

PERANCANGAN APLIKASI *TRACER STUDY* DENGAN METODE *AGILE* UNTUK EVALUASI KESELARASAN KARIER ALUMNI

Mubarad Al Ghyfari¹, Rony Setiawan²

Program Studi Manajemen Informatika, Fakultas, Politeknik LP3I Bandung

Jl. Pahlawan No 59, Sukaluyu, Kec. Cibeunying Kaler, Bandung

E-mail: *mubaradalghyfari@gmail.com

Abstrak – *Tracer Study* merupakan metode yang digunakan untuk melacak keberlanjutan alumni dalam dunia kerja dan akademik, serta mengevaluasi kesesuaian kurikulum dengan kebutuhan industri yang terus berkembang. Penelitian ini bertujuan untuk merancang dan mengembangkan aplikasi *tracer study* berbasis web untuk mengukur keselarasan horizontal dan keselarasan vertikal, serta memberikan *output* berupa rekomendasi atensi terhadap aspek-aspek yang perlu dievaluasi atau disesuaikan agar lebih selaras dengan kebutuhan industri dan dunia kerja. Aplikasi ini dirancang dengan antarmuka yang responsif dan interaktif untuk mendukung visualisasi data *real-time*. Metode pengembangan yang digunakan adalah *Agile* dengan tahapan analisis data, desain sistem, pengembangan dan pengujian. Aplikasi ini dikembangkan menggunakan teknologi React.js untuk frontend, Express.js untuk backend, dan MongoDB sebagai database utama dalam pengelolaan data alumni serta hasil *tracer study*. Hasil penelitian menunjukkan bahwa aplikasi ini mampu meningkatkan efisiensi pengumpulan data, akurasi analisis, dan memberikan rekomendasi strategis untuk penyesuaian kurikulum sesuai tren pasar kerja. Dengan demikian, aplikasi ini berkontribusi pada pengembangan sistem informasi pendidikan yang inovatif dan relevan dengan kebutuhan industri.

Kata Kunci: *Agile*, Sistem Informasi, *Tracer Study*

I. PENDAHULUAN

Tracer Study menjadi salah satu metode paling penting dalam mengevaluasi keberlanjutan lulusan dalam dunia kerja dan akademik. Perguruan tinggi perlu mengetahui bagaimana keselarasan antara kompetensi yang diberikan selama perkuliahan dengan kebutuhan industri agar dapat melakukan perbaikan kurikulum yang lebih adaptif. Hal ini berkaitan dengan aspek relevansi sebagai salah satu indikator keberhasilan perguruan tinggi, dimana institusi pendidikan bertanggung jawab untuk menghasilkan lulusan yang memiliki daya saing dan mampu bersaing di dunia kerja (Fitriani et al., 2024). Keselarasan antara pendidikan dan pekerjaan dapat diukur melalui dua aspek utama, yaitu keselarasan vertikal dan horizontal. Keselarasan vertikal mengacu pada sejauh mana tingkat pendidikan alumni sesuai dengan pekerjaan yang dijalani. Pengukuran keselarasan vertikal dapat dilakukan menggunakan skala likert(1-4), di mana skor 1 menunjukkan ketidakseusian total, dan skor -4 menunjukkan kesesuaian penuh antara bidang studi dan pekerjaan. Sedangkan keselarasan horizontal mengukur keterkaitan antara bidang keilmuan yang dipelajari dengan pekerjaan, terlepas dari jenjang pendidikannya. Namun, proses *tracer study* yang masih menggunakan *google form* sering kali kurang efektif dan cukup memakan waktu dalam pengelolaan data, serta memiliki partisipasi alumni yang rendah.

Seiring dengan perkembangan teknologi yang semakin pesat telah menghasilkan inovasi yang lebih canggih, cepat dan mudah diakses, salah

satunya adalah aplikasi berbasis *website* yang dapat digunakan melalui *browser* dengan koneksi internet atau intranet (Nursyanti et al., 2019; Susanto et al., 2020). Dalam konteks pendidikan, digitalisasi sistem *tracer study* menjadi solusi efektif dalam pengumpulan data alumni. Sistem ini memungkinkan pelacakan data lulusan dilakukan dengan lebih terstruktur serta dilengkapi fitur interaktif dan tampilan yang *user-friendly* guna meningkatkan partisipasi alumni. Dalam merancang situs web, aspek *user interface* (UI) dan *user experience* (UX) harus di perhatikan dengan baik. UI dirancang untuk memastikan kenyamanan pengguna dalam berinteraksi dengan sistem, sementara UX berfokus pada pengalaman pengguna sebagai faktor utama dalam proses perancangan (Billy Indra Irawan, 2022). Oleh karena itu, aspek *user experience* menjadi faktor penting yang perlu diperhatikan dalam desain sistem, sehingga aplikasi tidak hanya berfungsi dengan baik tetapi juga memberikan pengalaman yang optimal bagi pengguna.

Dalam pengembangan sistem, pemilihan teknologi yang tepat menjadi faktor krusial agar sistem dapat berjalan dengan optimal. Pada sisi *frontend* React.js dipilih sebagai *framework* dari *javascript* yang memiliki keunggulan dalam hal kecepatan, kesederhanaan dan skalabilitas (Khalif & Wibowo, 2022). Dengan pendekatan berbasis komponen, React.js memungkinkan pengembang untuk membangun antarmuka yang lebih interaktif dan dinamis. Selain itu, dalam arsitektur *Model-View-Controller* (MVC), React.js hanya berfokus pada bagian view, sehingga proses pengembangan

menjadi lebih terstruktur dan efisien (Nursaid et al., 2020; Panjaitan & Pakpahan, 2021). Sementara itu pada sisi *backend* menggunakan Express.js untuk mengelola komunikasi data secara cepat dan efisien, disebutkan juga Express.js terintegrasi dengan mesin rendering view atau bagian *frontend*, memungkinkan pengembangan untuk mengelola *respons* dengan lebih efisien dengan cara memasukan data secara langsung kedalam tampilan atau antarmuka pengguna (Nasution, 2021).

Untuk memastikan sistem yang dikembangkan dapat memenuhi kebutuhan pengguna, perancangan sistem dijelaskan menggunakan *Use Case Diagram*. Diagram ini digunakan untuk menggambarkan interaksi antara aktor dengan sistem, termasuk bagaimana fitur-fitur yang tersedia dapat digunakan oleh masing-masing pengguna.

Untuk penyimpanan data, MongoDB dipilih sebagai basis data utama, karena MongoDB merupakan basis data *open-source* yang populer dalam menangani data berukuran besar, sambil tetap mempertahankan performa tinggi dan kemudahan dalam skalabilitas (Aziira et al., 2023). Sebagai *database NoSQL*, MongoDB menawarkan fleksibilitas yang lebih baik dibandingkan sistem basis data relasional. Sistem basis data relasional (RDBMS) lebih menekankan pada konsistensi data, sedangkan *NoSQL* unggul dalam skalabilitas yang lebih tinggi, memungkinkan pemrosesan yang lebih cepat serta mampu menangani beban kerja dalam jumlah besar (Susanti et al., 2020).

