

BAB II

TINJAUAN PUSTAKA

2.1. Studi Sebelumnya

Penelitian sebelumnya yang dilakukan oleh Sumigar dkk pada tahun 2022, berfokus pada pengembangan sistem *e-commerce* untuk UMKM di Kecamatan Ranoyapo dengan metode *Extreme Programming* (XP) guna mengatasi keterbatasan pemasaran akibat belum adanya platform khusus. Sistem berbasis *website* ini dilengkapi fitur manajemen produk oleh *admin* toko, verifikasi pembayaran oleh *admin* utama, serta kemudahan bagi pengguna dalam bertransaksi, memberikan ulasan, dan mencari produk. Implementasi sistem menunjukkan peningkatan efisiensi pemasaran UMKM dan memudahkan masyarakat lokal menemukan produk yang dibutuhkan. Keunggulan metode XP terlihat dari kemampuan adaptasi sistem terhadap masukan pengguna secara dinamis. Penelitian ini merekomendasikan pengembangan integrasi pembayaran digital dan opsi login melalui pihak ketiga seperti *Google* atau *Facebook* untuk meningkatkan kenyamanan dan keamanan transaksi [4]. Penelitian ini menunjukkan bagaimana XP dapat diterapkan dalam konteks bisnis kecil-menengah, yang mirip dengan kondisi Gutsmuhs sebagai startup awal.

Pada penelitian selanjutnya dilakukan oleh Sari dkk pada tahun 2020, berfokus pada pengembangan sistem *e-commerce* untuk UMKM di Kabupaten Musi Rawas Utara dengan metode *Extreme Programming* (XP) yang menekankan fleksibilitas dan adaptasi cepat terhadap perubahan kebutuhan tanpa merombak desain sistem secara menyeluruh. Sistem ini memiliki fitur manajemen kategori UMKM, data barang, pemesanan, konfirmasi pembayaran, serta antarmuka untuk *admin* dan anggota. Keunggulan utama terletak pada kemampuan mengakomodasi perubahan secara *real-time* sesuai prinsip XP, meskipun terdapat tantangan teknis saat modifikasi kode dan database mendadak. Penelitian ini menegaskan efektivitas XP dalam pengembangan *e-commerce* UMKM dan merekomendasikan penguatan dokumentasi serta

optimasi respons sistem untuk meningkatkan kualitas dan fleksibilitas pengembangan selanjutnya [5]. Penelitian ini memiliki relevansi dalam konteks penggabungan berbagai fitur bisnis yang umum terdapat pada UMKM, yang juga menjadi fondasi utama dalam sistem yang dikembangkan untuk GutsMuths.

Penelitian berikutnya dilakukan oleh Suherman dkk pada tahun 2024, berfokus pada pengembangan sistem *e-commerce* untuk penjualan alat elektronik menggunakan metode *Extreme Programming* (XP) guna meningkatkan promosi dan penjualan UMKM di sektor elektronik. Dengan pendekatan XP yang iteratif dan melibatkan umpan balik pengguna, sistem yang dibangun berhasil mengintegrasikan fitur utama seperti registrasi, login, dan manajemen saran, serta menunjukkan tingkat keberhasilan pengujian fungsional sebesar 100%. Sistem ini terbukti memperluas jangkauan pelanggan dan meningkatkan efisiensi operasional, sementara rekomendasi pengembangan ke depan meliputi optimalisasi keamanan data dan integrasi pembayaran digital. Penelitian ini menegaskan efektivitas XP dalam pengembangan *e-commerce* UMKM sekaligus menawarkan solusi praktis atas tantangan pemasaran digital di bidang elektronik [6]. Penelitian ini menjelaskan bagaimana fungsi transaksi dan pengelolaan produk diterapkan sesuai dengan modul *e-commerce* dalam sistem GutsMuths.

