

Perancangan Dan Implementasi *Back-End* Dalam Pengembangan Bagian Admin Dan *Authentication* Pada Aplikasi Berbasis Web *NIUNIQ* Menggunakan Metode *Iterative Incremental*

1st Leonard Lesmana
Sistem Informasi
Telkom University
Bandung, Indonesia

leonardlesmana@student.telkomuniversity.ac.id

2nd Ir. Ari Fajar Santoso, M.T.
Sistem Informasi
Telkom University
Bandung, Indonesia

arifajar@telkomuniversity.ac.id

3rd Taufik Nur Adi, S.Kom., M.T.,
Ph.D.

Sistem Informasi
Telkom University
Bandung, Indonesia

taufikna@telkomuniversity.ac.id

Abstrak— *Usaha Mikro, Kecil, dan Menengah (UMKM) merupakan jenis usaha produktif yang dijalankan oleh individu maupun badan usaha yang memenuhi syarat sebagai usaha mikro. Di Indonesia, UMKM berkembang di berbagai sektor, salah satunya yang menarik perhatian adalah sektor kerajinan tangan (handicraft). Produk handicraft adalah hasil kerajinan yang dapat dimanfaatkan untuk kegiatan sehari-hari maupun dinikmati dari sisi nilai estetikanya. Banyak masyarakat Indonesia memiliki keahlian dalam bidang ini dan mampu menghasilkan karya seni yang mendapat pengakuan baik di tingkat nasional maupun internasional. Namun demikian, tidak sedikit pula yang melakukan plagiarisme terhadap karya orang lain, lalu mengklaimnya sebagai hasil ciptaannya sendiri. Praktik ini sangat merugikan karena dapat mengurangi kepercayaan konsumen dan menghilangkan nilai orisinalitas produk tersebut. Untuk mengatasi masalah tersebut, dikembangkanlah Niuniq, sebuah aplikasi berbasis web yang bertujuan untuk memverifikasi keaslian produk UMKM. Aplikasi ini dibangun dengan pendekatan metode iterative incremental, yaitu model pengembangan yang memungkinkan proses revisi pada tahap-tahap sebelumnya jika ditemukan kesalahan. Setelah sistem dikembangkan, dilakukan pengujian fungsional menggunakan metode black-box testing untuk mengevaluasi berbagai layanan dalam sistem tersebut. Hasil dari pengujian menunjukkan penilaian terhadap performa sistem layanan yang telah dirancang. Tugas akhir ini berhasil menghasilkan sebuah REST API yang nantinya akan menjadi komponen penting dalam pengembangan aplikasi Niuniq sebagai alat verifikasi keaslian produk UMKM.*

Kata kunci— *UMKM, kerajinan tangan, REST API, aplikasi web, iterative incremental*

I. PENDAHULUAN

Transformasi digital yang didorong oleh Revolusi Industri 4.0 telah mengubah ekonomi Indonesia secara signifikan, dengan UMKM sebagai pilar utama pertumbuhan dan ketahanan ekonomi. UMKM mencakup sekitar 66 juta unit usaha, menyumbang 61% PDB, dan menyerap 97% tenaga kerja. Namun, tantangan muncul dalam bentuk pelanggaran hak kekayaan intelektual dan maraknya produk palsu di ranah digital, yang menyebabkan kerugian hingga Rp291 triliun dan menurunkan kepercayaan konsumen, terutama pada sektor kerajinan tangan. Untuk mengatasinya, dikembangkan aplikasi Niuniq sebagai solusi verifikasi

keaslian produk berbasis web dengan QR Code. Aplikasi ini dibangun menggunakan metode Iterative Incremental dan teknologi seperti Node.js, Express.js, serta MongoDB, dan diuji dengan black-box testing. Hasilnya, sistem ini diharapkan mampu meningkatkan kepercayaan konsumen, melindungi UMKM, dan memperkuat ekosistem digital yang aman di Indonesia.

II. KAJIAN TEORI

A. Usaha Mikro, Kecil dan Menengah (UMKM)

Menurut Undang-Undang No. 20 Tahun 2008, kriteria UMKM dibedakan berdasarkan aset dan omzet tahunan. Usaha Mikro memiliki aset maksimal Rp 50 juta dan omzet maksimal Rp 300 juta. Usaha Kecil memiliki aset lebih dari Rp 50 juta hingga Rp 500 juta dan omzet lebih dari Rp 300 juta hingga Rp 2,5 miliar. Usaha Menengah memiliki aset lebih dari Rp 500 juta hingga Rp 10 miliar dan omzet lebih dari Rp 2,5 miliar hingga Rp 50 miliar.

B. Aplikasi Web dan Arsitektur REST API

Sistem perangkat lunak yang dirancang untuk diakses oleh pengguna melalui *web browser* dalam arsitektur *client-server*. Untuk memungkinkan komunikasi antara *client* (antarmuka pengguna) dan *server (back-end)*, digunakan sebuah antarmuka yang disebut API (*Application Programming Interface*). REST API adalah gaya arsitektur yang paling populer untuk membangun API ini, yang memanfaatkan protokol standar web (HTTP) untuk menciptakan komunikasi yang stateless, efisien, dan skalabel.

C. Framework

Dalam pengembangan perangkat lunak modern, aplikasi besar memerlukan fondasi yang kuat dan terstruktur, yang sering diwujudkan melalui kerangka kerja (*framework*). *Framework* bukan sekadar pustaka, tetapi memberikan cetak biru arsitektural yang mengatur struktur dan alur kerja aplikasi. Bagian ini membahas konsep dasar *framework*, prinsip arsitekturnya, serta jenis-jenis desain yang membedakannya. Secara umum, *framework* adalah struktur konseptual untuk menyelesaikan masalah kompleks dalam pengembangan perangkat lunak.

D. Node.js dan Express.js

Node.js secara resmi didefinisikan sebagai sebuah lingkungan eksekusi runtime *JavaScript* yang bersifat *open-source* dan lintas platform. Tujuan utamanya adalah untuk memungkinkan eksekusi kode *JavaScript* di luar lingkungan peramban web (*web browser*), yang secara historis merupakan satu-satunya tempat di mana *JavaScript* dapat berjalan. Express.js adalah sebuah kerangka kerja aplikasi web yang cepat, minimalis, dan fleksibel, yang dirancang dan dibangun secara spesifik untuk Node.js

E. Software Development Life Cycle (SDLC)

Proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya.

