

BAB II KAJIAN PUSTAKA

2.1 Tinjauan Pustaka

Penggunaan arsitektur *microservices* (MSA) dalam sistem *e-learning* telah menjadi salah satu pendekatan yang menarik perhatian para peneliti karena kemampuannya dalam mengelola kompleksitas sistem dan meningkatkan skalabilitas. Palamarchuk (2022) menjelaskan bahwa MSA memungkinkan aplikasi besar dipecah menjadi layanan-layanan kecil yang independen dan dapat dikelola secara terpisah. Dengan pendekatan ini, sistem *e-learning* dapat dirancang secara modular, di mana setiap layanan memiliki fungsi spesifik tanpa saling bergantung secara langsung, sehingga meningkatkan *fleksibilitas* dan *efisiensi* dalam pengembangan serta pemeliharaan sistem. Hal ini sangat berbeda dibandingkan dengan arsitektur *monolitik* yang sering digunakan dalam *Learning Management Systems* (LMS). Dinova & Utomo (2024) menyoroti bahwa meskipun sederhana, arsitektur *monolitik* memiliki keterbatasan dalam hal adaptabilitas terhadap perubahan kebutuhan sistem, terutama dalam konteks sistem pendidikan yang dinamis.

Selain itu, arsitektur berbasis *Cloud Computing* telah diadopsi secara luas dalam sistem *e-learning* untuk mendukung era Pendidikan 4.0, yang merupakan adaptasi dari perkembangan Industri 4.0. Hendradi et al. (2019) menguraikan bahwa arsitektur berbasis *cloud* tidak hanya memberikan *efisiensi* operasional, tetapi juga memungkinkan penyimpanan data yang besar dan aksesibilitas tinggi, yang menjadi kunci keberhasilan sistem pembelajaran modern. Di sisi lain, Liu & Yu (2023) menekankan bahwa dengan meningkatnya jumlah pengguna dan volume data, sistem *e-learning* berbasis *Big Data* menjadi penting untuk menangani tantangan teknis dan pedagogis. Solusi berbasis *Big Data* memungkinkan analisis data yang lebih baik, yang mendukung personalisasi pembelajaran dan pengambilan keputusan berbasis data.

Arsitektur modular juga diterapkan dalam sistem rekomendasi afektif untuk lingkungan belajar adaptif. Salazar et al., (2024) mengembangkan arsitektur generik yang memisahkan mesin emosi dari mesin karakteristik pribadi, memberikan *fleksibilitas* tinggi dalam penyesuaian konten pembelajaran. Sistem ini dirancang untuk meningkatkan keterlibatan siswa dengan memberikan rekomendasi konten yang relevan berdasarkan emosi dan preferensi belajar individu.

Dalam konteks pengembangan LMS, Rabiman et al., (2020) menggunakan model *Hannafin* dan *Peck* yang melibatkan tahapan analisis kebutuhan, desain, pengembangan, dan implementasi. Pendekatan ini terbukti meningkatkan efisiensi dan kualitas pembelajaran, khususnya dalam lingkungan pendidikan vokasi. Evaluasi yang dilakukan berdasarkan kemudahan penggunaan, fungsionalitas, komunikasi visual, dan desain pembelajaran menunjukkan bahwa LMS berbasis metode ini sangat mudah diterapkan.

Framework enterprise architecture, seperti TOGAF dan Zachman, juga menjadi pendekatan yang relevan dalam merancang sistem *e-learning* yang terintegrasi dan terstruktur. Setyawan (2020) menunjukkan bagaimana *framework* ini dapat mendukung integrasi data antarbagian organisasi dan meningkatkan operasional institusi pendidikan. Pendekatan ini memberikan dasar yang kuat untuk memastikan bahwa sistem *e-learning* tidak hanya memenuhi kebutuhan teknis tetapi juga mendukung keberlanjutan pendidikan.