Berikutnya adalah penelitian yang dilakukan oleh Ariyanto dkk pada tahun 2023. Penelitian ini bertujuan untuk mengembangkan aplikasi *Android* berbasis *REST API* untuk layanan bimbingan konseling di sekolah guna mempermudah pengaduan anonim dan interaksi *real-time* antara siswa dan guru konseling. Aplikasi ini menggunakan *Kotlin*, *MySQL*, dan *REST API*, dengan pengujian fungsional yang menunjukkan semua endpoint berstatus kode 200. Sistem menyediakan fitur pengaduan, riwayat respons, dan formulir saran, serta berhasil mengatasi tantangan integrasi *database* dan keamanan *API*. Penelitian ini membuktikan efektivitas *REST API* dalam layanan pendidikan dan merekomendasikan pengembangan fitur analitik serta integrasi dengan platform *e-learning* sekolah [7]. Penelitian ini berperan penting sebagai sumber

referensi teknis dalam pemanfaatan *REST API* untuk menghubungkan dua sistem yang berada di lingkungan berbeda, yang juga merupakan metode utama yang digunakan dalam penelitian ini.

Selanjutnya adalah Ardiansah dkk melakukan penelitian pada tahun 2023. Tujuan dari penelitian ini adalah mengembangkan sistem informasi akademik berbasis web untuk SDN Kuala Teladas menggunakan metode *Extreme Programming* (XP) guna mengatasi pengelolaan data manual yang rentan hilang. Sistem ini dirancang mudah diakses dan efisien, dengan *PHP* untuk *backend* dan *SQLyog* untuk database. Hasilnya, sistem mencakup fitur *input* nilai siswa, rekap nilai, dan laporan akademik yang disesuaikan untuk guru dan administrator. Pengujian *Technology Acceptance Model* (TAM) menunjukkan kelayakan sistem mencapai 92,71% dan aspek keamanan 93,56%, menandakan sistem memenuhi kebutuhan fungsional, mudah digunakan, dan aman. Penelitian ini berkontribusi pada penerapan XP yang efektif di sekolah dasar terpencil serta validasi kelayakan sistem melalui model TAM. Pengembangan selanjutnya disarankan untuk mencakup fitur analitik perkembangan siswa dan integrasi platform pembelajaran digital [8]. Penelitian ini menunjukkan bagaimana proses penerapan sistem web yang mengelola data dari berbagai jenis pengguna (*admin* dan mahasiswa) dalam satu sistem yang terorganisir, mirip dengan struktur peran pengguna pada sistem Gutsmuhs (*admin, influencer, customer*).

Berdasarkan hasil dari jurnal dan penelitian sebelumnya, peneliti dapat memberikan perbandingan data penelitian pada Tabel 2.1.

Tabel 2.1 Penelitian Sebelumnya

No	Penulis	Tahun	Domain	Tujuan	Metode	Hasil
1	Sumigar, dkk [4]	2022	<i>E-Commerce</i> Produk UMKM Kecamatan Ranoyapo Menggunakan Metode <i>Extreme Programming</i>	Membuat sistem <i>e-commerce</i> produk UMKM di Kecamatan Ranoyapo	<i>Extreme Programming</i> (XP)	Sistem Pemasaran dan Penjualan produk UMKM
2	Sari, dkk [5]	2020	Penerapan <i>E-commerce</i> Menggunakan Metode <i>Extreme Programming</i> pada UMKM Kabupaten Muratara	Menciptakan pemasaran dan penjualan produk UMKM berbasis teknologi informasi	<i>Extreme Programming</i> (XP)	Sistem Pemasaran dan Penjualan produk UMKM
3	Suherman, dkk [6]	2024	Sistem <i>E-commerce</i> untuk Meningkatkan Penjualan Alat Elektronik Menggunakan Metode <i>Extreme Programming</i>	Membuat sistem penjualan alat elektronik berbasis <i>e-commerce</i>	<i>Extreme Programming</i> (XP)	Sistem yang dikembangkan berhasil memperluas jangkauan pelanggan dan meningkatkan efisiensi operasional toko