Penelitian sebelumnya menunjukkan bahwa database *NoSQL* lebih unggul dalam menangani pertumbuhan data pada era *Big Data* dibandingkan database relasional, di mana studi membandingkan performa database *Redis (NoSQL)* dengan *MySQL* dan menemukan bahwa Redis memiliki waktu eksekusi yang lebih cepat dalam operasi dasar, yaitu *create* (87.58% lebih cepat), *update* (85.53%), dan *read* (57.09%), dengan rata rata keunggulan performa sebesar 79.15% lebih baik dari *MySQL* sehingga penggunaan *NoSQL* secara efektif dapat menangani data yang dinamis dan berkembang pesat (Fadli et al., 2020).

Dengan fleksibilitas ini, MongoDB lebih cocok untuk aplikasi yang memerlukan skalabilitas tinggi dan manajemen data yang dinamis, serta menjadi solusi dalam pengelolaan data besar yang terus berkembang pesat, termasuk dalam sistem *tracer study* yang membutuhkan akses data *real-time* dan analisis yang lebih akurat.

II. TINJAUAN PUSTAKA

Tracer Study

Salah satu indikator keberhasilan perguruan tinggi adalah tingkat relevansinya. Dalam hal ini, perguruan tinggi bertanggung jawab sebagai penyelenggara pendidikan yang dapat menghasilkan lulusan berdaya saing tinggi di dunia

kerja. *Tracer Study* berfungsi untuk menganalisis keterkaitan antara *input* pendidikan, proses pembelajaran, *output* yang dihasilkan, serta dampaknya. Selain itu, *Tracer Study* juga menjadi salah satu persyaratan dalam proses akreditasi yang ditetapkan oleh Badan Akreditasi Nasional Perguruan Tinggi (BAN-PT) (Fitriani et al., 2024).

Website

Aplikasi berbasis web adalah salah satu bentuk kemajuan teknologi yang dapat diakses melalui *browser* menggunakan jaringan seperti internet atau intranet (Nursyanti et al., 2019). Kemajuan teknologi informasi telah menjadi bagian yang tidak terpisahkan dari kehidupan *modern*. Dibandingkan dengan teknologi di masa lalu, perkembangan ini berlangsung dengan lebih pesat, menghasilkan sistem yang semakin canggih, efisien, dan cepat (Susanto et al., 2020).

Agile

Konsep *Agile Software Development* pertama kali diperkenalkan secara luas pada tahun 2001 oleh Kent Beck bersama 16 rekannya. *Agile Software Development* didefinisikan sebagai pendekatan dalam membangun perangkat lunak dengan menerapkannya secara langsung serta membantu orang lain dalam proses pengembangannya. Metode ini menggunakan pendekatan pengembangan secara bertahap (*incremental*) yang menekankan kecepatan dalam pengembangan, perilsan perangkat lunak secara berkala, efisiensi proses, serta menghasilkan kode berkualitas tinggi dengan keterlibatan aktif dari pengguna dalam setiap tahap pengembangannya (Panjaitan & Pakpahan, 2021).

Javascript

Javascript merupakan bahasa pemrograman berbasis skrip yang dijalankan dalam dokument HTML. Pengguna *Javascript* memungkinkan peningkatan tampilan serta fungsionalitas pada aplikasi berbasis web yang dikembangkan (Mariko, 2019).

React.js

React.js merupakan *library* dari bahasa pemrograman *Javascript* yang bersifat *open source* dan digunakan untuk mengembangkan antarmuka *website* (Khalif & Wibowo, 2022). React.js menawarkan berbagai keunggulan, seperti kecepatan, kesederhanaan, dan skalabilitas. Dengan arsitektur MVC (*Model-View-Controller*), React.js berperan dalam menangani bagian *view* saja (Panjaitan & Pakpahan, 2021).

Express.js

Express.js adalah *framework* aplikasi web untuk Node.js yang ringan dan fleksibel, serta menawarkan berbagai fitur yang kuat untuk pengembangan aplikasi web maupun mobile. Saat

ini, Express.js menjadi salah satu framework yang paling populer karena menyediakan berbagai mekanisme yang memudahkan dalam penggunaannya (Nasution, 2021).

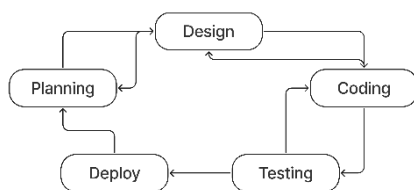
MongoDB

MongoDB adalah salah satu jenis basis data non-relasional (*NoSQL*) yang dikembangkan untuk mengatasi kendala performa pada sistem manajemen basis data relasional (RDBMS), terutama dalam menangani data tidak terstruktur seperti dokumen, media sosial, dan email. Basis data *NoSQL* dirancang untuk memproses data tidak terstruktur dengan lebih efisien (Aziira et al., 2023).

III. METODE PENELITIAN

Pengembangan sistem *tracer study* ini menggunakan metode Agile, yang memungkinkan proses pengembangan dilakukan secara iterative dan adaptif sesuai dengan kebutuhan pengguna. Agile merupakan salah satu metode pengembangan perangkat lunak yang populer saat ini karena menawarkan fleksibilitas tinggi dalam menyesuaikan perubahan kebutuhan dalam waktu yang relatif singkat. Selain itu, pendekatan ini memungkinkan pengembang untuk beradaptasi secara cepat selama proses pengembangan, sehingga meningkatkan efisiensi dalam pengembangan perangkat lunak (Puspa Putri, 2019; Putra & Tanaem, 2022).

Dalam pengembangan sistem *tracer study* LP3I, metode Agile diterapkan dengan pendekatan *scrum*, yang membagi pengembangan menjadi beberapa sprint dengan tahapan yang bisa dilihat pada Gambar 1.



Gambar 1. Diagram alur pengembangan Agile

Diagram ini menggambarkan tahapan iteratif dalam pengembangan aplikasi *tracer study*, dimulai dari perencanaan, analisis kebutuhan, desain, implementasi, pengujian, hingga evaluasi dan perbaikan secara berulang. Pendekatan ini memungkinkan fleksibilitas dalam menyesuaikan fitur aplikasi berdasarkan *feedback* pengguna.

1. Planning (Perencanaan)

Tahap ini merupakan langkah awal dalam pengembangan sistem, dimana dilakukan identifikasi kebutuhan sistem serta perencanaan strategi pengembangannya. Pada tahap ini, dilakukan analisis terhadap fitur-fitur yang

akan dikembangkan, seperti pengisian survei alumni, analisis keselarasan horizontal dan vertikal, serta ditentukan juga teknologi yang digunakan dalam pengembangan sistem, yaitu MongoDB sebagai database utama. Perencanaan ini dilakukan dengan membagi fitur-fitur sistem ke dalam beberapa *sprint* agar pengembangannya dapat dilakukan secara bertahap dan berkelanjutan.