Dengan prevalensi arsitektur *microservices*, *cloud computing*, dan *big data*, terdapat peluang besar untuk mengatasi tantangan yang dihadapi sistem *e-learning* saat ini. Milovanović, (2021) mengusulkan integrasi lebih lanjut antara MSA dan LMS yang ada, seperti Moodle, untuk menciptakan sistem pembelajaran yang lebih *adaptif* dan *efektif*. Dengan demikian, penelitian lebih lanjut dalam area ini menjadi krusial untuk menghadirkan solusi inovatif yang dapat memenuhi kebutuhan pembelajaran di era digital.

Arsitektur *Microservicess* (Telang, 2023) adalah pendekatan pengembangan perangkat lunak di mana sebuah aplikasi dibagi menjadi layanan-layanan kecil yang

independen, yang dapat dikembangkan, diterapkan, dan diskalakan secara terpisah. Setiap *microservices* menjalankan fungsi bisnis tertentu dan berkomunikasi dengan *microservices* lainnya melalui API (*Application Programming Interface*).

Arsitektur *monolith* mengintegrasikan semua komponen aplikasi dalam satu unit, memudahkan pengembangan awal dan *deployment*. Namun, seiring bertambahnya kompleksitas, *monolith* menjadi sulit diskalakan dan diperlihara karena semua modul bergantung. *Skalabilitas* bersifat vertikal, artinya seluruh aplikasi harus diskalakan meskipun hanya satu modul yang membutuhkan peningkatan sumber daya (Hussein et al., 2024).

Sebaliknya, *microservices* memecah aplikasi menjadi layanan kecil yang otonom, memungkinkan pengembangan, pengujian, dan *deployment independen*. Setiap layanan dapat menggunakan teknologi yang berbeda dan diskalakan secara horizontal. Namun, *microservices* memperkenalkan kompleksitas tambahan dalam komunikasi antar layanan dan memerlukan mekanisme orkestrasi serta monitoring yang lebih canggih (Elektronik et al., 2023).

Dari segi performa, pengujian menunjukkan bahwa arsitektur *monolith* cenderung menggunakan lebih banyak central processing unit (CPU) di bawah beban berat dibandingkan *microservices*. *Microservices* membagi beban kerja secara lebih merata antar layanan, meningkatkan efisiensi sumber daya (Gos & Zabierowski, 2020).

Pemilihan antara *monolith* dan *microservices* harus mempertimbangkan kebutuhan aplikasi, kapasitas tim, serta faktor non-fungsional seperti *maintainability*, *security*, dan *fault tolerance*. *Microservices* cocok untuk aplikasi besar dan kompleks, sementara *monolith* lebih efektif untuk aplikasi kecil hingga menengah (Su et al., 2024).

Berdasarkan penelitian yang dilakukan oleh Hussein et al. (2024), Elektronik et al. (2023), Su et al. (2024), serta Gos dan Zabierowski (2020) maka dapat disusun tabel berikut yang menunjukkan perbedaan antara arsitektur *microservices* dan *monolith*.

Tabel 2. 1. Perbedaan Arsitektur Secara Konseptual antara Microservices dengan Monolith

ASPEK	MICROSERVICES	MONOLITH
Struktur Aplikasi	Aplikasi terdiri dari layanan-layanan kecil yang berdiri sendiri, masing-masing memiliki codebase dan tanggung jawab	Aplikasi tunggal dengan satu codebase, semua komponen (UI, logika bisnis, database) digabung dalam satu unit
Modularitas	Layanan otonom dengan tanggung jawab spesifik	Terdapat modul internal tetapi saling terkait erat
Komunikasi antar komponen	Komunikasi antar layanan melalui jaringan menggunakan API/HTTP, gRPC dsb	Komunikasi langsung dalam satu proses (in-process calls) tanpa overhead jaringan
Database	Setiap layanan memiliki database sendiri (database per service) untuk menghindari ketergantungan antar layanan	Satu database yang dibagikan oleh seluruh aplikasi
Teknologi yang digunakan	Setiap layanan bisa menggunakan stack teknologi yang berbeda	Biasanya menggunakan satu stack teknologi untuk seluruh aplikasi
Deployment	Setiap layanan bisa di-deploy secara independen. Perubahan pada satu layanan tidak memengaruhi layanan lain	Aplikasi dikompilasi dan dideploy sebagai satu unit. Perubahan kecil memerlukan rebuild dan redeploy seluruh aplikasi
Scalability	Menskalakan layanan tertentu sesuai kebutuhan (horizontal scaling)	Menskalakan seluruh aplikasi meskipun hanya satu bagian yang memerlukan resource lebih (vertical scaling)
Kegagalan sistem	Lebih tahan banting. Kegagalan pada satu layanan tidak memengaruhi layanan lainnya (fault isolation)	Rentan terhadap single point of failure. Jika satu komponen gagal, seluruh aplikasi bisa down