No	Penulis	Tahun	Domain	Tujuan	Metode	Hasil
4	Ariyanto, dkk [7]	2023	Integrasi <i>REST API</i> berbasis <i>Android</i> untuk layanan bimbingan dan konseling	Mengembangkan aplikasi berbasis <i>Android</i> yang mengintegrasikan <i>REST API</i> untuk meningkatkan layanan bimbingan dan konseling di sekolah	<i>White Box Testing</i> , Pengujian <i>REST API</i> dengan <i>Postman</i>	Aplikasi berhasil dibangun menggunakan <i>Kotlin</i> dan <i>MySQL</i> serta diuji tanpa <i>error</i> , diharapkan dapat meningkatkan aksesibilitas dan efisiensi layanan konseling di lingkungan pendidikan
5	Ardiansah, dkk [8]	2023	Penerapan <i>Extreme Programming</i> Dalam Sistem Informasi Akademik SDN Kuala Teladas	Merancang bangun sistem informasi akademik berbasis <i>website</i> untuk pengelolaan data akademik sekolah secara efektif dan efisien	<i>Extreme Programming</i> (XP), <i>Technology Acceptance Model</i> (TAM)	Sistem berhasil diuji dengan tingkat kelayakan sebesar 92,71%, termasuk kategori sangat layak untuk meningkatkan efektivitas sekolah dalam menyajikan data akademik.

2.2. Dasar Teori

2.2.1. Sistem Informasi Terintegrasi

Sistem informasi terintegrasi adalah sistem yang menggabungkan beberapa komponen atau subsistem menjadi satu kesatuan yang saling terhubung, sehingga data dapat mengalir secara konsisten antar bagian dalam organisasi. Menurut Sanchez dkk (2020), proses sistem informasi terintegrasi adalah sistem pengelolaan data yang memungkinkan lebih efisien di organisasi dengan menghubungkan komponen sistem atau berbagai sistem [1]. Menurut Gaol dkk (2020), pengintegrasian sistem informasi mampu menyediakan data yang lebih akurat bagi manajemen puncak dalam pengambilan keputusan pasar, dan dengan diterapkannya sistem terintegrasi, setiap sektor dapat beroperasi secara optimal [9].

2.2.2. Website

Website adalah kumpulan halaman web yang saling terhubung dan dapat diakses melalui internet menggunakan browser. Setiap *website* memiliki alamat unik yang disebut *Uniform Resource Locator* (URL). Menurut Sari dkk (2020), *website* merupakan kumpulan halaman digital yang memuat informasi berupa teks, gambar, maupun video, yang dapat diakses secara global melalui jaringan internet oleh pengguna dari berbagai belahan dunia [2]. Menurut Rebah dkk (2022), *website* memungkinkan pengguna mengakses halaman-halaman yang tersedia di berbagai situs web melalui *browse* [10]. Menurut Ramadhanti dkk (2020), situs web *e-commerce* adalah kumpulan halaman web yang terdiri dari informasi data yang dikelola oleh sebuah perusahaan untuk melakukan proses jual beli melalui jaringan internet [11].

2.2.3. Arsitektur

Arsitektur sistem adalah rancangan struktur dan pengaturan bagian-bagian dalam sebuah sistem, mencakup hubungan antar bagian, aturan kerja, serta interaksi antar komponen untuk mencapai tujuan yang diinginkan. Menurut Rizki dkk (2024), arsitektur tidak hanya merujuk pada struktur fisik,

tetapi juga mencakup konteks rekayasa perangkat lunak, yang melibatkan komponen sistem seperti jaringan, perangkat keras, dan perangkat lunak yang dirancang dan diorganisasikan secara sistematis. Dalam konteks konstruksi fisik maupun digital, arsitektur mencakup desain keseluruhan yang meliputi jenis konstruksi di dunia nyata maupun virtual [12].