2. Design (Perancangan)

Pada tahap ini, dilakukan perancangan sistem untuk memastikan bahwa sistem yang dibangun sesuai dengan kebutuhan pengguna. Perancangan yang dilakukan meliputi arsitektur sistem, desain *UI/UX*, serta perancangan alur kerja aplikasi. Perancangan arsitektur sistem bertujuan untuk menggambarkan hubungan antara *frontend*, *backend*, dan *database* dalam pengelolaan data *tracer study*. Selain itu, melakukan desain antarmuka pengguna (*UI/UX*) untuk memastikan aplikasi mudah digunakan oleh alumni dan admin kampus. Perancangan sistem ini juga mencakup pembuatan diagram dan *use case*, yang menggambarkan skenario interaksi pengguna dengan sistem.

3. Coding (Pengembangan Program)

Setelah tahap desain selesai, pengembangan sistem dilakukan dengan menerapkan *code* program sesuai dengan spesifikasi yang telah dirancang. Pada tahap ini, *frontend* dikembangkan menggunakan React.js yang bertanggung jawab dalam pembuatan tampilan aplikasi dan interaksi pengguna. *Backend* dikembangkan menggunakan Express.js, yang berfungsi sebagai penyedia *API* (*Application Programming Interface*) untuk memproses data *tracer study* dan menangani logika bisnis aplikasi. Untuk penyimpanan data, menggunakan MongoDB, yang dipilih karena fleksibilitasnya dalam menangani data yang dinamis dan skalabilitas yang tinggi. Pengembangan dilakukan dalam beberapa *sprint*, di mana setiap *sprint* menghasilkan fitur yang dapat diuji sebelum masuk ke tahap selanjutnya.

4. Testing (Pengujian)

Setelah fitur dikembangkan, dilakukan tahap pengujian untuk memastikan bahwa sistem berjalan dengan spesifikasi yang sesuai dan telah ditentukan. Pengujian dilakukan dengan metode *Black Box Testing*, yang berfokus pada pengujian fungsionalitas sistem, seperti *login*, *input* data alumni, analisis keselarasan, serta pemberian atensi kepada pihak kampus. Selain itu, dilakukan juga pengujian performa *database* untuk memastikan bahwa MongoDB dapat menangani jumlah data alumni dalam skala besar tanpa mengalami penurunan performa. Evaluasi sistem juga dilakukan

dengan melibatkan pengguna akhir, yaitu pengelola akademik dan alumni, guna mendapatkan umpan balik terkait penggunaan sistem *launching* sepenuhnya.

Dengan menerapkan metode *Agile* dalam pengembangan dapat dilakukan secara lebih fleksibel dan adaptif terhadap kebutuhan pengguna. Jika ditemukan perubahan atau peningkatan fitur berdasarkan hasil pengujian dan evaluasi, sistem dapat segera diperbarui dalam iterasi berikutnya hanya tanpa mengganggu pengembangan yang sudah berjalan.

Dalam penelitian ini, data dikumpulkan menggunakan beberapa metode untuk memastikan keakuratan serta relevansi informasi yang diperoleh. Adapun metode yang digunakan sebagai berikut:

1. Studi Literatur

Penelitian dengan mengumpulkan referensi dari berbagai sumber, seperti jurnal ilmiah, buku, dan penelitian sebelumnya yang berkaitan dengan *tracer study* serta pengelolaan data alumni. Metode ini bertujuan untuk mendapatkan dasar teori yang kuat dalam pengembangan sistem.

2. Observasi

Observasi dilakukan dengan melihat langsung bagaimana proses *tracer study* sebelumnya dilakukan. Dari hasil observasi ini, diperoleh pemahaman mengenai kekurangan sistem lama dan kebutuhan yang harus dipenuhi oleh sistem yang dikembangkan.

3. Wawancara

Wawancara dilakukan dengan pihak kampus, khususnya bagian akademik yang nantinya akan menjadi bagian yang menangani *tracer study*, untuk mengetahui kebutuhan sistem serta fitur yang diharapkan. Informasi yang didapat membantu dalam perancangan sistem agar sesuai dengan kebutuhan pengguna.

4. Kuesioner

Pengumpulan data juga dilakukan melalui kuesioner yang diberikan kepada alumni. Tujuannya adalah mendapatkan informasi langsung mengenai pengalaman pengguna dalam pengisian *tracer study*, serta kendala yang mungkin mereka hadapi pada sistem sebelumnya.

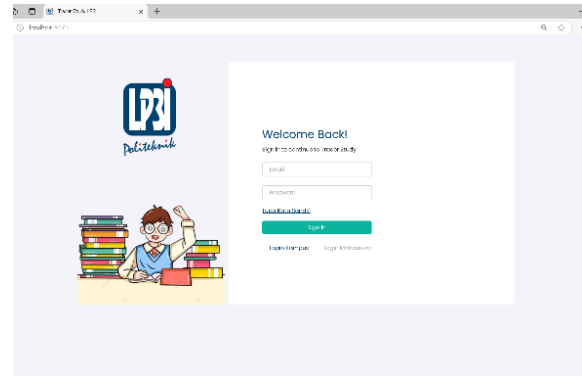
IV. HASIL DAN PEMBAHASAN

A. Hasil Pengembangan Aplikasi

Aplikasi *Tracer Study* yang telah dikembangkan merupakan sistem berbasis *website* yang dirancang untuk membantu institusi pendidikan dalam melacak keberlanjutan alumni di dunia kerja dan akademik. Berikut ini adalah penjelasan beberapa fitur utama yang telah diimplementasikan meliputi:

1. Halaman *Login*

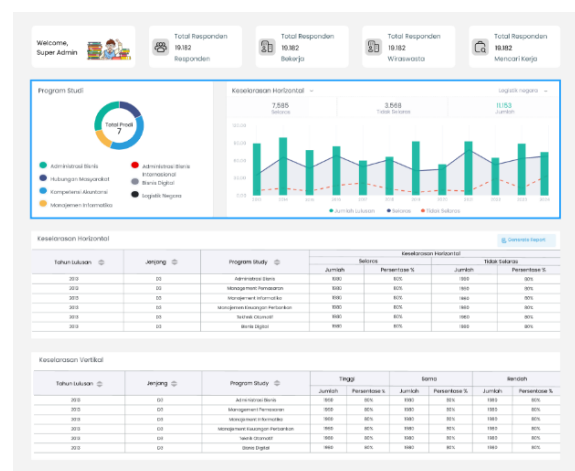
Fitur ini memungkinkan pengguna untuk masuk ke dalam sistem berdasarkan hak akses yang dimiliki. Setiap pengguna harus memasukkan kredensial yang valid untuk dapat mengakses fitur sesuai perannya.