2.2 Penelitian sebelumnya

Perkembangan teknologi informasi telah mendorong inovasi signifikan dalam sistem e-learning, menciptakan berbagai pendekatan dan arsitektur untuk meningkatkan efektivitas pembelajaran daring. Penelitian-penelitian terbaru menunjukkan bahwa personalisasi, fleksibilitas, dan skalabilitas menjadi faktor utama dalam desain sistem e-learning yang sukses.

Roux et al., (2022) mengusulkan arsitektur *multi-layer* dengan *Explainable AI* (XAI) untuk sistem rekomendasi hybrid yang memberikan saran pembelajaran

personal kepada siswa dan informasi pemantauan bagi pengajar. Milovanović, (2021) mengembangkan arsitektur e-learning berbasis *microservices* yang terintegrasi dengan LMS seperti Moodle menggunakan pendekatan *Domain Driven Design* (DDD). Sistem ini meningkatkan *fleksibilitas*, *skalabilitas*, dan mempermudah integrasi data melalui *cloud* data platform.

Krалеva et al., (2020) merancang LMS ramah anak menggunakan user centered design (UCD) yang menghadirkan antarmuka intuitif dan fitur gamifikasi. LMS ini dirancang khusus untuk siswa usia dini dengan dukungan spesifikasi SCORM untuk interoperabilitas materi ajar. Salazar et al., (2024) memperkenalkan sistem rekomendasi berbasis afeksi yang menyesuaikan materi pembelajaran berdasarkan emosi siswa, memperkuat personalisasi pembelajaran. Di sisi lain, Hendradi, Khanapi, & Mahfuzah, (2019) mengusulkan arsitektur e-learning berbasis cloud computing yang mendukung fleksibilitas distribusi konten di era Education 4.0.

Setyawan, (2020) menggunakan TOGAF Framework untuk merancang enterprise architecture sistem e-learning di Kanisius School, menciptakan blueprint pengembangan sistem yang terintegrasi. Palamarchuk, (2022) mengusulkan arsitektur e-learning berbasis *microservices* yang fokus pada skalabilitas dan toleransi gangguan dengan layanan independen yang mudah dikembangkan.

Penelitian – penelitian yang telah dikaji menunjukkan berbagai pendekatan inovatif dalam pengembangan sistem e-learning, mulai dari arsitektur *microservices*, *cloud computing*, hingga sistem rekomendasi berbasis kecerdasan buatan (AI) dan afeksi. Setiap studi berkontribusi pada peningkatan fleksibilitas, skalabilitas, dan personalisasi dalam pembelajaran daring. Penelitian – penelitian tersebut menjadi acuan dan dasar dalam pengembangan penelitian ini. Rangkuman hasil penelitian disajikan dalam tabel berikut.