2.2.4. *Point-to-Point Integration*

Point-to-Point Integration adalah metode menghubungkan dua sistem atau aplikasi secara langsung menggunakan antarmuka khusus (*custom interface*) atau *API* untuk pertukaran data tanpa menggunakan perantara. Menurut Laakkonen (2023), integrasi ini melibatkan model data dari sistem asal dan tujuan yang dihubungkan melalui kode, yang memungkinkan pertukaran informasi antara keduanya. Setiap komponen tambahan yang terhubung secara signifikan meningkatkan jumlah integrasi yang dikelola [13].

2.2.5. *Application Programming Interface (API)*

Application Programming Interface (API) adalah kumpulan aturan, standar, dan alat yang memungkinkan berbagai aplikasi atau sistem untuk saling berinteraksi dan bertukar data secara terorganisir tanpa harus memahami cara kerja internal masing-masing. Menurut Lamothe dkk (2021), *API* merupakan antarmuka yang berfungsi sebagai titik akses ke komponen perangkat lunak yang dapat digunakan kembali. *API* bukanlah perangkat lunak yang berdiri sendiri, melainkan merupakan bagian dari pustaka perangkat lunak, kerangka kerja, atau layanan web yang menyediakannya [14].

2.2.6. *Representational State Transfer (REST)*

Representational State Transfer (REST) adalah arsitektur standar untuk mendesain sistem terdistribusi seperti web services yang mengandalkan protokol *HTTP/HTTPS*. Ini merupakan pendekatan stateless yang mengatur komunikasi antara client dan server dengan prinsip-prinsip tertentu untuk membuat sistem yang *scalable*, modular, dan efisien. Menurut Asgar dkk (2022), *REST* merupakan suatu pendekatan arsitektur perangkat lunak yang menggunakan prinsip-prinsip sederhana dan fleksibel untuk komunikasi antar

sistem, di mana data ditransfer melalui antarmuka standar seperti *HTTP*, mirip dengan cara kerja aplikasi *website* pada umumnya. Klien mengirim permintaan ke server menggunakan protokol *HTTP*, dan server memberikan respons kembali kepada klien [15]. Menurut Arcuri (2019), setiap sumber daya dalam *REST* harus diidentifikasi menggunakan URI. Representasi sumber daya, seperti *JSON* atau *XML*, yang dikirimkan ke klien tidak bergantung pada format aslinya, misalnya data yang berasal dari baris dalam basis data relasional. Pengelolaan sumber daya dilakukan dengan metode *HTTP* yang sesuai, seperti menggunakan permintaan *DELETE* untuk menghapus sumber daya, bukan *POST* [16].

