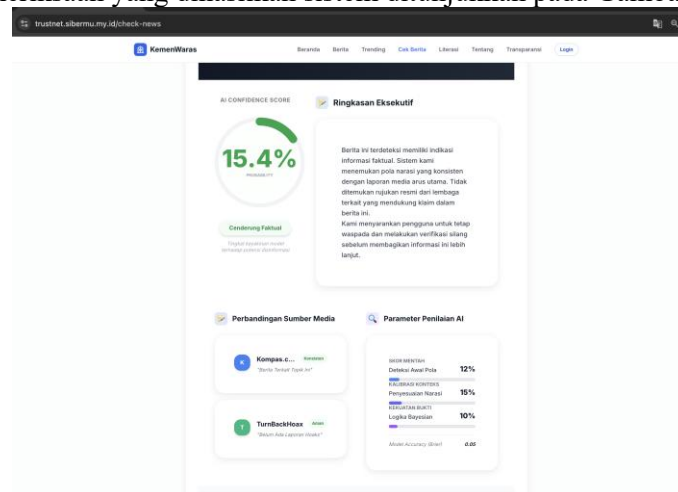
Gambar 3 Halaman Beranda Aplikasi KemerWaras

Gambar 3 merupakan halaman beranda menampilkan slogan ajakan verifikasi sebelum menyebarkan informasi beserta dua tombol akses utama, yaitu untuk memeriksa berita secara langsung dan untuk melihat daftar isu yang sedang viral. Pengguna yang memilih memeriksa berita kemudian diarahkan ke form pemasukan konten yang akan diverifikasi, sebagaimana ditunjukkan pada Gambar 4.



Gambar 4 Antarmuka Input Verifikasi Berita

Gambar 4 pengguna dapat menempelkan teks berita atau mengunggah gambar berita yang ingin diperiksa. Sistem menampilkan jumlah karakter yang dimasukkan beserta tips kroscek untuk membantu pengguna menyusun masukan yang lebih lengkap sebelum menekan tombol analisis. Hasil pemeriksaan yang dihasilkan sistem ditunjukkan pada Gambar 5.



Gambar 5 Output Analisis AI Confidence Score dan Parameter Penilaian

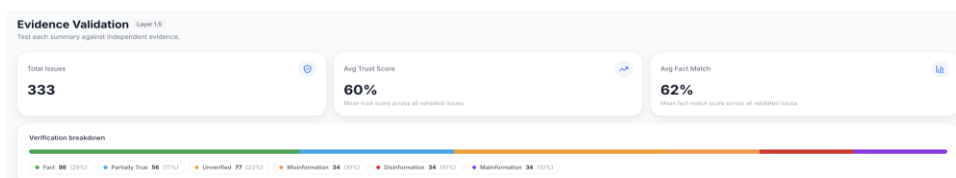
Gambar 5 memperlihatkan skor AI confidence score beserta rincian parameter penilaian, yaitu deteksi awal pola, kalibrasi konteks, dan kekuatan bukti berbasis logika Bayesian, disertai

ringkasan eksekutif yang menjelaskan dasar penilaian secara naratif. Selain hasil pemeriksaan per berita, aplikasi juga menyediakan halaman dasbor operasional yang merangkum kondisi seluruh isu yang sedang ditangani, sebagaimana ditunjukkan pada Gambar 6.



Gambar 6 Halaman dasbor operasional aplikasi TrustNet

Gambar 6 panel sebaran risiko dan prioritas menampilkan proporsi isu berdasarkan tingkat urgensi penanganan, sedangkan antrean di sisi lain memuat isu yang tingkat keyakinannya berubah signifikan dan menunggu peninjauan analis. Increment sprint ketiga berupa modul validasi bukti ditampilkan pada halaman tersendiri yang memuat jumlah isu tervalidasi, rerata skor kepercayaan, rerata skor kecocokan fakta, dan sebaran status verifikasi. Tampilan modul tersebut ditunjukkan pada Gambar 7.



Gambar 7 Halaman validasi bukti sebagai increment sprint ketiga

Gambar 7 mencerminkan kondisi operasional aplikasi hingga saat pengujian dilakukan. Selama periode pengamatan operasional berkelanjutan sekitar empat hari, aplikasi telah menghimpun 2.004 artikel berita dengan laju rata-rata 495 artikel per 24 jam, mengekstraksi 3.079 isu dari tren pemberitaan, dan memvalidasi 333 isu melalui modul validasi bukti. Rerata skor kepercayaan tercatat sebesar 60% dan rerata skor kecocokan fakta sebesar 62%. Capaian ini menunjukkan bahwa seluruh increment dapat berjalan sebagai satu rantai proses yang utuh, bukan sekadar modul yang berdiri sendiri.