Tabel 2. 2 Matrik Penelitian

No	Judul dan Penulis	Metode	Hasil
1	Design and Development E-Learning System by LMS in Vocational Education. Rabiman, Muhammad Nurtanto, Nur Kholifah. International Journal of Scientific & Technology Research, Vol 9, Issue 01, Jan 2020	Hannafin and Peck model (Needs Analysis, Design, Development, Implementation). Data dikumpulkan melalui kuesioner dan observasi.	LMS yang dikembangkan dinilai "sangat layak" dan meningkatkan kepuasan serta kualitas pembelajaran di pendidikan vokasi
2	Design and Implementation of an E-Learning Platform Using N-Tier Architecture. Nawar S. Alseelawi, Enas K. Adnan, Hussein T. Hazim, Haider TH. Salim ALRikabi, Khalid Waleed Nasser. International Journal of Interactive Mobile Technologies (iJIM), Vol 14, No 6, 2020	ADDIE Model (Analysis, Design, Development, Implementation, Evaluation) dan N-Tier Architecture (UI, BAL, DAL, DB)	Platform e-learning berbasis N-Tier Architecture meningkatkan interaksi antara dosen dan mahasiswa di universitas Irak, serta memecahkan masalah pada sistem pendidikan tradisional
3	Pengembangan Mobile LMS dengan UCD menggunakan Flutter Framework. Condro Kartiko, Ariq Cahya Wardhana, Diovianto Putra Rakhmadani. Jurnal Media Informatika Budidarma, Vol 6, No 2, April 2022	User Centered Design (UCD) menggunakan Flutter Framework.	Mobile LMS yang dikembangkan berfungsi sesuai kebutuhan pengguna (mahasiswa), memudahkan pembelajaran daring melalui smartphone
4	A Generic Architecture of an Affective Recommender System for E-Learning Environments. Juan Camilo Salazar, Jose Aguilar, Julián Monsalve-Pulido, Edwin Montoya. Universal Access in the Information Society, 2024	Pengembangan arsitektur modular berbasis analisis emosi menggunakan teknik pengenalan emosi dan profil siswa.	Arsitektur dapat merekomendasikan materi pembelajaran berdasarkan emosi dan karakteristik siswa, meningkatkan personalisasi dan efektivitas pembelajaran
5	Designing Effective EA for E-Learning System in Kanisius School using TOGAF Framework. Albertus Henny Setyawan, Rachmawan Adwitia Atmaja, Nilo Legowo. International Journal of Advanced Science and Technology, Vol 29, No 5, 2020	Enterprise Architecture menggunakan TOGAF Framework dengan pendekatan ADM (Architecture Development Method).	Blueprint sistem e-learning untuk Kanisius School yang dapat digunakan sebagai acuan pengembangan sistem di masa depan
6	Cloud Computing Based E-Learning System Architecture in Education 4.0.	Pengembangan arsitektur e-learning berbasis Cloud Computing	Arsitektur cloud-based e-learning yang diusulkan mampu memenuhi kebutuhan

Lanjutan Tabel 2. 2 Matrik Penelitian

No	Judul dan Penulis	Metode	Hasil
	P. Hendradi, M. Khanapi, S. N. Mahfuzah. Journal of Physics: Conference Series, Vol 1196, 2019	dengan lima lapisan (Infrastructure, Software, Resource Management, Service, Application Business Layer).	Education 4.0, meningkatkan skalabilitas dan efisiensi sumber daya
7	Methods of Building Microservices Architecture of E-Learning Systems. Y. A. Palamarchuk Information Technologies and Computer Engineering, 2022	Pengembangan arsitektur microservices untuk e-learning, menggunakan pendekatan domain-driven design dan API Gateway.	Arsitektur microservices meningkatkan skalabilitas, fleksibilitas, dan keandalan sistem e-learning dengan memisahkan fungsi ke dalam layanan independen
8	A Multi Layers Architecture For An E-Learning Hybrid Recommender System. Lisa Roux, Pantxika Dagorret, Patrick Etcheverry, Thierry Nodenot, Christophe Marquesuzaa, Philippe Lopistéguy. IADIS International Conference Cognition and Exploratory Learning in Digital Age - CELDA '21, Oct 2021	Pengembangan sistem rekomendasi hibrida berbasis multi-layer dengan Explainable AI untuk e-learning. Fokus pada personalisasi rekomendasi dan transparansi dalam proses rekomendasi.	Sistem mampu memberikan rekomendasi yang relevan untuk siswa dan informasi bagi guru untuk memantau kemajuan siswa. Arsitektur multi-layer memungkinkan interpretasi hasil oleh pengguna
9	Microservices Architecture in E-learning. Ana Milovanović 2021 International Conference on E-business Technologies (EBT)	Domain Driven Design (DDD) untuk mengidentifikasi microservices dan integrasi dengan LMS (contohnya Moodle) menggunakan API Gateway dan Message Broker.	Arsitektur microservices yang diusulkan memungkinkan integrasi fleksibel dengan LMS seperti Moodle dan Cloud Data Platform, meningkatkan skalabilitas dan modularitas sistem e-learning
10	An Approach to Designing and Developing an LMS Framework Appropriate for Young Pupils. Radoslava Kraveva, Mehrudin Sabani, Velin Kravev, Dafina Kostadinova International Journal of Electrical and Computer Engineering (IJECE), Vol. 10, No. 2, April 2020	User-Centered Design (UCD) dalam merancang LMS yang ramah anak dengan fokus pada antarmuka intuitif dan fitur pembelajaran interaktif.	LMS yang dikembangkan sesuai untuk anak-anak usia dini dengan antarmuka yang menarik dan mudah digunakan. Fitur gamifikasi dan pendekatan berbasis SCORM meningkatkan keterlibatan siswa muda dalam pembelajaran daring