2.2.7. API Integration

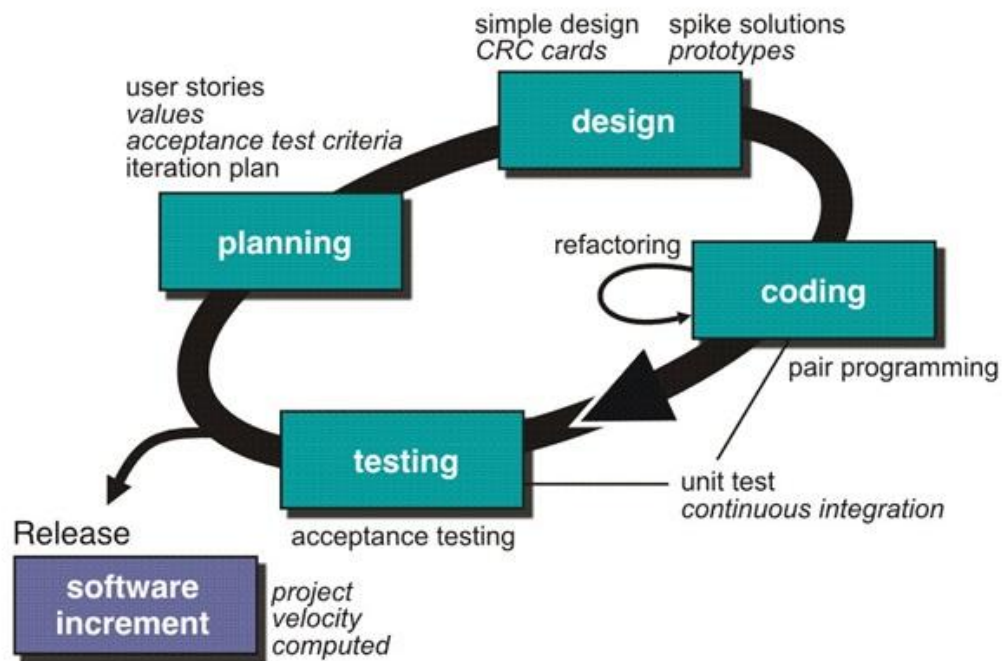
API Integration adalah proses menghubungkan dua atau lebih aplikasi, sistem, atau layanan menggunakan *Application Programming Interfaces (API)* untuk bertukar data dan fungsionalitas secara otomatis dan efisien. Ini memungkinkan sistem yang berbeda untuk berkomunikasi dan bekerja sama tanpa memerlukan integrasi manual atau kode custom yang kompleks. Menurut Ofoeda dkk (2019), dalam integrasi dengan *API*, terdapat dua pihak yang terlibat. Di satu sisi, ada penyedia *API* yang menawarkan layanan mereka melalui eksposur *API*, sementara di sisi lain, terdapat konsumen *API* yang memanfaatkan layanan tersebut dengan berinteraksi langsung dengan *API* [17].

2.2.8. Metode Extreme Programming

Extreme Programming (XP) adalah metodologi agile dalam pengembangan perangkat lunak yang menekankan kualitas, adaptabilitas, dan kolaborasi tim melalui praktik teknis dan nilai-nilai khusus. XP cocok untuk proyek dengan kebutuhan yang sering berubah dan bertujuan memberikan nilai bisnis lebih cepat dengan mengurangi risiko melalui iterasi singkat dan umpan balik terus-menerus. Menurut Gedriyansah (2020), *Extreme Programming (XP)* merupakan salah satu metode pengembangan perangkat lunak yang banyak diterapkan oleh para pengembang dalam pembuatan aplikasi, dengan tujuan untuk menyederhanakan setiap tahapan dalam proses pengembangan sehingga lebih adaptif dan fleksibel terhadap perubahan kebutuhan [18].

Tabel 2.2 Perbandingan Metode Extreme Programming dengan Metode Lain

Kriteria	<i>Extreme Programming (XP)</i>	<i>Waterfall</i>	<i>Scum</i>	<i>Spiral</i>
Pendekatan	Iteratif dan inkremental	Linear dan berurutan	Iteratif dan inkremental	Iteratif dan berbasis risiko
Fleksibilitas	Sangat fleksibel terhadap perubahan	Tidak fleksibel terhadap perubahan	Fleksibel	Cukup fleksibel dengan manajemen risiko
Dokumentasi	Minim, fokus pada komunikasi langsung	Lengkap dan formal	Cukup fleksibel dalam dokumentasi	Menekankan pada dokumentasi risiko
Keterlibatan Klien	Sangat tinggi, klien terlibat dalam setiap iterasi	Terbatas, lebih banyak di awal proyek	Tinggi, terutama melalui product owner	Sedang, melalui penilaian risiko berkala
Kecepatan Pengembangan	Tinggi (karena iterasi pendek dan berkelanjutan)	Lambat karena mengikuti urutan tahapan	Tinggi (sprint pendek)	Sedang (tergantung pada siklus risiko)
Tim Pengembangan	Kecil dan kolaboratif	Dapat besar, tapi kurang kolaboratif	Cross-functional dan kolaboratif	Biasanya tim ahli di berbagai bidang
Fokus Utama	Kualitas kode, komunikasi, dan feedback cepat	Perencanaan awal dan dokumentasi detail	Nilai bisnis, backlog, dan iterasi	Manajemen risiko dan kelayakan teknis
Relevansi dengan penelitian	Sangat relevan karena sesuai untuk tim kecil dan kebutuhan cepat	Kurang relevan karena kurang adaptif	Cukup relevan tapi lebih cocok untuk tim besar	Relevan bila risiko teknis proyek tinggi