3.3 Hasil Pengujian Black Box

Pengujian dilaksanakan terhadap 30 skenario yang mencakup sepuluh modul aplikasi. Ringkasan hasil pengujian per modul ditunjukkan pada Tabel 3.

Tabel 3 Rekapitulasi hasil black box testing per modul

No	Modul yang diuji	Jumlah skenario	Lulus	Tidak lulus
1	Autentikasi dan manajemen peran	3	3	0
2	Akuisisi berita terjadwal	3	3	0
3	Tren isu	3	3	0
4	Penelusuran pengetahuan	3	2	1
5	Validasi bukti	4	4	0
6	Kalibrasi probabilitas	3	3	0
7	Evolusi keyakinan	3	3	0
8	Asesmen intelijen	3	3	0
9	Dasbor operasional	3	3	0
10	Penjadwal proses	2	1	1
	Jumlah	30	28	2

Tabel 3 memperlihatkan bahwa delapan dari sepuluh modul lulus seluruh skenario ujinya, sedangkan modul penelusuran pengetahuan dan penjadwal proses masing-masing menyisakan

satu skenario yang tidak lulus. Rincian skenario yang tidak lulus pada kedua modul tersebut diuraikan pada Tabel 4.

Tabel 4 Rincian skenario uji yang tidak lulus

Skenario	Hasil yang diharapkan	Hasil yang teramati
Menelusuri pengetahuan untuk isu tanpa padanan dokumen	Sistem menampilkan pesan bahwa dokumen pendukung tidak ditemukan	Sistem menampilkan daftar kosong tanpa pesan penjas
Memantau status penjadwal ketika koneksi pembaruan langsung terputus	Sistem menampilkan status terakhir yang diketahui beserta waktu pembaruan	Sistem menampilkan peringatan bahwa angka mungkin tertinggal beberapa detik

Tabel 4 menunjukkan bahwa kedua skenario yang tidak lulus berkaitan dengan penanganan kondisi tepi dan bukan kegagalan pada fungsi utama modul. Berdasarkan persamaan (1), tingkat kelulusan pengujian adalah 28 dibagi 30 dikalikan 100% atau sebesar 93,3%. Kedua skenario yang tidak lulus tidak menghentikan jalannya rantai proses dan tergolong kekurangan pada penanganan kondisi tepi, yaitu ketiadaan pesan penjas saat hasil penelusuran kosong dan ketidakjelasan status ketika koneksi pembaruan langsung terputus. Kedua temuan tersebut dicatat sebagai butir product backlog baru untuk dikerjakan pada sprint berikutnya.

3.4 Pembahasan

Tingkat kelulusan sebesar 93,3% berada sedikit di bawah capaian 95,5% yang dilaporkan pada pengembangan sistem deteksi informasi palsu berbasis BERT sebagaimana diuraikan pada bagian pendahuluan. Perbandingan ini perlu dibaca dengan hati-hati karena jumlah dan cakupan skenario pada kedua penelitian tidak sama, sehingga angka tersebut lebih tepat dipahami sebagai indikasi bahwa kesesuaian fungsional aplikasi berada pada rentang yang wajar dibanding sebagai pembuktian keunggulan.

Nilai utama penerapan Scrum pada penelitian ini terletak pada kemampuannya menyerap perubahan kebutuhan. Dua perubahan yang tercatat, yaitu penyisipan lapisan validasi bukti dan pelatihan ulang model hingga versi ketiga belas, sama-sama muncul dari hasil evaluasi increment dan bukan dari analisis kebutuhan di awal. Pada model Waterfall yang digunakan pada sebagian besar penelitian sejenis, perubahan semacam itu umumnya baru dapat ditangani setelah seluruh tahap selesai atau menuntut pengulangan tahap analisis. Metode prototyping memang menyediakan ruang evaluasi berulang, namun tidak menyediakan artefak pengelolaan prioritas seperti product backlog yang pada penelitian ini terbukti memudahkan penempatan kebutuhan baru tanpa mengganggu pekerjaan yang sedang berjalan.