2.3 Landasan Teori

Untuk mendukung penelitian ini, diperlukan landasan teori yang kuat agar analisis yang dilakukan dapat berjalan secara sistematis dan terarah. Landasan teori berfungsi sebagai pijakan dalam mengkaji permasalahan, sehingga penelitian dapat mencapai tujuan yang telah ditetapkan serta memberikan manfaat yang relevan sesuai dengan konteks penelitian.

2.3.1 E-Learning

E-Learning adalah pembelajaran jarak jauh (*distance Learning*) yang memanfaatkan teknologi komputer, jaringan komputer dan/atau Internet. *E-Learning* memungkinkan pembelajar untuk belajar melalui komputer di tempat mereka masing-masing tanpa harus secara fisik pergi mengikuti pelajaran/perkuliahan di kelas. *E-Learning* sering pula dipahami sebagai suatu bentuk pembelajaran berbasis web yang bisa diakses dari intranet di jaringan lokal atau internet. *E-learning* adalah sistem pendidikan *virtual* yang didukung oleh teknologi informasi.

E-learning merupakan metode pembelajaran berbasis teknologi informasi yang memungkinkan proses belajar-mengajar dilakukan secara *fleksibel* tanpa batasan waktu dan tempat. Dalam *e-learning*, berbagai *platform* seperti *Learning Management System* (LMS) digunakan untuk mengelola materi pembelajaran, memfasilitasi komunikasi, dan melacak kemajuan peserta didik. *Moodle*, misalnya adalah salah satu *platform open-source* yang populer, dirancang untuk mendukung kegiatan belajar *online* dengan menyediakan alat seperti pengelolaan tugas, ujian, dan materi berbasis multimedia (Alrikabi et al., 2022). Pada masa pandemi COVID-19, *e-learning* menjadi solusi utama untuk menjaga kontinuitas pendidikan di tengah pembatasan fisik. *Platform* seperti *Zoom*, *Google Classroom*, dan *Moodle* telah digunakan secara luas untuk memenuhi kebutuhan pembelajaran daring, meskipun tantangan seperti akses internet yang tidak merata dan kurangnya pelatihan teknis bagi pengajar masih menjadi kendala (Hasan & Khan, 2020; Hoq, 2020).

Keunggulan utama *e-learning* terletak pada *fleksibilitas* dan *aksesibilitasnya*, yang memungkinkan peserta didik untuk belajar sesuai kecepatan dan preferensi mereka sendiri. Namun, kelemahan seperti kurangnya interaksi tatap muka dan potensi rasa isolasi bagi peserta didik juga perlu diperhatikan. Oleh karena itu, kombinasi teknologi yang inovatif, seperti penggunaan perangkat pintar dan integrasi kecerdasan buatan, semakin penting untuk menciptakan lingkungan pembelajaran yang lebih cerdas dan adaptif (Chahal & Patel, 2021; Alrikabi et al., 2022). Dengan terus berkembangnya teknologi, *e-learning* tidak hanya mendukung proses pembelajaran formal tetapi juga menjadi sarana penting dalam pengembangan keterampilan digital dan pembelajaran seumur hidup.