Gambar 2. Halaman *Login*

2. Halaman *Dashboard*

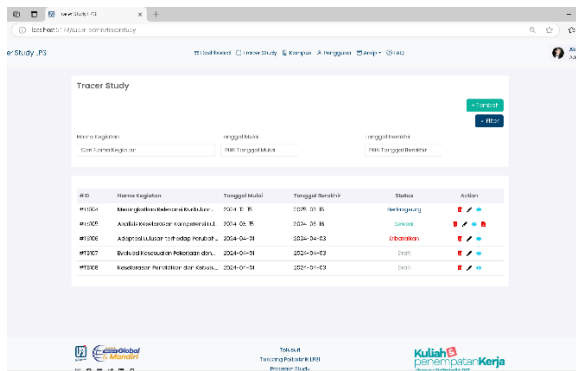
Dashboard berfungsi sebagai pusat informasi yang menampilkan ringkasan data *tracer study*, seperti jumlah alumni yang telah bekerja atau melanjutkan studi, serta analisis keselarasan kompetensi alumni dengan industri dalam bentuk grafik dan statistik *real-time*.



Gambar 3. Halaman *Dashboard*

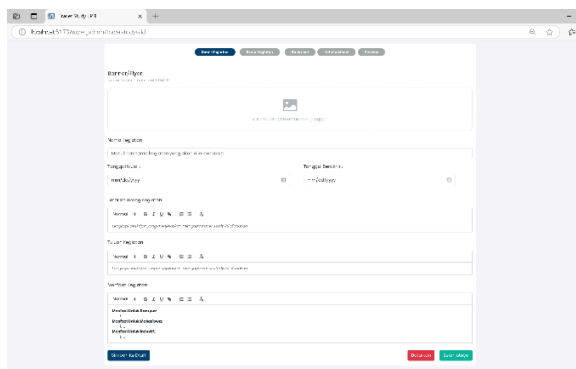
3. Halaman *List Tracer Study*

Fitur ini memungkinkan pengelola akademik untuk mengelola data *tracer study*, termasuk menambahkan, mengedit, dan menghapus kegiatan tracer.

Gambar 4. Halaman *List Tracer Study*

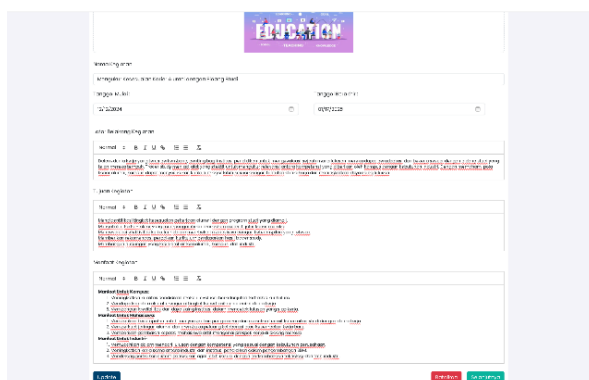
4. Halaman Tambah Kegiatan *Tracer*

Fitur ini dapat diakses melalui tombol tambah, untuk menambah kegiatan *tracer study* baru dengan mengisi informasi yang dibutuhkan, seperti tahun pelaksanaan, kategori tracer, dan detail lainnya.

Gambar 5. Halaman Tambah Kegiatan *Tracer*

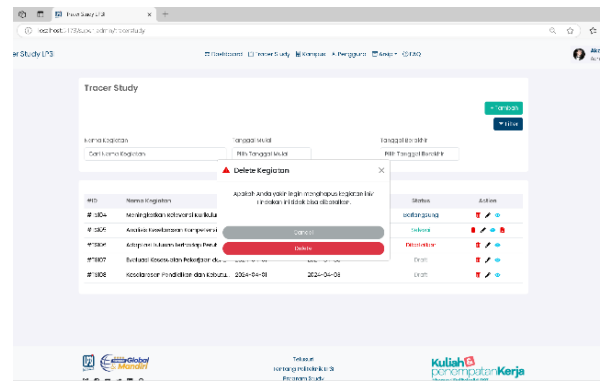
5. Halaman *Edit Kegiatan Tracer*

Fitur ini berfungsi jika terdapat kesalahan atau perubahan data kegiatan, memungkinkan untuk dapat melakukan *edit* pada data kegiatan *tracer study*.

Gambar 6. Halaman *Edit Kegiatan Tracer*

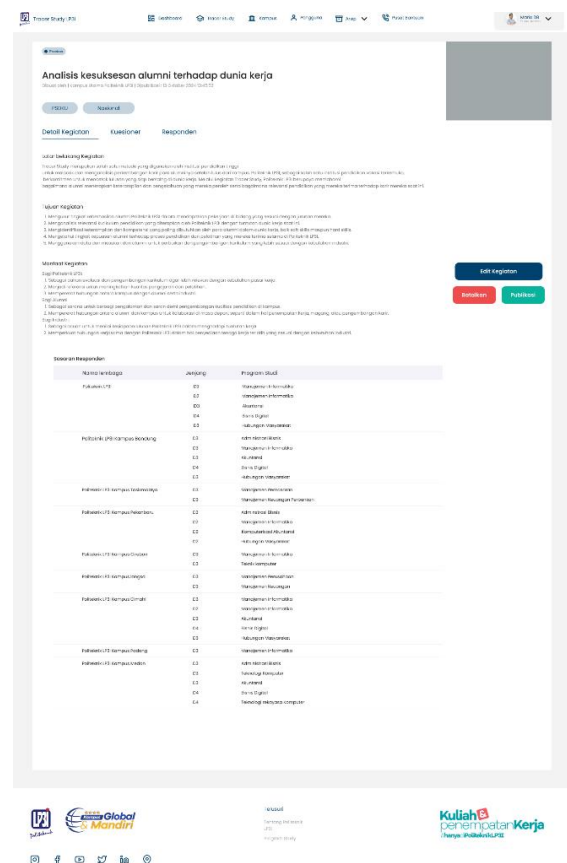
6. Hapus Kegiatan *Tracer*

Apabila terdapat kegiatan *tracer* yang tidak relevan atau sudah tidak diperlukan, maka akan bisa menghapusnya dengan menekan *icon* sampah.