Pendekatan berorientasi objek yang menekankan modularitas arsitektur tetap relevan dan tidak bertentangan dengan Scrum, karena keduanya bekerja pada aras yang berbeda. Modularitas antarlapisan pada aplikasi TrustNet justru menjadi prasyarat yang memungkinkan setiap sprint menghasilkan increment yang dapat dijalankan secara mandiri. Dengan demikian, temuan penelitian ini menempatkan Scrum sebagai pelengkap praktik perancangan yang telah mapan, bukan sebagai penggantinya.

Keterbatasan penelitian ini perlu dikemukakan. Pertama, pengujian yang dilaporkan terbatas pada kesesuaian fungsional melalui black box testing dan belum mencakup pengujian penerimaan pengguna maupun pengukuran ketergunaan, sehingga penilaian dari sisi pengguna akhir belum tersedia sebagaimana lazim dilakukan pada penelitian pengembangan sistem yang melibatkan responden. Kedua, penelitian dijalankan oleh satu tim pada satu produk, sehingga temuan mengenai keunggulan Scrum belum dapat digeneralisasi tanpa perbandingan langsung terhadap tim lain yang mengerjakan produk serupa dengan metode berbeda. Ketiga, ukuran keberhasilan proses masih bertumpu pada catatan sprint review dan belum menyertakan metrik proses seperti velocity antarsprint.

4. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan beberapa hal berikut.

1. Aplikasi web TrustNet untuk deteksi gangguan informasi berhasil dikembangkan menggunakan metode Scrum melalui enam sprint yang masing-masing menghasilkan increment berupa modul yang dapat dijalankan.
2. Aplikasi berhasil dioperasikan pada aliran berita nyata dengan capaian 2.004 artikel terhimpun, 3.079 isu terekstraksi, dan 333 isu tervalidasi bukti, dengan rerata skor kepercayaan 60% dan skor kecocokan fakta 62%.
3. Pengujian black box terhadap 30 skenario pada sepuluh modul menghasilkan 28 skenario lulus atau tingkat kelulusan sebesar 93,3%, dengan dua ketidaksesuaian berupa penanganan kondisi tepi yang tidak menghentikan jalannya rantai proses.
4. Pada proyek ini, mekanisme sprint review mampu menyerap dua perubahan kebutuhan yang tidak teridentifikasi pada perencanaan awal, yaitu penyisipan lapisan validasi bukti dan pelatihan ulang model hingga versi ketiga belas; temuan ini mencerminkan pengalaman pada satu proyek dan belum tentu dapat digeneralisasi tanpa pengujian pada proyek lain.
5. Kelebihan pendekatan Scrum pada penelitian ini adalah keleluasaan menambahkan kebutuhan baru tanpa mengulang tahap analisis, sedangkan kekurangannya adalah kebutuhan komitmen tim yang konsisten untuk menjalankan seremoni Scrum secara berkelanjutan, sebagaimana umum disyaratkan pada penerapan kerangka kerja ini.

5. SARAN

Beberapa saran untuk penelitian lanjutan dapat dikemukakan sebagai berikut.

1. Melakukan pengujian penerimaan pengguna dan pengukuran ketergunaan dengan melibatkan analisis informasi sebagai responden, sehingga kualitas aplikasi dapat dinilai dari sisi pengguna akhir.
2. Menambahkan pencatatan metrik proses seperti velocity dan burndown chart antarsprint agar kontribusi Scrum terhadap produktivitas pengembangan dapat diukur secara kuantitatif.
3. Melakukan penelitian pembandingan yang mempertemukan tim dengan metode Scrum dan tim dengan metode Waterfall pada spesifikasi produk yang sama untuk menguji generalisasi temuan.
4. Menyempurnakan penanganan kondisi tepi pada modul penelusuran pengetahuan dan penjadwal proses sesuai dua temuan pengujian yang belum lulus.

UCAPAN TERIMA KASIH

Penelitian ini didanai melalui skema Penelitian Dosen Pemula, Direktorat Riset, Teknologi, dan Pengabdian kepada Masyarakat, Kementerian Pendidikan Tinggi, Sains, dan Teknologi Republik Indonesia. Penulis mengucapkan terima kasih kepada Universitas Siber Muhammadiyah atas dukungan fasilitas penelitian.