F. Iterative Incremental

Metode pengembangan perangkat lunak yang modelnya berbentuk secara bertahap dan berulang-ulang. Pengembangan *Iterative* dan *Incremental* adalah kombinasi dari desain iteratif atau metode iteratif dan model *build incremental* untuk pengembangan. Proses ini memungkinkan pengembangan kembali ke bagian awal jika terjadi kesalahan.

G. Unified Modelling Language (UML)

UML adalah bahasa pemodelan visual standar yang berperan sebagai cetak biru dalam rekayasa perangkat lunak. UML membantu memvisualisasikan, merinci, membangun, dan mendokumentasikan sistem berorientasi objek, sehingga mempermudah komunikasi antar pengembang dan pemangku kepentingan.

H. Entity Relationship Diagram (ERD)

Sebuah alat pemodelan data konseptual yang digunakan untuk merancang struktur logis sebuah database. ERD secara visual menggambarkan entitas-entitas utama dalam suatu sistem, atribut-atribut yang mendeskripsikannya, serta hubungan (relasi) yang ada di antara entitas-entitas tersebut, berfungsi sebagai fondasi sebelum implementasi fisik *database*.

I. Blackbox Testing

Software testing adalah suatu tahapan di mana pengujian dilakukan pada beberapa aspek dalam suatu aplikasi seperti kualitas, fitur/fungsi, keamanan, serta kinerja agar memenuhi syarat dan kebutuhan pengguna. *Black-box testing* adalah sebuah metode validasi perangkat lunak di mana fungsionalitas sistem diuji tanpa melihat struktur kode internal, implementasi, atau alur kerja internalnya. Pengujian ini berfokus sepenuhnya pada analisis input dan output untuk memverifikasi bahwa sistem berperilaku sesuai dengan spesifikasi dan persyaratan yang telah ditentukan.

III. METODE

Model konseptual menggambarkan hubungan antar konsep secara rinci dan menjadi rancangan awal sistem. Dalam penelitian ini, identifikasi masalah pada bisnis UMKM menghasilkan kebutuhan akan aplikasi Niuniq berbasis web untuk memverifikasi produk kerajinan tangan.

Analisis kebutuhan dilakukan agar solusi yang dikembangkan relevan dengan kondisi di lapangan.

A. Sistematika Penyelesaian Masalah

Langkah atau tahapan-tahapan yang akan dilakukan dalam penelitian untuk mendapatkan suatu solusi dari masalah yang dialami. Dalam sistematika penyelesaian masalah pada penelitian ini, metode yang digunakan pada penelitian ini adalah metode *Iterative Incremental*. Peneliti menggunakan metode *Iterative Incremental* karena metode ini memungkinkan adanya pengulangan kembali pada tahap sebelumnya apabila terdapat suatu perubahan konsep atau tahap tertentu tidak mencapai target.

B. Metode Pengumpulan Data

Untuk memperoleh data yang akurat dan relevan, digunakan metode kualitatif agar peneliti memahami konteks dan kebutuhan para pemangku kepentingan. Teknik pengumpulan data meliputi: (1) Studi Literatur untuk membangun dasar teori dari berbagai sumber terkait UMKM, digitalisasi, dan verifikasi produk; (2) Observasi langsung di UMKM "Roemah Bamboe" guna memahami proses bisnis dan isu plagiarisme; (3) Wawancara dengan pihak UMKM untuk menggali kebutuhan fitur aplikasi yang mampu melindungi karya dan meningkatkan kepercayaan konsumen; serta (4) Dokumentasi berupa foto produk, catatan produksi, dan data pendukung perancangan sistem.

C. Alasan Pemilihan Metode

Pemilihan metode *Iterative Incremental* didasarkan pada beberapa keunggulan yang sangat sesuai dengan karakteristik proyek pengembangan aplikasi Niuniq. Berbeda dengan metode tradisional seperti *Waterfall* yang bersifat kaku dan sekuensial, *Iterative Incremental* menawarkan pendekatan yang lebih fleksibel dan adaptif, seperti yang dapat dilihat dalam Tabel 1 (A) berikut ini:

TABEL 1
(A)

Kriteria	Waterfall	Iterative Incremental
Fleksibilitas	Rendah, perubahan sulit diakomodasi.	Sedang hingga Tinggi, fleksibel dalam setiap iterasi.
Keterlibatan Klien	Rendah, hanya di awal (fase analisis) dan akhir.	Tinggi, klien memberikan feedback di akhir setiap iterasi.
Penanganan Kebutuhan	Kebutuhan harus didefinisikan lengkap di awal.	Kebutuhan awal didefinisikan, namun dapat disempurnakan.
Kecepatan Pengiriman	Lambat, produk jadi dikirim di akhir proyek.	Cepat, versi fungsional dikirim setiap iterasi/increment.

Kriteria	Waterfall	Iterative Incremental
Visibilitas Proyek	Rendah, progres sulit terlihat hingga tahap akhir.	Tinggi, progres terlihat setelah setiap penambahan (increment).
Cocok Untuk	Proyek dengan kebutuhan yang pasti dan tidak berubah.	Proyek di mana fungsionalitas dapat dikirim secara bertahap.

D. Tahap Pengembangan

Tahap pengembangan sistem adalah inti penelitian ini dan menggunakan metode Iterative Incremental, yang membagi proses menjadi beberapa iterasi. Setiap iterasi menghasilkan bagian perangkat lunak yang bisa diuji. Tahapannya meliputi:

1. Perencanaan (*Planning*), yaitu analisis kebutuhan untuk *increment* tertentu;
2. Perancangan (*Design*), mencakup desain arsitektur *back-end*, *database* (ERD), dan diagram UML;
3. Pengembangan (*Coding*), yaitu pengkodean modul Admin dan *Authentication* menggunakan Node.js, Express.js, dan MongoDB;
4. Pengujian (*Testing*), dengan *Black-Box Testing* untuk memvalidasi fungsi endpoint REST API; dan
5. Evaluasi (*Evaluation*), yaitu meninjau hasil pengujian sebagai dasar iterasi selanjutnya.

IV. HASIL DAN PEMBAHASAN

Bab ini membahas analisis dan perancangan sistem untuk menyelesaikan masalah sebelumnya. Fokusnya pada arsitektur *back-end* modul Admin dan *Authentication*, dimulai dari analisis proses bisnis hingga perumusan kebutuhan fungsional. Hasilnya digunakan untuk merancang struktur data, alur sistem, arsitektur aplikasi, dan spesifikasi API, dengan pendekatan Iterative Incremental serta bantuan UML dan ERD.