Gambar 7. Hapus Kegiatan *Tracer*

7. Preview Kegiatan *Tracer*

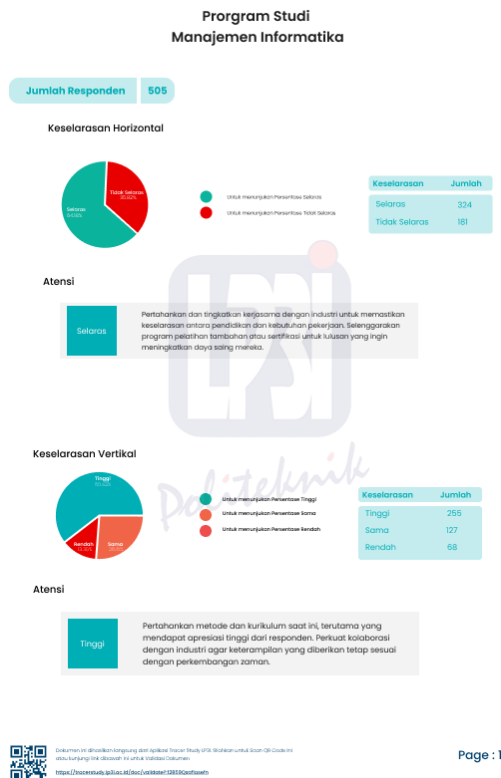
Fitur ini berfungsi untuk melihat *detail* kegiatan *tracer*, yang Dimana berfungsi juga untuk melakukan pembatalan sebelum kegiatan di *publish*.

Gambar 8. Halaman *Preview Kegiatan Tracer*

8. *Report PDF*

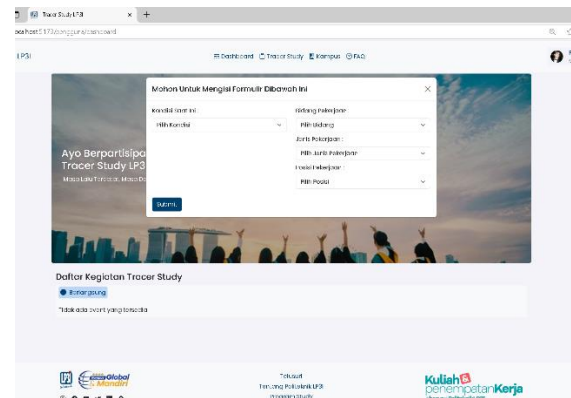
Fitur ini berfungsi untuk menghasilkan laporan dalam bentuk PDF yang berisi rekomendasi serta atensi berdasarkan hasil *tracer study* yang telah dikumpulkan. Laporan ini dirancang untuk memberikan gambaran menyeluruh mengenai keselarasan antara program studi dengan dunia kerja, sehingga dapat menjadi acuan bagi pihak akademik, pengelola institusi,

maupun pemangku kebijakan dalam melakukan evaluasi dan perbaikan kurikulum. Dengan adanya laporan ini, perguruan tinggi dapat lebih mudah mengidentifikasi aspek yang perlu ditingkatkan, menyesuaikan kebijakan Pendidikan dengan kebutuhan industri, serta meningkatkan kualitas lulusan agar lebih siap menghadapi dunia kerja.

Gambar 9. Halaman *Report PDF*

9. *Dashboard Alumni*

Dashboard alumni berfungsi sebagai pusat informasi bagi alumni yang mengikuti *tracer study*. Di dalamnya terdapat dua fitur utama, yaitu pengisian form keselarasan horizontal untuk mengukur relevansi pekerjaan dengan bidang studi, serta event kegiatan *tracer study* yang menampilkan informasi acara terkait alumni. Fitur ini memudahkan alumni dalam berpartisipasi dan tetap terhubung dengan kampus.

Gambar 10. Halaman *Dashboard Alumni*

10. *Form Kuesioner*

Form kuesioner berisi pertanyaan-pertanyaan yang digunakan untuk mengukur keselarasan vertikal, yaitu sejauh mana pendidikan yang diperoleh oleh alumni sesuai dengan bidang kerja yang mereka jalani.

Gambar 11. Halaman *Form Kuesioner*

Pada umumnya, aplikasi *tracer study* hanya berfokus pada pengukuran keselarasan horizontal dan vertikal alumni dengan dunia kerja. Keselarasan horizontal mengacu pada keterkaitan antara bidang studi yang diambil alumni dengan pekerjaan mereka saat ini, sedangkan keselarasan vertikal menilai apakah kurikulum yang diberikan oleh institusi Pendidikan sudah sesuai dengan tren pasar kerja saat ini.

Namun, dalam perancangan aplikasi *Tracer Study LP3I*, terdapat fitur tambahan yang menjadi kelebihan utama, yaitu fitur atensi. Fitur ini tidak hanya mengukur keselarasan horizontal dan vertikal, tetapi juga memberikan peringatan atau rekomendasi kepada pihak kampus berdasarkan rata-rata nilai yang diperoleh dari hasil *tracer study*. Dengan adanya fitur ini, kampus dapat dengan lebih cepat mengidentifikasi aspek yang perlu dievaluasi atau di sesuaikan dengan kebutuhan industri.

Atensi yang diberikan berdasarkan kategori nilai yang dihasilkan dari analisis data alumni, yang memberikan gambaran mengenai parameter yang digunakan untuk menentukan apakah sebuah program studia atau kampus memerlukan Tindakan

evaluasi lebih lanjut. Nilai dan rekomendasi atensi dapat dilihat pada Table 1 dan Table 2.

Tabel 1. Kategori Keselarasan Vertikal

No	Deskripsi	Kriteria	Atensi
1	Mayoritas responden menilai bahwa prodi telah memberikan keterampilan yang relevan	Tinggi	Pertahankan kurikulum yang di apresiasi dan perkuat kerja sama dengan industri
2	Sebagian responden merasa ada kesesuaian, tetapi belum optimal	Sama	Tingkatkan kerja sama dengan industri dan selenggarakan pelatihan atau sertifikasi guna meningkatkan daya saing lulusan.
3	Mayoritas responden menilai prodi belum selaras atau tidak relevan dengan karir mereka	Rendah	Identifikasi dan sesuaikan mata kuliah yang kurang relevan, serta tingkatkan pelatihan soft dan hard skills.

Tabel 2. Kategori Keselarasan Horizontal

No	Kriteria	Min/Max	Atensi
1	Sangat Tidak Selaras	0 – 20	Evaluasi kurikulum dan metode pengajaran untuk mengidentifikasi penyebab lulusan sulit untuk mendapatkan pekerjaan relevan.
2	Tidak Selaras	21 – 40	Perbaiki program pembelajaran dengan pelatihan keterampilan praktis serta adakan forum diskusi dengan pemberi kerja.
3	Cukup Selaras	41 – 60	Pantau relevansi keterampilan secara berkala dan adakan workshop serta seminar dengan praktisi industri.
4	Selaras	61 – 80	Tingkatkan kerja sama dengan industri dan

			selenggarakan pelatihan atau sertifikasi guna meningkatkan daya saing lulusan.
5	Sangat Selaras	81 – 100	Pertahankan metode terbaik dalam pengajaran serta promosikan keberhasilan alumni untuk meningkatkan reputasi institusi.