Gambar 2.1 Tahapan Metode Extreme Programming

Tahapan utama dalam XP meliputi perencanaan, perancangan, pengkodean, dan pengujian, dengan fokus pada komunikasi yang intensif, kesederhanaan kode, dan respon cepat terhadap perubahan kebutuhan pelanggan. Berdasarkan gambar tahapan pengembangan di atas, berikut ini merupakan uraian dari masing-masing tahapannya:

1. *Planning* (Perencanaan)

Perencanaan adalah tahap awal dalam pengembangan perangkat lunak yang melibatkan pengumpulan dan analisis kebutuhan pengguna serta pemangku kepentingan. Pada tahap ini, tim pengembang mendefinisikan tujuan, fitur, fungsi, dan *output* yang diharapkan dari perangkat lunak, sekaligus membuat jadwal pelaksanaan dan estimasi sumber daya yang dibutuhkan untuk proyek tersebut. Menurut Ahmad dkk (2020), *Extreme Programming* (XP) berfokus pada identifikasi alur proses perangkat lunak yang akan dikembangkan, perumusan tujuan akhir dari perangkat lunak, penentuan fitur-fitur yang akan disediakan, fungsi utama aplikasi, serta perencanaan alur kerja dalam proses pengembangannya [19].

2. *Design* (Perancangan)

Langkah selanjutnya adalah proses mendetail yang dilakukan setelah perencanaan, di mana arsitektur sistem dan spesifikasi teknis dirancang secara rinci. Tahap ini mencakup pembuatan blueprint sistem, termasuk struktur data, algoritma, antarmuka pengguna, serta model-model seperti diagram *Unified Modeling Language* (UML) yang menggambarkan interaksi sistem dan komponen-komponennya. Menurut Pratama dkk (2022), dalam tahap perancangan, dilakukan aktivitas modelisasi mulai dari modelisasi arsitektur, modelisasi sistem, hingga modelisasi *database* (basis data) [20].

3. *Coding* (Pengkodean)

Pengkodean merupakan tahap pelaksanaan di mana rancangan teknis yang telah disusun sebelumnya diubah menjadi kode program menggunakan bahasa pemrograman tertentu. Pada fase ini, pengembang membuat kode sumber yang dapat dijalankan oleh komputer dengan mengimplementasikan seluruh fungsi dan fitur yang sudah dirancang. Menurut Fauzan dkk (2023), tahapan ini mewujudkan rancangan model sistem menjadi kode program yang menghasilkan prototipenya. Proses ini melibatkan *Coding* (pengkodean) yang dikerjakan oleh *Programmer* atau *Software Engineer* berdasarkan rencana dan desain yang telah dirancang pada tahap sebelumnya [21].

4. *Testing* (Pengujian)

Pengujian adalah proses evaluasi perangkat lunak setelah tahap pengkodean selesai, yang bertujuan untuk memastikan program berjalan sesuai dengan spesifikasi dan bebas dari kesalahan. Proses ini mencakup berbagai jenis pengujian seperti *unit testing*, *integration testing*, dan *user acceptance testing*, yang dilakukan untuk menemukan serta memperbaiki kesalahan dan menjamin kualitas serta keandalan perangkat lunak sebelum digunakan secara luas. Menurut Sulistiani dkk (2021), fokus utama ini terletak pada pengujian fitur serta fungsionalitas sistem yang dilakukan menggunakan metode *black box testing* [22].