A. Analisis Gap

Analisis gap dilakukan untuk mengidentifikasi kesenjangan antara kondisi proses bisnis eksisting dengan kebutuhan ideal guna menciptakan ekosistem digital yang aman dan terpercaya bagi UMKM kerajinan tangan. Analisis ini secara spesifik menyoroti ketiadaan fungsi-fungsi fundamental terkait autentikasi, otorisasi, dan administrasi yang menjadi inti dari penelitian ini, seperti yang dapat dilihat dalam Tabel 2 (A).

TABEL 2 (A)

Referensi	Fitur A: Back-end REST API	Fitur B: Konteks UMKM	Fitur C: Verifikasi Produk	Fitur D: Metode Iterative	Status & Celah Penelitian
Riady et al. (2024)	F	N	N	N	Fokus pada teknis REST API tanpa konteks UMKM/verifikasi.
Prabawati et al. (2024)	F	F	N	P	Membangun sistem informasi untuk UMKM, namun bukan verifikasi.
Shiddieq et al. (2023)	P	F	N	N	Menggunakan GIS untuk pemetaan, bukan verifikasi produk.
Pratama et al. (2021)	F	N	P	F	Fokus pada QR Code untuk smart packaging, bukan untuk UMKM.
Penelitian Ini	F	F	F	F	Mengisi celah dengan mengintegrasikan ke-empat fitur: membangun back-end REST API untuk verifikasi produk UMKM menggunakan metode Iterative.

Keterangan:

- N: Not Fulfilled (Kebutuhan sama sekali tidak terpenuhi)
- P: Partially Fulfilled (Kebutuhan sebagian terpenuhi)
- F: Fully Fulfilled (Kebutuhan sudah terpenuhi sepenuhnya)

Dari analisis tabel di atas, terlihat jelas bahwa kesenjangan utama terletak pada ketiadaan infrastruktur tata kelola (*governance*) dan keamanan digital. Solusi yang diusulkan tidak hanya sekadar menyediakan fitur, tetapi membangun fondasi sistem yang aman, terkelola, dan terpercaya melalui implementasi modul *Authentication* dan Admin yang solid.

B. Analisis Aktor

Analisis aktor bertujuan untuk mengidentifikasi dan mendefinisikan entitas, baik manusia maupun sistem eksternal, yang akan berinteraksi dengan aplikasi Niuniq. Identifikasi ini didasarkan pada proses bisnis yang telah dianalisis dan dimodelkan dalam Use Case Diagram. Setiap aktor memiliki peran, tanggung jawab, dan hak akses yang berbeda-beda, yang krusial dalam perancangan sistem, terutama untuk implementasi keamanan dan otorisasi, seperti yang dapat dilihat pada Tabel 3 (A).

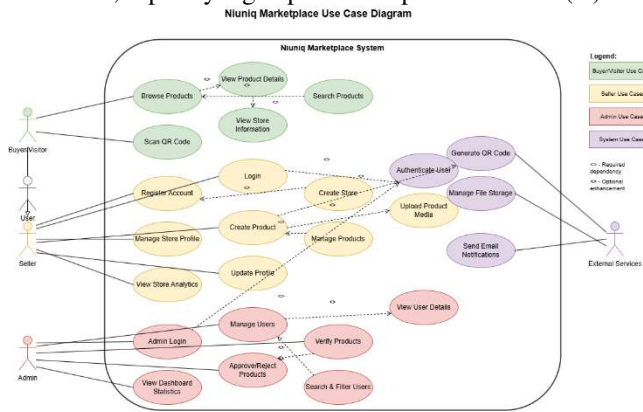
TABEL 3 (A)

No	Aktor	Peran dan Tanggung Jawab	Hak Akses (CRUD)
1	Admin	Admin adalah aktor utama dengan akses tertinggi. Tugasnya meliputi login ke portal admin, mengelola pengguna, memverifikasi produk, dan memantau sistem melalui	Product: R, U (verifikasi/tolak). User: C, R, U, D. Store: R, U, D.

No	Aktor	Peran dan Tanggung Jawab	Hak Akses (CRUD)
		dashboard demi menjaga keamanan dan kinerja.	
2	Seller (Penjual)	Aktor ini adalah penjual yang telah terdaftar dan mengelola tokonya. Tugasnya mencakup mengatur profil, mengelola produk, mengajukan verifikasi, mengunduh QR code terverifikasi, dan memantau analitik toko.	User: R, U (profil sendiri). Store: C, R, U (toko sendiri). Product: C, R (produk sendiri).
3	User (Pengguna)	Aktor ini adalah pengguna terdaftar tanpa toko yang berfungsi sebagai identitas awal dalam sistem. Tugasnya mencakup mendaftar, login, dan memperbarui profil, serta bisa menjadi 'Seller' setelah membuat toko.	User: R, U (profil sendiri).
4	Buyer/Visitor (Pembeli/Pengunjung)	Aktor ini adalah pengguna umum yang belum login, hanya bisa melihat. Ia dapat mencari produk, melihat detail toko dan produk, serta memindai QR code untuk membuka halaman verifikasi dan riwayat produk.	User: C (registrasi). Product: R (terbatas).
5	External Services (Layanan Eksternal)	Aktor ini bukan pengguna manusia, melainkan sistem atau layanan pihak ketiga yang terhubung dengan backend Niuniq, seperti layanan email (SMTP), penyimpanan file (CDN), dan generator QR code.	-

C. Use Case Diagram

Berfungsi untuk memvisualisasikan fungsionalitas sistem dari perspektif pengguna (end-user). Diagram ini menggambarkan interaksi antara aktor dengan fitur-fitur utama (use case) yang disediakan oleh sistem, serta hubungan antar use case itu sendiri. Diagram ini menjadi cetak biru fungsional yang merangkum seluruh kapabilitas sistem yang diusulkan, seperti yang dapat dilihat pada Gambar 1(A)



GAMBAR 1 (A)

D. Use Case Scenario

Memberikan deskripsi tekstual yang terperinci untuk setiap use case yang telah diidentifikasi pada diagram sebelumnya. Skenario ini menguraikan langkah-langkah interaksi antara aktor dan sistem secara sekuensial, mencakup kondisi awal (*pre-condition*), kondisi akhir (*post-condition*), alur normal (*main flow*), dan alur alternatif (*alternative flow*). Tujuannya adalah untuk memperjelas bagaimana sebuah fungsionalitas berjalan dari awal hingga akhir, yang sangat berguna bagi tim pengembang dalam memahami logika bisnis, seperti yang dapat dilihat pada Tabel 4 (A) Use Case Scenario: Admin, Tabel 4 (B) Use Case Scenario: Approve/Reject Products, Tabel 4 (C) Use Case Scenario:

User Register Account, dan Tabel 4 (D) Use Case Scenario: Seller Creates Product.