B. Analisis Sistem

Sebelumnya, *tracer study* dilakukan secara manual menggunakan *google form*. Metode ini memiliki beberapa kekurangan, seperti kesulitan dalam pengelolaan data, serta kurangnya visualisasi data yang memudahkan pengambilan Keputusan. Kampus harus mengelola data secara manual untuk mendapatkan *output* terkait keselarasan lulusan dengan dunia kerja, yang memakan waktu.

Sebagai solusi, *Tracer Study* LP3I dirancang dalam bentuk *website base system* yang memungkinkan pengelolaan data secara lebih efisien. Dengan aplikasi ini, data yang dikumpulkan dapat langsung diolah dan dianalisis, sehingga memberikan hasil yang lebih cepat dan akurat. Selain itu, fitur atensi yang ditambahkan dalam system ini memungkinkan pihak kampus untuk segera mengambil Tindakan berdasarkan hasil *tracer study*.

Dari sisi teknologi, aplikasi ini menggunakan React.js untuk *frontend* karena kemampuannya dalam membangun *user interface* yang interaktif dan responsif, sehingga meningkatkan pengalaman pengguna. *Backend* menggunakan Express.js, yang ringan dan cepat dalam mengolah data. Sedangkan database yang digunakan adalah MongoDB, yang fleksibel dalam mengolah data tidak terstruktur, seperti hasil survei dari alumni yang memiliki format bervariasi, dengan kombinasi teknologi ini, system *Tracer Study* LP3I dapat memberikan pengalaman yang lebih *user-friendly*, data yang lebih terstruktur, serta memungkinkan analisis yang lebih mendalam untuk meningkatkan kualitas pendidikan.

Model kebutuhan fungsional system divisualisasikan menggunakan *Use Case Diagram*, yang menggambarkan bagaimana aktor berinteraksi dengan sistem. Diagram ini membantu dalam memahami fitur yang tersedia dalam sistem serta pihak yang dapat mengaksesnya.

Sistem ini melibatkan tiga aktor utama, yaitu admin, alumni dan pihak kampus.

1. Admin memiliki kendali penuh dalam mengelola data *tracer study*, menganalisis tingkat keselarasan lulusan dengan dunia kerja, serta menentukan langkah perbaikan berdasarkan hasil analisis.

2. Alumni bertindak sebagai responden yang mengisi kuesioner *tracer study*, yang menjadi sumber utama dalam penentuan tingkat keselarasan horizontal dan vertikal.
3. Pihak kampus memanfaatkan hasil analisis *tracer study* sebagai bahan evaluasi guna menyesuaikan kurikulum dengan kebutuhan industri.

Interaksi antara para aktor dengan sistem Tracer Study LP3I digambarkan dalam *Use Case Diagram* yang dapat dilihat pada gambar 1 Diagram alur pengembangan Agile

C. Rancangan Sistem

Rancangan sistem Tracer Study LP3I dijelaskan dari segi arsitektur sistem dan alur kerja aplikasi. Rancangan ini bertujuan untuk memberikan gambaran mengenai bagaimana komponen-komponennya saling berelasi, serta bagaimana data diproses dalam sistem.

1. Arsitektur Sistem

Sistem Tracer Study LP3I ini dirancang menggunakan arsitektur berbasis web yang terdiri dari tiga lapisan utama, yaitu *frontend*, *backend*, dan *database*. Masing-masing memiliki peran spesifik dalam mengelola data dan menyediakan layanan kepada pengguna.

a. *Frontend (Client-Side)*

Dibangun menggunakan React.js, yang memungkinkan tampilan antarmuka yang dinamis, interaktif, dan responsif. Pengguna dapat mengakses sistem melalui browser untuk melakukan pengisian *tracer study*, melihat hasil analisis, serta mendapatkan rekomendasi berdasarkan data yang telah dikumpulkan.

b. *Backend (Server-Side)*

Menggunakan Node.js dengan *framework* Express.js untuk menangani permintaan dari *frontend*. *Backend* bertanggung jawab dalam pengelolaan data, otentikasi pengguna, serta penyajian data kepada pengguna sesuai dengan perannya dalam sistem.

c. *Database*

Penyimpanan data menggunakan MongoDB, yang mampu menangani data tidak terstruktur seperti hasil survei dari alumni dengan format bervariasi. MongoDB memungkinkan pencarian data yang cepat serta fleksibilitas dalam manajemen informasi *tracer study*.

Dengan arsitektur ini, sistem mampu mengelola data dengan efisien serta memberikan pengalaman pengguna yang optimal.

2. Database Tracer Study LP3I

Salah satu komponen krusial dalam sistem ini adalah *database*. Oleh karena itu, sistem Tracer Study LP3I dirancang untuk menyimpan dan mengelola informasi *tracer study* secara efisien,

mendukung pencarian data yang cepat, serta memberikan fleksibilitas dalam proses analisis. Dengan desain ini, sistem dapat memastikan bahwa data alumni tetap aman, mudah diakses, dan dapat digunakan untuk keperluan evaluasi akademik maupun pengambilan keputusan strategis oleh pihak kampus.

D. Implementasi Sistem

Pada tahap ini, sistem Tracer Study LP3I mulai direalisasikan dalam bentuk kode program dengan menggunakan teknologi yang telah ditentukan. Implementasi sistem mencakup penggunaan teknologi, proses pengembangan aplikasi, serta aspek keamanan untuk memastikan sistem berjalan dengan optimal dan aman digunakan.

1. Teknologi yang Digunakan

Sistem ini dikembangkan menggunakan teknologi berbasis web dengan *stack* sebagai berikut:

a. *Frontend*

Dibangun menggunakan React.js. Untuk komunikasi dengan *backend*, digunakan Axios sebagai HTTP *client* untuk melakukan *request* ke *API*.

b. *Backend*

Menggunakan Node.js dengan *framework* Express.js, yang berfungsi sebagai *server* untuk menangani permintaan dari *frontend*, mengelola data, serta menyediakan berbagai *endpoint API*

c. *Database*

MongoDB digunakan sebagai *database* utama.

d. Autentikasi

Menggunakan JWT (*JSON Web Token*) untuk mengamankan akses ke sistem, JWT memungkinkan pengguna mendapatkan token yang akan digunakan dalam setiap *request* ke *server*, sehingga hanya pengguna yang terautentikasi yang bisa mengakses data.

e. Keamanan Data

Bcrypt digunakan untuk melakukan *hashing* pada password pengguna sebelum disimpan ke *database*, sehingga meningkatkan keamanan sistem.