2.3.2 *Learning Management System (LMS)*

Sistem Manajemen Pembelajaran atau *Learning Management System (LMS)* adalah *platform* berbasis web yang dirancang untuk mendukung proses pembelajaran dalam berbagai bentuk, termasuk pembelajaran daring, tatap muka, maupun campuran (*blended learning*). LMS memungkinkan penyampaian, pengelolaan, dan pelacakan materi serta aktivitas pembelajaran secara efisien. Dalam konteks pendidikan tinggi, LMS telah menjadi alat yang penting untuk meningkatkan pembelajaran *kolaboratif*, *fleksibilitas*, dan *aksesibilitas* tanpa batasan waktu dan tempat (Turnbull et al., 2021 ; Barreto et al., 2020). LMS juga memungkinkan institusi pendidikan dan organisasi bisnis untuk mengelola pelatihan, meningkatkan keterlibatan peserta, serta mengurangi biaya operasional melalui *otomatisasi* dan *eliminasi* kebutuhan akan ruang fisik (Barreto et al., 2020).

Seiring dengan berkembangnya teknologi, LMS telah melalui *transformasi signifikan*, mulai dari mesin pengajaran pada tahun 1924 hingga *platform* berbasis *cloud* seperti *Moodle* yang diluncurkan pada 2012 (Barreto et al., 2020). LMS modern mencakup berbagai fitur, seperti integrasi multimedia, penilaian otomatis, gamifikasi, dan pelacakan kemajuan peserta didik (Turnbull et al., 2021). Keberhasilan adopsi

LMS sering dievaluasi menggunakan model-model seperti *technology acceptance model* (TAM) dan *DeLone and McLean IS Success Model*, yang membantu mengidentifikasi faktor-faktor eksternal seperti persepsi kegunaan, kemudahan pengguna, dan dukungan organisasi yang memengaruhi penerimaan LMS di lingkungan pendidikan (Al-Nuaimi & Al-Emran, 2021).

Selain itu, LMS mendukung model pembelajaran jarak jauh yang memberikan *fleksibilitas* waktu dan ruang, memungkinkan penjarar menjadi fasilitator daripada hanya sebagai penyampai informasi. LMS juga mengakomodasi pembelajaran berbasis data untuk analisis hasil belajar yang lebih baik (Cavus, 2015 ; Turnbull et al., 2020)

2.3.3 **Arsitektur *E-Learning***

Arsitektur *e-learning* mengacu pada struktur dan desain sistem yang mendukung proses pembelajaran elektronik, yang memungkinkan penyampaian materi pendidikan melalui media digital dan *platform online*. Arsitektur ini mencakup berbagai komponen seperti perangkat keras, perangkat lunak, jaringan komunikasi, serta elemen-elemen pengelolaan dan interaksi yang memungkinkan akses, distribusi, dan penggunaan materi pembelajaran oleh siswa dan pengajar. Menurut Anderson (2020), arsitektur *e-learning* harus dirancang untuk mendukung *fleksibilitas* dalam pengelolaan konten dan memberikan pengalaman pengguna yang baik, sehingga dapat meningkatkan efektivitas pembelajaran.

Komponen utama dari arsitektur *e-learning* termasuk basis data, antarmuka pengguna, manajemen konten, serta pengelolaan kursus yang memungkinkan pengaturan dan pelacakan kemajuan siswa. Menurut Kumar et al (2021), penerapan arsitektur berbasis *cloud* dalam sistem *e-learning* memungkinkan akses yang lebih cepat dan fleksibel terhadap materi pembelajaran, serta skalabilitas yang lebih baik dalam mengelola jumlah pengguna yang besar. Penelitian lain oleh Al-Zoubi et al (2020) mengemukakan bahwa penggunaan arsitektur berbasis *microservices* dalam *e-learning* memberikan keuntungan dalam hal modularitas dan kemampuan untuk

melakukan pembaruan secara independen pada berbagai komponen sistem tanpa mengganggu kinerja keseluruhan. Di sisi lain, studi oleh Sillanpaa et al (2019) menyoroti pentingnya desain arsitektur yang responsif dan adaptif untuk memastikan pengalaman pembelajaran yang optimal pada berbagai perangkat lunak, baik *desktop* maupun *mobile*.