TABEL 4 (A)

Atribut	Deskripsi
Nama Use Case	Admin Login
ID Use Case	UC-ADM-01
Aktor	Admin
Trigger	Admin ingin mengakses dashboard administratif untuk mengelola sistem.
Pre-Condition	Admin berada di halaman login aplikasi dan memiliki kredensial (email dan password) yang valid dengan peran 'admin'.
Post-Condition	Admin berhasil di autentikasi dan diarahkan ke halaman utama dashboard admin. Sebuah sesi aman (menggunakan JWT) telah dibuat.
Alur Utama	1. Admin membuka halaman login. 2. Memasukkan email dan password. 3. Menekan tombol "Login". 4. Sistem mengirim kredensial ke backend. 5. Backend memvalidasi data dengan database. 6. Backend memeriksa apakah peran pengguna adalah 'admin'. 7. Jika valid, backend membuat dan mengirim token (JWT). 8. Frontend menyimpan token dan mengarahkan ke dashboard.
Alur Alternatif	6a. Jika kredensial salah, sistem menampilkan pesan "Email atau password salah". 6b. Jika peran bukan 'admin', sistem menampilkan pesan "Akses ditolak. Anda tidak memiliki hak akses admin."

TABEL 4 (B)

Atribut	Deskripsi
Nama Use Case	Approve/Reject Products
ID Use Case	UC-ADM-02
Aktor	Admin
Trigger	Admin ingin meninjau dan membuat keputusan terhadap produk yang diajukan oleh Seller.
Pre-Condition	Admin telah berhasil login dan berada di halaman dashboard admin, khususnya pada daftar produk yang berstatus 'pending'.
Post-Condition	Status produk yang dipilih berubah menjadi 'verified' (jika disetujui) atau 'rejected' (jika ditolak). Notifikasi dikirimkan kepada Seller terkait.
Alur Utama	1. Admin memilih produk dari daftar 'pending'. 2. Sistem menampilkan detail produk yang dikirim Seller. 3. Admin mengevaluasi konten. 4. Admin memilih "Approve" atau "Reject". 5. Jika "Approve": 5a. Status produk diubah jadi 'verified'. 5b. Sistem membuat QR code unik. 5c. Notifikasi email dikirim ke Seller. 6. Jika "Reject": 6a. Sistem menampilkan kolom alasan penolakan. 6b. Admin mengisi alasan dan menekan "Confirm Reject". 6c. Status diubah jadi 'rejected' dan alasan disimpan. 6d. Sistem mengirim email penolakan ke Seller.
Alur Alternatif	4a. Admin membatalkan aksi dan kembali ke daftar produk 'pending'.

TABEL 4
(C)

Atribut	Deskripsi
Nama Use Case	Register Account
ID Use Case	UC-USR-01
Aktor	User (calon pengguna)
Trigger	Pengguna baru ingin membuat akun di aplikasi Niuniq.
Pre-Condition	Pengguna berada di halaman pendaftaran aplikasi.
Post-Condition	Akun pengguna baru berhasil dibuat dan disimpan di database. Pengguna diarahkan ke halaman login atau langsung login.
Alur Utama	1. Pengguna membuka halaman pendaftaran. 2. Mengisi formulir (email, password, konfirmasi). 3. Menekan tombol "Register". 4. Sistem memvalidasi input di frontend. 5. Data dikirim ke backend. 6. Backend mengecek apakah email sudah terdaftar. 7. Jika belum, password di-hash dan data disimpan dengan peran 'User'. 8. Sistem menampilkan pesan sukses dan mengarahkan ke halaman login.
Alur Alternatif	6a. Jika email sudah terdaftar, sistem menampilkan pesan error "Email sudah digunakan oleh akun lain."

TABEL 4
(D)

Atribut	Deskripsi
Nama Use Case	Create Product
ID Use Case	UC-SEL-01
Aktor	Seller
Trigger	Seller ingin mendaftarkan produk baru untuk dijual dan diverifikasi keasliannya.
Pre-Condition	Seller telah berhasil login dan memiliki toko yang sudah terdaftar. Seller berada di dashboard tokonya.
Post-Condition	Data produk baru disimpan di database dengan status 'pending'. Produk masuk ke dalam antrian verifikasi Admin.
Alur Utama	1. Seller menekan "Tambah Produk Baru". 2. Sistem menampilkan formulir produk. 3. Seller mengisi data lengkap (nama, deskripsi, harga, media, dll.). 4. Menekan "Simpan" atau "Ajukan untuk Verifikasi". 5. Sistem memvalidasi input. 6. Data dikirim ke backend. 7. Backend menyimpan produk dengan status isVerification 'pending'. 8. Sistem menampilkan pesan bahwa produk sedang menunggu verifikasi Admin.
Alur Alternatif	5a. Jika validasi input gagal (misal: ada kolom wajib yang kosong), sistem menampilkan pesan error dan menyorot kolom yang bermasalah.

E. Analisis Kebutuhan Fitur

Dari hasil analisis proses bisnis, aktor, dan use case, dapat disintesis serangkaian kebutuhan fitur tingkat tinggi (*high-level features*). Fitur-fitur ini merangkum kapabilitas utama yang harus dimiliki oleh aplikasi Niuniq untuk dapat memenuhi tujuannya, dengan penekanan pada modul Admin dan Authentication.

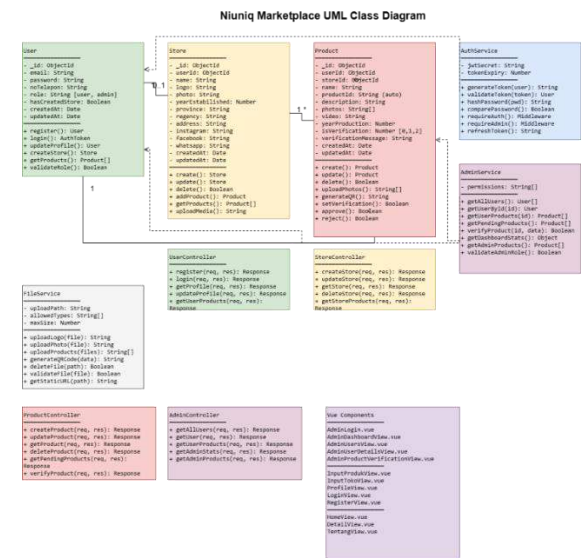
F. Analisis Kebutuhan Fungsional Sistem

Bertujuan merinci fitur menjadi persyaratan yang spesifik, terukur, dapat diuji, dan jelas. Kebutuhan fungsional menjelaskan perilaku sistem terhadap input atau kondisi tertentu. Daftar ini menjadi acuan teknis bagi tim pengembang *backend*, seperti ditunjukkan pada Tabel 6(A).