DAFTAR PUSTAKA

- [1] T. A. Roshinta, E. Kumala, dan I. F. Dinata, "Sistem Deteksi Berita Hoax Berbahasa Indonesia Bidang Kesehatan," *remik*, vol. 7, no. 2, hlm. 1167–1173, 2023, doi: 10.33395/remik.v7i2.12369.
- [2] R. Dicki Prabowo, I. Widaningrum, dan J. Karaman, "Sistem Deteksi Berita Hoax Pemilu 2024 Indonesia Menggunakan Algoritma K-nearest Neighbor (KNN) dan Support Vector Machine (SVM)," *JIKO J. Inform. Dan Komput.*, vol. 9, no. 1, hlm. 93, 2025, doi: 10.26798/jiko.v9i1.1424.
- [3] Y. Zahra, "Understanding and Mitigating AI-Generated Hoax Information," *J. Sist. Inf. Dan Tek. Inform. JAFOTIK*, vol. 3, no. 2, hlm. 56–61, 2025, doi: 10.70356/jafotik.v3i2.84.

-
- [4] I. Pangesti, N. A. Zen, dan D. Kurnianto, "Evaluasi Kinerja Algoritma Naive Bayes pada Sistem Deteksi Berita Hoax," *J. Inform. Commun. Technol. JICT*, vol. 5, no. 2, hlm. 103–110, 2023, doi: 10.52661/j_ict.v5i2.238.
 - [5] K. Mumbunan, M. M. B. Bawata, M. P. Kusen, V. Tarigan, dan A. Yusupa, "Perbandingan Algoritma Naive Bayes Dan Support Vector Machine Dalam Deteksi Berita Hoax Berbahasa Indonesia," *Riau J. Tek. Inform.*, vol. 4, no. 1, 2025, doi: 10.30606/rjti.v4i1.3262.
 - [6] D. Novianto, L. Tommy, S. Sujono, dan Y. S. Japriadi, "Perbandingan Kinerja Model XGBoost dan IndoBERT untuk Deteksi Teks Hoax Terselubung Berbahasa Indonesia," *J. Ilm. Sist. Inf. Dan Ilmu Komput.*, vol. 6, no. 2, hlm. 208–221, 2026, doi: 10.55606/juisik.v6i2.2411.
 - [7] R. R. M. Fahri, "Klasifikasi Teks Berbahasa Indonesia Menggunakan Word2Vec dan K-nearest Neighbor untuk Identifikasi Berita Hoax," *Rabit J. Teknol. Dan Sist. Inf. Univrab*, vol. 11, no. 1, hlm. 122–129, 2026, doi: 10.36341/rabit.v11i1.6674.
 - [8] M. S. Napitupulu, "Deteksi Konten Berita Hoax Tentang Dunia Kesehatan Dimedia Sosial Dengan Menggunakan Metode Text Rank," *J. Ilmu Komput. Teknol. Dan Inf.*, vol. 2, no. 1, hlm. 11–15, 2024, doi: 10.62866/jurikti.v2i1.120.
 - [9] A. Ripa'i, F. Santoso, dan F. Lazim, "Deteksi Berita Hoax dengan Perbandingan Website Menggunakan Pendekatan Deep Learning Algoritma BERT," *G-Tech J. Teknol. Terap.*, vol. 8, no. 3, hlm. 1749–1758, 2024, doi: 10.33379/gtech.v8i3.4541.
 - [10] M. A. Rofiqi dan A. H. Mujianto, "Implementation of Naive Bayes Method and Natural Language Processing in Web-Based Online Hoax News Detection System," *SMATIKA J.*, vol. 16, no. 2, hlm. 216–228, 2026, doi: 10.32664/smatika.v16i02.2338.
 - [11] F. Agustian dan S. Jaya, "Pengembangan Sistem Deteksi Informasi Palsu pada Media Online Menggunakan Algoritma BERT," *Rabit J. Teknol. Dan Sist. Inf. Univrab*, vol. 11, no. 2, hlm. 2751–2760, 2026, doi: 10.36341/rabit.v11i2.7885.
 - [12] D. Tawakalna dan J. Zeniarja, "Sistem Chatbot Kesehatan Mental Berbasis LLM dengan Deteksi Emosi dan Retrieval Augmented Generation," *Rabit J. Teknol. Dan Sist. Inf. Univrab*, vol. 11, no. 1, hlm. 1398–1412, 2026, doi: 10.36341/rabit.v11i1.7347.
 - [13] K. Yuliawan dan P. S. Deda, "Sistem Informasi Kependudukan Berbasis Web dengan Deteksi Duplikasi dan Validasi Semantik NIK," *J. Komput. Teknol. Inf. Sist. Komput. JUKTISI*, vol. 5, no. 2, hlm. 1865–1872, 2026, doi: 10.62712/juktisi.v5i2.1480.
 - [14] Much. R. Maulana, N. I. Royanti, dan R. A. Winoto, "Rancang Bangun Sistem Smart Posbindu Penyakit Tidak Menular (PTM) Berbasis Android untuk Deteksi Dini dan Monitoring Kesehatan Masyarakat," *LogicLink*, 2025, doi: 10.28918/logiclink.v2i2.12989.
 - [15] C. A. P. Tungady dan H. D. Purnomo, "Perancangan Sistem Pendukung Objek Deteksi untuk Permainan Kartu Cardfight! Vanguard Menggunakan Aplikasi Roboflow dan Flask," *J. Teknol. Sist. Inf. Dan Apl.*, vol. 6, no. 3, hlm. 283–290, 2023, doi: 10.32493/jtsi.v6i3.30303.
 - [16] Ahyath, M. Zahrani2, A. Rayadi, S. G. Z. Arnowo, dan Widiatry, "Pengembangan Sistem Deteksi Stunting Balita Berbasis Web Menggunakan Pendekatan OOAD dan Standar Z-Score WHO," *J. Komput. Teknol. Inf. Sist. Komput. JUKTISI*, vol. 5, no. 2, hlm. 1272–1286, 2026, doi: 10.62712/juktisi.v5i2.1327.
 - [17] M. L. Taufikurrahman dan P. Handoko, "Implementasi Algoritma YOLO dan FaceNet dalam Sistem Keamanan untuk Deteksi Orang Tak Dikenal," *Rabit J. Teknol. Dan Sist. Inf. Univrab*, vol. 11, no. 2, hlm. 3865–3881, 2026, doi: 10.36341/rabit.v11i2.8206.
-