2. Proses Pengembangan

Proses pengembangan sistem *Tracer Study* LP3I dilakukan secara bertahap agar sistem dapat berjalan dengan baik dan optimal. Tahapan pengembangan ini mencakup perancangan sistem, pengembangan *frontend*, pengembangan *backend*, dan integrasi *API*.

a. Perancangan Sistem

Sebelum implementasi kode, dilakukan perancangan sistem untuk memastikan struktur dan alur kerja aplikasi sesuai dengan kebutuhan. Perancangan ini mencakup Pembuatan *use case diagram* untuk

menggambarkan interaksi pengguna dengan sistem. Perancangan arsitektur sistem yang membagi sistem menjadi 3 lapisan utama, yaitu *frontend*, *backend*, dan *database*. Pembuatan *mockup UI* untuk menentukan tampilan dan pengalaman pengguna dalam mengakses aplikasi.

b. Pengembangan *Frontend*

Frontend dikembangkan menggunakan React.js untuk menciptakan antarmuka yang interaktif dan responsif. Beberapa aspek utama yang diterapkan dalam *frontend* adalah, *state manajemen* menggunakan *useState* dan *useEffect* untuk mengelola data dalam komponen. *API handling* memanfaatkan Axios untuk mengambil dan mengirim data dari *frontend* ke *backend*. *Routing* menggunakan *React Router* untuk navigasi antar halaman aplikasi. *Form handling* validasi input sebelum data dikirim ke *backend* guna memastikan keakuratan data.

c. Pengembangan *Backend*

Backend dikembangkan menggunakan Node.js dengan framework Express.js sebagai *server API*. Beberapa fitur utama yang diimplementasikan yaitu, *manajemen database* menggunakan MongoDB dengan Mongoose sebagai ODM untuk mengelola data. Autentikasi dan Otorisasi menggunakan JWT (*JSON Web Token*) untuk memastikan keamanan sistem. *Hasing password* menggunakan *bcrypt* untuk mengenkripsi *password* sebelum disimpan ke *database*. Pembuatan *API endpoint* dibuat untuk menangani permintaan seperti registrasi pengguna, pengisian *tracer study*, analisis data, dan manajemen pengguna.

d. Integrasi *API* dan Pengujian

Tahap akhir dalam proses pengembangan adalah menghubungkan *frontend* dengan *backend* agar sistem dapat berfungsi dengan baik. Beberapa Langkah yang dilakukan adalah, konfigurasi *CORS* (*Cross-Origin Resource Sharing*) menggunakan *middleware CORS* agar *frontend* dapat berkomunikasi dengan *backend* secara aman. Pengujian *API* menggunakan *Postman* untuk memastikan setiap *endpoint* berjalan sesuai dengan kebutuhan. *Debugging* dan *Error Handling* menambahkan mekanisme log dan penanganan *error* untuk meningkatkan stabilitas sistem.

Dengan tahapan ini, sistem *Tracer Study LP3I* dapat berfungsi dengan baik, memiliki keamanan yang kuat, serta memberikan pengalaman yang optimal.

E. Pengujian Sistem

Pengujian sistem dilakukan menggunakan metode *Black Box Testing*, yaitu metode pengujian

perangkat lunak yang berfokus pada evaluasi fungsionalitas sistem tanpa memeriksa struktur *internal* atau kode sumbernya (Wijaya & Astuti, 2021). Pengujian ini bertujuan untuk memastikan bahwa setiap fungsi dalam aplikasi berjalan sesuai dengan tugasnya berdasarkan *input* yang diberikan dan *output* yang dihasilkan. Selain itu, pengujian ini juga memperhatikan aspek tampilan, fungsi-fungsi yang tersedia, serta kesesuaian alur kerja sistem dengan yang telah dirancang (Ningrum et al., 2019; Uminingsih et al., 2022). Metode ini digunakan untuk mendeteksi berbagai kecacatan dan kekurangan dalam perangkat lunak, seperti fungsionalitas yang salah atau hilang, kesalahan antarmuka, kesalahan dalam struktur data, akses ke *database external*, kesalahan performa, serta kesalahan dalam proses eksekusi atau terminisasi sistem (Hermawan et al., 2020; Permatasari et al., 2023).

Dalam pengujian ini, berbagai fitur utama diuji untuk melihat apakah sistem dapat merespon dengan benar terhadap setiap skenario penggunaan. Komponen atau fitur yang diuji serta hasil pengujiannya dapat dilihat pada Table 3.

Tabel 3. Komponen Uji Tracer Study LP3I

No	Komponen	Hasil yang Diharapkan	Hasil
1	Login	Pengguna dapat login ke aplikasi	Sesuai
2	Logout	Pengguna dapat keluar dari aplikasi	Sesuai
3	Dashboard	Dapat menampilkan ringkasan data analisis keselarasan horizontal dan vertikal	Sesuai
4	PDF Report	Menghasilkan laporan berbentuk PDF	Sesuai
5	Tambah Kegiatan Tracer	Admin dapat menambah kegiatan <i>tracer</i>	Sesuai
6	Edit Kegiatan Tracer	Admin dapat mengubah kegiatan <i>tracer</i>	Sesuai
7	Hapus Kegiatan Tracer	Admin dapat menghapus kegiatan <i>tracer</i>	Sesuai
8	Preview Kegiatan Tracer	Admin dapat melihat detail kegiatan	Sesuai
9	Tambah PSDku	Admin dapat menambah PSDku	Sesuai
10	Edit PSDku	Admin dapat mengubah data PSDku	Sesuai
11	Hapus PSDku	Admin dapat menghapus data	Sesuai

		PSDKu	
12	Tambah Program Studi	Admin dapat menambah program studi	Sesuai
13	Edit Program Studi	Admin dapat mengubah data program studi	Sesuai
14	Hapus Program Studi	Admin dapat menghapus data program studi	Sesuai
15	<i>Detail</i> Program Studi	Admin dapat melihat <i>detail</i> program studi	Sesuai
16	<i>Dashboard</i> Alumni	Alumni dapat melihat <i>list event</i> yang berlangsung	Sesuai
17	<i>Form</i> Kuesioner	Alumni dapat mengisi form kuesioner	Sesuai

V. KESIMPULAN DAN SARAN

Kesimpulan

Dengan adanya aplikasi *Tracer Study LP3I*, kampus dapat dengan cepat mengetahui tingkat keselarasan lulusan dengan dunia kerja, sehingga dapat dijadikan bahan evaluasi dalam perbaikan kurikulum. Selain itu, fitur-fitur yang tersedia memungkinkan pengelolaan data lebih sistematis dan mempermudah pihak kampus dalam mengambil keputusan berbasis data. Dari sisi pengguna, alumni juga mendapatkan manfaat dengan adanya sistem ini, karena mereka dapat mengisi *tracer study* dengan mudah dan memberikan *feedback* terhadap pendidikan yang telah mereka tempuh. Dengan demikian, aplikasi *Tracer Study LP3I* tidak hanya membantu kampus dalam proses evaluasi akademik, tetapi juga memberikan pengalaman yang lebih terstruktur bagi pada alumni dalam berkontribusi terhadap peningkatan kualitas pendidikan untuk kedepannya.