TABEL 6
(A)

ID Kebutuhan	Nama Kebutuhan	Deskripsi Fungsional	Aktor Terkait
Modul Autentikasi (F-AUTH)			
F-AUTH-01	Pendaftaran Akun Pengguna	Sistem harus menerima data pendaftaran (<i>email</i> dan <i>password</i>), melakukan hashing pada <i>password</i> , lalu menyimpannya ke koleksi <i>Users</i> di database dengan peran default sebagai 'User'.	User
F-AUTH-02	Login Pengguna	Sistem harus memvalidasi <i>email</i> dan <i>password</i> pengguna dengan data di database. Jika valid, sistem akan menghasilkan dan mengembalikan <i>JSON Web Token (JWT)</i> berisi ID, peran pengguna, dan waktu kedaluwarsa.	User, Seller, Admin
F-AUTH-03	Proteksi Rute Berbasis Peran	Sistem harus menggunakan <i>middleware</i> untuk mengamankan endpoint tertentu. Akses diberikan hanya jika <i>JWT</i> dalam request header valid dan perannya sesuai, misalnya hanya 'Admin' yang boleh mengakses endpoint <code>`/api/admin/*`</code> .	User, Seller, Admin
Modul Admin (F-ADM)			
F-ADM-01	Login Khusus Admin	Saat login, selain memvalidasi kredensial, sistem harus memastikan peran pengguna adalah 'admin'. Jika bukan, akses ditolak meskipun email dan password valid.	Admin
F-ADM-02	Mengambil Daftar Pengguna	Sistem harus menyediakan <i>endpoint</i> API yang memungkinkan Admin untuk mengambil daftar semua pengguna yang terdaftar di sistem, dengan opsi untuk melakukan paginasi, pencarian, dan pemfilteran.	Admin
F-ADM-03	Melihat Detail Pengguna	Sistem harus menyediakan <i>endpoint</i> API yang memungkinkan Admin untuk mengambil data detail dari satu pengguna spesifik berdasarkan ID pengguna.	Admin
F-ADM-04	Mengambil Daftar Produk Pending	Sistem harus menyediakan <i>endpoint</i> API yang mengembalikan daftar semua produk yang saat ini memiliki status 'pending' untuk ditinjau oleh Admin.	Admin
F-ADM-05	Menyetujui Produk	Sistem harus menyediakan <i>endpoint</i> API yang memungkinkan Admin untuk mengubah status sebuah produk (berdasarkan ID produk) dari 'pending' menjadi 'verified'. Aksi ini juga harus memicu pembuatan QR code.	Admin
F-ADM-06	Menolak Produk	Sistem harus menyediakan <i>endpoint</i> API yang memungkinkan Admin untuk mengubah status sebuah produk (berdasarkan ID produk) dari 'pending' menjadi 'rejected' dan	Admin

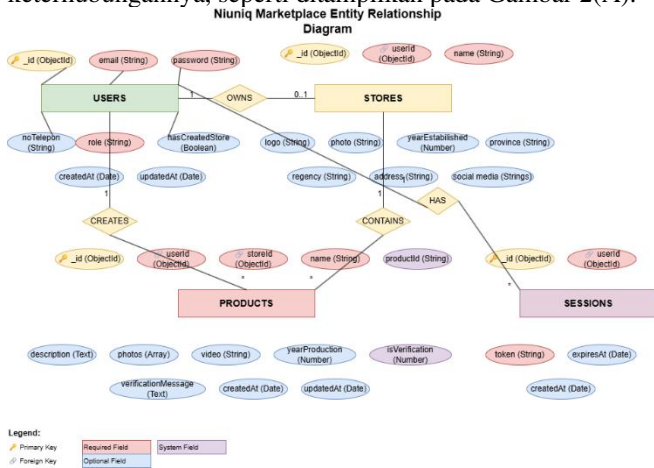
ID Kebutuhan	Nama Kebutuhan	Deskripsi Fungsional	Aktor Terkait
		menyimpan verificationMessage (alasan penolakan) yang diberikan.	
Modul Penjual (F-SEL)			
F-SEL-01	Membuat Toko	Sistem harus memungkinkan pengguna dengan peran 'User' untuk membuat entitas toko baru yang terhubung dengan ID pengguna mereka. Setelah berhasil, peran pengguna harus diperbarui menjadi 'Seller'.	User
F-SEL-02	Membuat Produk	Sistem harus memungkinkan pengguna dengan peran 'Seller' untuk membuat entitas produk baru. Produk yang baru dibuat harus secara otomatis memiliki status 'pending'.	Seller



GAMBAR 3 (A)

G. Entity Relationship Diagram (ERD)

Digunakan untuk memodelkan struktur logis data dalam sistem, ERD menggambarkan entitas utama, atribut, dan relasinya. Meski awalnya untuk database relasional, prinsip ERD tetap relevan dalam merancang skema database NoSQL seperti MongoDB, khususnya dalam menentukan entitas dan keterhubungannya, seperti ditampilkan pada Gambar 2(A).



GAMBAR 2 (A)

H. Class Diagram

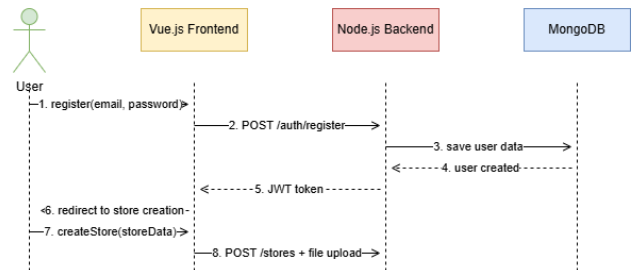
Menyediakan pandangan statis dari struktur sistem dengan menunjukkan kelas-kelas, atribut-atributnya, metode-metodenya, dan hubungan antar kelas. Diagram ini berfungsi sebagai cetak biru untuk pengkodean berorientasi objek dan membantu dalam memahami bagaimana berbagai bagian dari kode saling terkait, seperti yang dapat dilihat pada Gambar 3 (A)

I. Sequence Diagram

Diagram ini membantu merinci alur pesan dan urutan fungsi antara frontend, backend, dan database. Tiga alur utama dalam sistem Niuniq digambarkan pada Gambar 4, yaitu: (A) Pendaftaran Pengguna dan Pembuatan Toko, (B) Pembuatan Produk dan Verifikasi oleh Admin, dan (C) Manajemen Pengguna oleh Admin.