- [18] L. M. A. Ikhlas dan Nurdin, “Implementasi Sistem Deteksi Asap dan Api Berbasis IoT untuk Rumah Pintar Menggunakan Metode Prototyping,” *J. Sist. Inf. Kaputama JSIK*, vol. 9, no. 2, hlm. 98–107, 2025, doi: 10.59697/jsik.v9i2.1016.
- [19] L. B. Herdianto, “Pengembangan Sistem Pakar Deteksi Anemia Kehamilan Menggunakan Certainty Factor,” *KOMPUTEK*, vol. 7, no. 2, hlm. 38–48, 2023, doi: 10.24269/jkt.v7i2.2240.
- [20] F. Anwar dan A. Voutama, “Implementasi Sistem Deteksi Defacement Otomatis pada Website Pemerintah Menggunakan N8N,” *J. Teknol. Dan Sist. Inf. Bisnis*, vol. 8, no. 2, hlm. 153–158, 2026, doi: 10.47233/jteksis.v8i2.2578.
- [21] M. H. Z. Hidayatullah, D. Fitrianto, M. Rosidin, M. F. Gustafi, dan K. E. Setyaputri, “Perancangan UI/UX Website PDM Kebumen dengan Metode User Centered Design (UCD),” *Insect J. Inform. dan Secur.*, vol. 12, no. 1, hlm. 147–157, 2026, doi: 10.33506/insect.v12i01.5500.
- [22] P. Widiandana, M. H. Amali, A. Admaja, dan M. R. Saputra, “Perancangan Keamanan Informasi pada Sistem Persuratan Berbasis Web Menggunakan Autentikasi dan Middleware,” *Bull. Comput. Sci. Res.*, vol. 6, no. 4, hlm. 1544–1551, 2026.
- [23] P. Widiandana, M. A. Muttaqin, H. M. Riziq, dan F. A. Muiz, “Analisis Penerapan CAPTCHA, Autentikasi, dan Access Control pada Keamanan Informasi Website Akademik,” *JIFKOM J. Ilm. Inform. dan Komput.*, vol. 5, no. 2, hlm. 60–66, 2026.
- [24] B. J. Nababan, A. Haikal, dan S. R. Maulidya, “Pembuatan Sistem Deteksi Hardcode Kredensial pada Repository,” *ROUTERS J. Sist. Dan Teknol. Inf.*, hlm. 20–31, 2025, doi: 10.25181/rt.v3i1.3898.