Dalam konteks pendidikan tinggi, Zhang et al (2022) menemukan bahwa arsitektur *e-learning* yang menggabungkan elemen-elemen gamifikasi dapat meningkatkan keterlibatan dan motivasi mahasiswa dalam pembelajaran online. Selain itu, penelitian oleh Kim & Park (2020) menunjukkan bahwa arsitektur *e-learning* yang mengintegrasikan elemen-elemen analitik pembelajaran dapat membantu dalam memantau kemajuan siswa dan memberikan umpan balik yang lebih personal.

2.3.4 Algoritma Logistic Regression

Logistic Regression merupakan salah satu model regresi linier yang digunakan untuk memodelkan hubungan antara *variabel independen* dan *variabel dependen* dengan skala yang tidak terdefinisi secara linier. Metode ini efektif untuk memprediksi probabilitas terjadinya suatu peristiwa acak berdasarkan variabel prediktor dan banyak diterapkan dalam analisis klasifikasi, terutama ketika variabel dependen bersifat kategorikal (Starbuck, 2023). Metode multivariat ini memodelkan hubungan antara variabel independen dan variabel dependen dengan mempertimbangkan pengaruh simultan dari beberapa variabel prediktor untuk menghasilkan estimasi yang lebih akurat dan komprehensif (Boateng & Abaye, 2019). Banyak peneliti menggunakan metode ini sebagai algoritma pemodelan prediktif untuk memprediksi probabilitas terjadinya suatu peristiwa berdasarkan hubungan antara variabel independen dan variabel dependen (Panda et al., 2022).

Metode *Logistic Regression* melibatkan beberapa tahapan utama dalam proses analisis dan pemodelan data. Tahap pertama dimulai dengan pengumpulan dan praproses data, yang mencakup penganganan data hilang (*missing values*), normalisasi

data, serta pengkodean variabel kategorikal menjadi format numerik agar dapat digunakan dalam pemodelan.

Setelah data dipersiapkan, dataset dibagi menjadi dua bagian, yaitu *training set* dan *testing set*. Pada tahap pelatihan (*training*), model *Logistic Regression* dilatih menggunakan *training set* dengan cara mengestimasi koefisien regresi yang memaksimalkan *likelihood* dari hasil yang diamati. Proses ini menggunakan fungsi logistik (*logistic function*) untuk memodelkan probabilitas terjadinya suatu peristiwa biner berdasarkan variabel independen.

Setelah model berhasil dilatih, ia digunakan untuk memprediksi probabilitas pada *testing set*. *Logistic Regression* menghitung kemungkinan terjadinya suatu peristiwa berdasarkan satu atau lebih variabel prediktor menggunakan rumus dasar berikut:

$$\log\left(\frac{P(Y=1|X)}{1-P(Y=1|X)}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n \quad (2.1)$$

Dengan:

$P(Y = 1|X)$: Probabilitas bahwa Y bernilai 1 diberikan X

β_0 : Intercept (konstanta)

$\beta_1, \beta_2, \dots, \beta_n$: Koefisien regresi yang menentukan pengaruh variabel independen

$X_1, X_2, \dots, X_n$: Variabel independen

Jika $P(Y = 1|X) > 0.5$, maka prediksi cenderung $Y = 1$

Jika $P(Y = 1|X) < 0.5$, maka prediksi cenderung $Y = 0$

2.3.5 Confusion Matrix

Confusion matrix adalah sebuah tabel yang digunakan untuk mengevaluasi performa dari suatu model machine learning. Tabel ini menampilkan jumlah prediksi yang benar dan salah dari model tersebut. *Confusion matrix* terdiri dari empat komponen utama yaitu *true positive* (TP) yaitu prediksi positif yang benar, *true*

negative (TN) adalah prediksi negatif yang benar, *false positive* (FP) artinya prediksi positif yang salah dan yang terakhir *false negative* (FN) yang berarti prediksi negatif yang salah.