Niuniq Marketplace - Key User Flows Sequence Diagrams

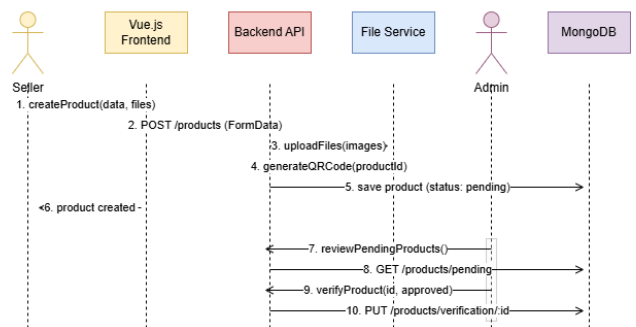
1. User Registration & Store Creation Flow



GAMBAR 4 (A)

Niuniq Marketplace - Key User Flows Sequence Diagrams

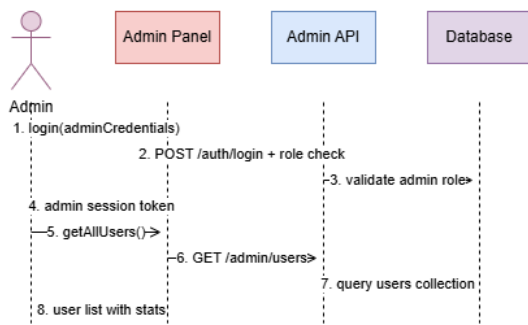
2. Product Creation & Admin Verification Flow



GAMBAR 4 (B)

Niuniq Marketplace - Key User Flows Sequence Diagrams

3. Admin User Management Flow

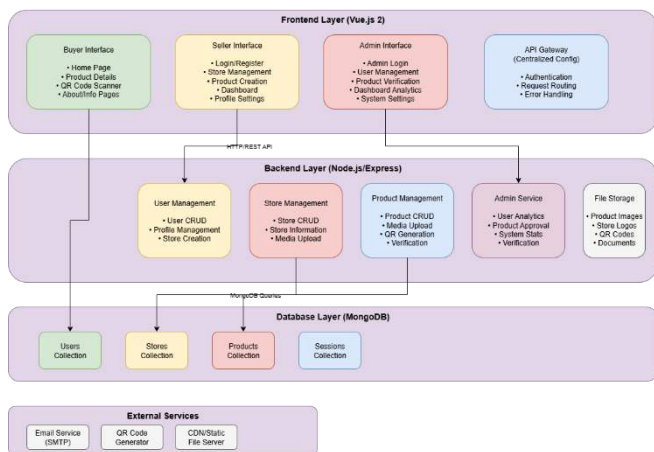


GAMBAR 4 (C)

J. Arsitektur Aplikasi

Arsitektur aplikasi memberikan gambaran tingkat tinggi (*high-level overview*) tentang bagaimana seluruh komponen sistem diorganisir dan berinteraksi satu sama lain. Ini mencakup lapisan-lapisan logis dan fisik dari aplikasi, dari antarmuka pengguna hingga penyimpanan data, Gambar 5 (A) menyajikan diagram alur sistem (*system flow*) yang menggambarkan arsitektur aplikasi Niuniq.

Niuniq Marketplace System Flow



GAMBAR 5 (A)

K. Perancangan REST API

Tahap krusial yang mendefinisikan "kontrak" antara frontend dan backend. Ini melibatkan pendefinisian endpoints (URL), metode HTTP yang diizinkan, serta format data yang diharapkan untuk permintaan dan respons. API yang dirancang dengan baik harus mengikuti prinsip-prinsip REST, seperti antarmuka yang seragam, statelessness, dan pemisahan client-server.

L. Iterasi Inkremental Fase Pertama: Modul Autentikasi

Pada fase pertama dari siklus *Iterative Incremental*, pengembangan difokuskan pada pembangunan fondasi keamanan sistem, yaitu modul Authentication. Modul ini krusial karena menjadi gerbang utama untuk semua interaksi pengguna di dalam aplikasi dan menjadi dasar bagi implementasi kontrol akses berbasis peran (*Role-Based Access Control - RBAC*). Pengembangan pada fase ini menggunakan *framework* Express.js, dan pengujian

fungsionalitas REST API dilakukan melalui metode *black-box* testing. Tahap-tahap pada iterasi pertama dapat dijelaskan pada berikut ini:

1. Tahap Perencanaan (*Planning*)

Pada tahap perencanaan fase pertama, fokus utama adalah merancang dan menjadwalkan pengembangan *endpoint* API yang esensial untuk fungsionalitas autentikasi dan otorisasi pengguna. Rencana ini mencakup pendaftaran pengguna baru, proses login, dan mekanisme penerbitan token untuk manajemen sesi.

2. Tahap Perancangan (*Design*)

Perancangan modul Autentikasi dilakukan setelah tahap perencanaan selesai, menghasilkan artefak seperti Use Case Diagram, ERD, Sequence Diagram, arsitektur aplikasi, dan spesifikasi REST API. Seluruh diagram dan rancangan teknis disajikan pada poin H hingga K.

3. Tahap Pengembangan (*Development*)

Pengembangan backend dilakukan menggunakan *framework* Express.js, dan pengujian awal setiap *endpoint* dilakukan menggunakan aplikasi Postman untuk memverifikasi respons API.

4. Tahap Pengujian (*Testing*)

Setelah pengembangan selesai, dilakukan pengujian *black-box* testing untuk memastikan fitur pada modul Autentikasi berfungsi sesuai kebutuhan, tanpa melihat kode internal. Pengujian dilakukan lewat Postman dan dirangkum dalam Tabel 9 (A).