Saran

Berdasarkan hasil penelitian terdapat beberapa aspek yang dapat ditingkatkan dalam penelitian selanjutnya, aplikasi tracer study ini dapat dikembangkan dengan fitur analitik yang lebih mendalam, seperti penerapan *machine learning* atau *data mining* untuk memprediksi tren karir alumni dan memberikan rekomendasi personalisasi.

DAFTAR PUSTAKA

- Aziira, A. H., Kamil, H., & Kartika, A. D. (2023). Pembangunan Sistem Informasi Pengelolaan Rencana Pembelajaran Semester (RPS). *Jurnal Nasional Teknologi Dan Sistem Informasi*, 9(2), 103–112. <https://doi.org/10.25077/teknosi.v9i2.2023.103-112>
- Billy Indra Irawan, G. P. M. (2022). Perancangan User Interface dan User Experience Situs Web CreativePub dengan Metode User Centered Design. *Automata*, 3(2).
- Fadli, A., Zulfa, M. I., Widhi Nugraha, A. W., Taryana, A., & Aliim, M. S. (2020). Analisis Perbandingan Unjuk Kerja Database SQL dan Database NoSQL Untuk Mendukung Era Big Data. *Jurnal Nasional Teknik Elektro*, 9(3). <https://doi.org/10.25077/jnte.v9n3.774.2020>
- Fitriani, L., Setiawan, R., & Anwar, D. N. (2024). Tracer Study Berbasis Website dengan menggunakan Metodologi Agile Framework Scrum. *Jurnal Algoritma*, 21(1), 35–46. <https://doi.org/10.33364/algoritma/v.21-1.1401>
- Hermawan, L. C., Mubarak, M. R., Mairudin, H., Mahdiyan, A., & Yulianti, Y. (2020). Pengujian Black Box pada Aplikasi Verifikasi Data Nasabah dengan Menggunakan Metode Boundary Value Analysis. *Jurnal Teknologi Sistem Informasi Dan Aplikasi*, 3(3), 119. <https://doi.org/10.32493/jtsi.v3i3.5331>
- Khalif, M., & Wibowo, R. (2022). Analisis Diskusi Penggunaan Library React Js Dalam Pengembangan Web Di Stack Overflow. *Program Studi Informatika Fakultas Komunikasi Dan Informatika UMS*. https://eprints.ums.ac.id/102693/12/L200180_217_Muh_Khalif_Rizaldi_W_Teknik_Informatika_Naspub%28Final%29.pdf
- Mariko, S. (2019). Aplikasi website berbasis HTML dan JavaScript untuk menyelesaikan fungsi integral pada mata kuliah kalkulus. *Jurnal Inovasi Teknologi Pendidikan*, 6(1), 80–91. <https://doi.org/10.21831/jitp.v6i1.22280>
- Nasution. (2021). Implementasi Mongo Db, Express Js, React Js Dan Node Js (Mern) Pada Pengembangan Aplikasi Formulir, Kuis, Dan Survei Online. *Informatics Engineering*, 1–160. <https://dspace.uui.ac.id/handle/123456789/38607>
- Ningrum, F. C., Suherman, D., Aryanti, S., Prasetya, H. A., & Saifudin, A. (2019). Pengujian Black Box pada Aplikasi Sistem Seleksi Sales Terbaik Menggunakan Teknik Equivalence Partitions. *Jurnal Informatika Universitas Pamulang*, 4(4), 125. <https://doi.org/10.32493/informatika.v4i4.3782>
- Nursaid, F. F., Hendra Brata, A., & Kharisma, A. P. (2020). Pengembangan Sistem Informasi Pengelolaan Persediaan Barang Dengan ReactJS Dan React Native Menggunakan Prototype (Studi Kasus : Toko Uda Fajri). *J-Ptiik.Ub.Ac.Id*, 4(1), 46–55. <http://j-ptiik.ub.ac.id>

- Nursyanti, R., Alamsyah, R. Y. R., & Perdana, S. (2019). Perancangan Aplikasi Berbasis Web Untuk Membantu Pengujian Kualitas Kain Tekstil Otomotif. *Explore - Jurnal Manajemen Sistem Informasi Dan Telematika*, 10(2), 153–159.
- Panjaitan, J., & Pakpahan, A. F. (2021). Perancangan Sistem E-Reporting Menggunakan ReactJS dan Firebase. *Jurnal Teknik Informatika Dan Sistem Informasi*, 7(1), 20–34. <https://doi.org/10.28932/jutisi.v7i1.3098>
- Permatasari, I., Adhania, F., Putri, S. A., & Nursari, S. R. C. (2023). Pengujian Black Box Menggunakan Metode Analisis Nilai Batas pada Aplikasi DANA. *KONSTELASI: Konvergensi Teknologi Dan Sistem Informasi*, 3(2), 373–387. <https://doi.org/10.24002/konstelasi.v3i2.8289>
- Puspa Putri, D. A. (2019). Rancang Bangun Media Pembelajaran Bahasa Arab Untuk Anak Usia Dini Berbasis Android. *Technologia: Jurnal Ilmiah*, 10(3), 156. <https://doi.org/10.31602/tji.v10i3.2230>
- Putra, D. J. K., & Tanaem, P. F. (2022). Perancangan Aplikasi Pembukuan Menggunakan Metode Agile Scrum. *Jurnal Teknik Informatika Dan Sistem Informasi*, 8(3), 509–521. <https://doi.org/10.28932/jutisi.v8i3.5060>
- Susanti, E., Sholeh, M., Yuliana, R., Sains, I., & Yogyakarta, T. A. (2020). Konversi Data Dari Database Relasional Mysql Ke Database Nosql Mongodb (Studi Kasus Pada Sistem Informasi Umkm). *Jurnal Gaung ...*, 13(July), 87–96. <http://www.jurnal.usahidsolo.ac.id/index.php/GI/article/view/520>
- Susanto, E. S., Aprianti, H., & Sejahtera, M. (2020). Sistem Informasi Pendaftaran Siswa Baru Pada Smkn 2 Sumbawa Besar Berbasis Web. *Jurnal Informatika, Teknologi Dan Sains*, 2(2), 129–135. <https://doi.org/10.51401/jinteks.v2i2.594>
- Uminingsih, Nur Ichsanudin, M., Yusuf, M., & Suraya, S. (2022). Pengujian Fungsional Perangkat Lunak Sistem Informasi Perpustakaan Dengan Metode Black Box Testing Bagi Pemula. *STORAGE: Jurnal Ilmiah Teknik Dan Ilmu Komputer*, 1(2), 1–8. <https://doi.org/10.55123/storage.v1i2.270>
- Wijaya, Y. D., & Astuti, M. W. (2021). Pengujian Blackbox Sistem Informasi Penilaian Kinerja Karyawan Pt Inka (Persero) Berbasis Equivalence Partitions. *Jurnal Digital Teknologi Informasi*, 4(1), 22. <https://doi.org/10.32502/digital.v4i1.3163>