TABEL 9 (A)

Deskripsi Pengujian	Data Masukan	Hasil yang Diharapkan	Hasil yang Didapatkan	Waktu (ms)	Status
Pendaftaran pengguna baru dengan email unik.	Method: POST Endpoint : /api/auth/register Body: { "name": "User Baru", "email": "user.baru@example.com", "password": "password123" }	Status Code: 201 Created Body : JSON berisi data pengguna baru.	Status Code : 201 Created Body : JSON berisi data pengguna baru.	412	Berhasil
Pendaftaran dengan email yang sudah terdaftar.	Method: POST Endpoint : /api/auth/register Body: { "name": "User Lain", "email": "leonard.lesmana@example.com",	Status Code: 400 Bad Request Body: { "User error "Email sudah terdaftar" }	Status Code : 400 Bad Request Body : Pesan error "Email sudah terdaftar"	155	Berhasil

	"password": "password456" }		h terdftar".		
Login berhasil dengan kredensial yang valid.	Method: POST Endpoint : /api/auth/login Body: { "email": "leonard.lesmana@example.com", "password": "password123" }	Status Code: 200 OK Body: JSON berisi JWT.	Status Code : 200 OK Body : JSON berisi JWT.	128	Berhasil
Login dengan password yang salah.	Method: POST Endpoint : /api/auth/login Body: { "email": "leonard.lesmana@example.com", "password": "passwordsalah" }	Status Code: 401 Unauthorized Body: Pesan error "Email atau password salah".	Status Code : 401 Unauthorized Body : Pesan error "Email atau password salah".	198	Berhasil
Mengakses rute admin dengan token peran 'User'.	Method: GET Endpoint : /api/admin/users Header: Authorization: Bearer <user_token>	Status Code: 403 Forbidden Body: Pesan error "Akses ditolak".	Status Code : 403 Forbidden Body : Pesan error "Akses ditolak".	95	Berhasil
Mengakses rute yang dilindungi tanpa token.	Method: GET Endpoint : /api/admin/users Header: (tidak ada)	Status Code: 401 Unauthorized Body: Pesan error "Token tidak ditemukan".	Status Code : 401 Unauthorized Body : Pesan error "Token tidak ditemukan".	88	Berhasil

5. Tahap Evaluasi (*Evaluation*)

Hasil pengujian menunjukkan semua fungsi utama modul Autentikasi berjalan sesuai harapan, termasuk pendaftaran, login, dan proteksi rute berbasis peran. Keamanan sistem dinilai sudah kuat, sehingga pengembangan dapat berlanjut ke fase berikutnya dengan penambahan fitur lanjutan yang bergantung pada autentikasi.

M. Iterasi Inkremental Fase Kedua: Modul Admin

Pengembangan pada tahap ini berfokus pada implementasi fungsionalitas yang spesifik untuk peran Admin, yang merupakan inti dari solusi verifikasi dan tata kelola platform Niuniq. Tahap-tahap pada iterasi kedua dapat dijelaskan pada berikut ini:

1. Tahap Perencanaan (*Planning*)

Perencanaan pada fase kedua berfokus pada pengembangan endpoint API yang akan digunakan oleh Admin untuk mengelola pengguna dan memverifikasi produk

2. Tahap Perancangan (*Design*)

Sama seperti fase sebelumnya, tahap perancangan untuk modul Admin mengacu pada artefak-artefak yang telah dibuat pada poin H sampai poin K.

3. Tahap Pengembangan (*Development*)

Pada tahap ini, *endpoint* API untuk modul Admin diimplementasikan menggunakan Express.js. Setiap *endpoint* dilindungi oleh *middleware* otorisasi yang memastikan hanya pengguna dengan peran 'Admin' yang dapat mengaksesnya.

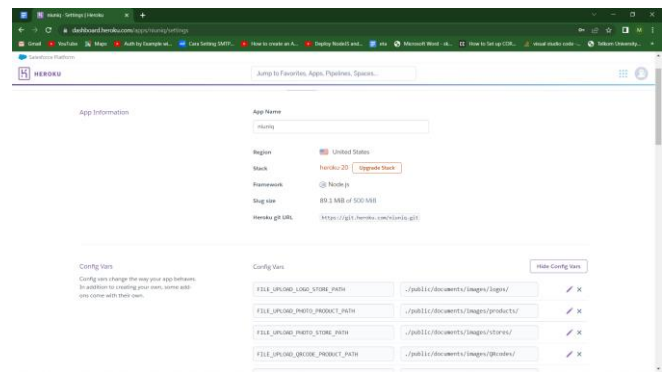
4. Tahap Pengujian (*Testing*)

Skenario pengujian mencakup kasus penggunaan yang valid oleh Admin serta upaya akses tidak sah oleh peran lain untuk memastikan keamanan dan integritas sistem, seperti yang dapat dilihat pada Tabel 11 (A)

TABEL 11 (A)

Deskripsi Pengujian	Data Masukan	Hasil yang Diharapkan	Hasil yang Didapatkan	Waktu (ms)	Status
Admin berhasil mengambil daftar semua pengguna.	Method: GET Endpoint : /api/admin/users Header: Authorization: Bearer <admin_token>	Status Code: 200 OK Body: JSON berisi daftar pengguna.	Status Code: 200 OK Body: JSON berisi daftar pengguna.	189	Berhasil
Admin berhasil mengambil daftar produk pending.	Method: GET Endpoint : /api/admin/products/pending Header: Authorization: Bearer <admin_token>	Status Code: 200 OK Body: JSON berisi daftar produk pending.	Status Code: 200 OK Body: JSON berisi daftar produk pending.	152	Berhasil
Admin berhasil menyetujui sebuah produk.	Method: PUT Endpoint : /api/admin/products/verify	Status Code: 200 OK Body: JSON berisi data produk	Status Code: 200 OK Body: JSON berisi data produk	215	Berhasil

	tion/71a9e3f2... Header: Authorization: Bearer <admin_token> Body: { "status": "verified" }	yang telah diperbarui.	yang telah diperbarui.		
Admin berhasil menolak sebuah produk dengan alasan.	Method: PUT Endpoint : /api/admin/product/verification/71a9e3f2... Header: Authorization: Bearer <admin_token> Body: { "status": "rejected", "message": "Foto produk tidak jelas." }	Status Code: 200 OK Body: JSON berisi produk dengan status 'rejected' dan pesan verifikasi.	Status Code: 200 OK Body: JSON berisi produk dengan status 'rejected' dan pesan verifikasi.	220	Berhasil
Pengguna non-admin gagal mengakses rute admin.	Method: GET Endpoint : /api/admin/product/pending Header: Authorization: Bearer <seller_token>	Status Code: 403 Forbidden Body: Pesan error "Akses ditolak".	Status Code: 403 Forbidden Body: Pesan error "Akses ditolak".	98	Berhasil



GAMBAR 6 (A)

V. KESIMPULAN

Berdasarkan proses penelitian hingga pengujian, backend aplikasi Niuniq untuk modul Admin dan Authentication berhasil dirancang dan diimplementasikan menggunakan Express.js dan MongoDB. Sistem menyediakan REST API yang mendukung verifikasi keaslian produk UMKM, termasuk fitur registrasi, autentikasi JWT, otorisasi berbasis peran, serta alur verifikasi oleh Admin. Metode Iterative Incremental terbukti efektif, dengan dua fase pengembangan: pertama untuk modul Authentication, kedua untuk logika bisnis Admin. Setiap komponen diuji terpisah sebelum integrasi. Pengujian black-box via Postman menunjukkan seluruh API berjalan sesuai spesifikasi, baik pada input valid maupun error, sehingga sistem dinyatakan siap diintegrasikan dengan frontend.

REFERENSI

- Addien, F. U., Atmiyati, S. U., & Setyawati, S. M. (2024). Dampak Penutupan Tiktok Shop di Indonesia. *Jurnal Inovasi, Riset, dan Pemerintahan Regional*, 8(2), 218–235.
- Astuti, W., dkk. (2024). *Meningkatkan Kepercayaan Konsumen Melalui Sistem Verifikasi Keaslian Produk Kecantikan Berbasis Blockchain dan QR Code*. Naskah Publikasi, Universitas Muhammadiyah Surakarta.
- Dinata, M. R. C., Fauzi, R., & Alam, E. N. (2022). Pengembangan Aplikasi Si-Book Untuk Monitoring Dan Evaluasi Kinerja Pegawai Pada Modul Assignment Dengan Metode Iterative Incremental. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 6(1), 1–9.
- Febriyanto, Triyono, J., & Rahayu, N. (2020). QRcode Verifikasi Sertifikat Sebagai Bukti Keabsahan Dokumen dalam Bidang Pendidikan. *Technomedia Journal (TMJ)*, 5(1), 96–108.
- Fikri, R. N. A., & Santoso, N. (2021). Pengembangan Aplikasi UMKM Penjualan Bahan Makanan Organik Siap Antar berbasis Android. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 5(9), 4042–4051.

- Tahap Evaluasi (*Evaluation*)
Pengujian fase kedua menunjukkan semua fungsi modul Admin berhasil diimplementasikan sesuai spesifikasi. Sistem mampu mendukung tugas pengawasan dan verifikasi oleh Admin, sesuai tujuan penelitian. Dengan selesainya modul Admin dan *Authentication*, pengembangan REST API dianggap selesai. Karena tidak ada fitur tambahan yang mendesak, *backend* siap dilanjutkan ke tahap *deployment*.
- Tahap Deployment
Merupakan proses mempublikasikan aplikasi *backend* yang telah diuji ke server agar dapat diakses secara global. Pada proyek ini, *deployment* dilakukan melalui layanan PaaS yang mendukung Node.js. Tujuannya adalah menyediakan REST API yang stabil bagi tim *frontend* untuk integrasi, sehingga pengembangan sisi klien dan server dapat berjalan paralel dan efisien. Gambar 6 (A) menunjukkan log keberhasilan deployment REST API Niuniq ke *cloud*.

6. Filiana, A., Rini, M. N. A., Prabawati, A. G., & Samat, R. A. (2022). Pengembangan REST API Untuk Informasi Pasar Tradisional Di Kota Yogyakarta Dengan Metode Iterative Incremental. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 6(3), 1234–1241.
7. Firdaus, A., & Ramadhani, D. A. (2021). Pengembangan Back End Berbasis REST API pada Sistem E-Partisipasi dan E-Inisiatif Patriot Pangan. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 5(11), 5123–5130.
8. Irianto, H., dkk. (2022). Digitalisasi UMKM sebagai Upaya Peningkatan Pemasaran dan Penjualan Online di Desa Tengkluk. *Journal of Cooperative, Small and Medium Enterprise Development*, 1(2), 60–64.
9. Lisgiani, R., & Nurmajid, S. (2022). Implementasi Autentikasi dari Sisi Backend pada Arsitektur Microservices Menggunakan Express JS. *Jurnal Infotronik*, 7(1), 27–32.
10. Mardiatmi, A. B. D., Dirkareshza, R., & Argo, J. G. (2021). Literasi dan Sosialisasi sebagai Upaya Menghindari Efek Hukum dan Ekonomi bagi Penjual dan Pembeli Barang Tiruan atau Barang Palsu pada UMKM. *Jurnal Pengabdian Masyarakat Berkemajuan*, 4(3), 1–8.
11. Maulani, A. N., dkk. (2024). *Meningkatkan Kepercayaan Konsumen Melalui Sistem Verifikasi Keaslian Produk Kecantikan Berbasis Blockchain dan QR Code*. Naskah Publikasi, Universitas Muhammadiyah Surakarta.
12. Mendrofa, L., Zendrato, B., & Zai, I. (2025). Pengaruh Digitalisasi Pada Peningkatan Efisiensi Operasional Usaha Mikro, Kecil, dan Menengah (UMKM) di Indonesia Tahun 2023. *Jurnal Ilmu Ekonomi, Pendidikan dan Teknik*, 2(1), 100–108.
13. Munawaroh, dkk. (2024). Hubungan Kepemimpinan Digital dan Daya Saing UKM. *Jurnal Manajemen dan Bisnis*, 12(1), 45–56.
14. Prabawati, A. G., dkk. (2024). Pengembangan Sistem Informasi Pengelolaan Data UMKM Gereja Studi Kasus: GKJ Klasis Klaten Timur. *JUSTIN (Jurnal Sistem dan Teknologi Informasi)*, 12(3), 441–450.
15. Rahmawati, D. (2022). Optimalisasi Teknologi dan Strategi Pemasaran Digital dalam Meningkatkan Daya Saing UMKM Bakpia Wong Yogyakarta. *Jurnal Nuansa*, 3(1), 282–298.
16. Riady, A. M. N., Paniran, P., & Sukmadana, I. M. B. (2024). Perancangan Backend Api Berbasis Rest-API pada Aplikasi Rekomendasi Resep Makanan. *Mars: Jurnal Teknik Mesin, Industri, Elektro Dan Ilmu Komputer*, 2(3), 94–106.
17. Shiddieq, D. F., Roji, F. F., Wufron, W., & Bekti, S. G. (2023). Model dan Implementasi Geographic Information System untuk Pemetaan UMKM di Kabupaten Garut. *Jurnal Algoritma*, 20(2), 386–397.
18. Solechan, Wijanarko, A., & Budi, S. (2023). Pengaruh Transformasi Digital Terhadap Kecepatan Layanan dan Kepercayaan Konsumen. *Jurnal Ekonomi dan Inovasi*, 10(2), 112–125.
19. Syahroni, A. W., Dewi, N. P., Ramadhani, N., Ubaidi, U., & Said, B. (2024). Uji Keamanan Back end Aplikasi Berbasis Website Menggunakan Metode Black Box Testing. *Jurnal PROCESSOR*, 19(2), 214–223.
20. Zaini, R. (2024). Efektivitas Digitalisasi UMKM dalam Meningkatkan Daya Saing Ekonomi di Era Industri 4.0. *Jurnal Pro Ekonomi*, 1(1), 26–33.