Berikut adalah rumus – rumus yang digunakan untuk menghitung mencari nilai dari confusion matrix dari suatu model:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.2)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2.3)$$

$$\text{Recall} = \frac{TP+TN}{TP+FN} \quad (2.4)$$

$$\text{F1 – Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2.5)$$

Dengan:

TP : True Positif

TN : True Negatif

FP : False Positif

FN : False Negatif

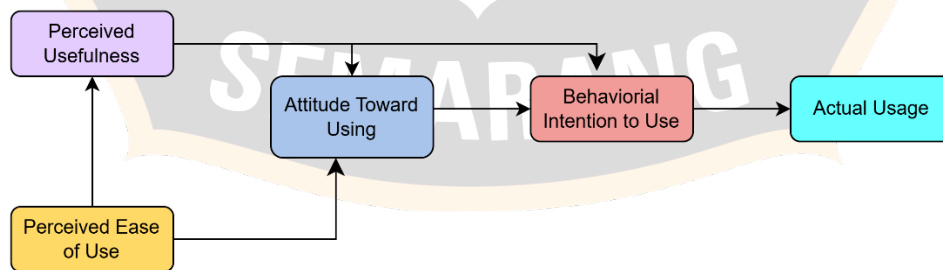
2.3.6 *Technology Acceptance Model* (TAM)

Dalam *technology acceptance model* (TAM), terdapat lima variabel utama yang berperan penting dalam menjelaskan penerimaan teknologi oleh pengguna. Variabel pertama, *perceived ease of use* (persepsi kemudahan penggunaan), mengacu pada sejauh mana pengguna percaya bahwa teknologi dapat mempermudah pekerjaan mereka tanpa membutuhkan banyak usaha. Keyakinan ini sangat memengaruhi keputusan pengguna untuk mencoba dan menggunakan teknologi secara berkelanjutan. Variabel kedua, *perceived usefulness* (persepsi kegunaan), menunjukkan tingkat keyakinan pengguna bahwa penggunaan teknologi akan

meningkatkan kinerja mereka. Jika pengguna merasa teknologi tersebut memberikan manfaat yang signifikan, mereka lebih cenderung untuk mengadopsinya (Davis, 1989).

Selanjutnya, *attitude toward using technology* (sikap terhadap penggunaan teknologi) merupakan perasaan positif atau negatif seseorang terhadap penggunaan teknologi tertentu. Sikap ini dipengaruhi oleh persepsi kemudahan dan kegunaan teknologi tersebut. Variabel keempat, *behavioral intention to use* (keinginan untuk menggunakan teknologi), mencerminkan niat seseorang untuk menggunakan teknologi, yang didorong oleh sikap terhadap teknologi dan keyakinan terhadap manfaat yang dirasakan. Keinginan ini memainkan peran penting dalam menentukan apakah seseorang akan benar-benar menggunakan teknologi tersebut atau tidak. Variabel terakhir, *actual technology usage* (penggunaan teknologi aktual), mengacu pada penggunaan teknologi secara nyata oleh pengguna. Variabel ini secara langsung dipengaruhi oleh *behavioral intention* dan mencerminkan implementasi perilaku dalam konteks penggunaan teknologi (Utami, 2021).

Dalam konteks pembelajaran daring, variabel-variabel TAM memberikan kerangka yang relevan untuk menganalisis faktor-faktor yang memengaruhi penerimaan *platform e-learning*. Persepsi kemudahan penggunaan dan kegunaan e-learning memengaruhi kepuasan siswa terhadap pembelajaran daring, sedangkan sikap positif terhadap teknologi meningkatkan niat mereka untuk menggunakannya secara berkelanjutan (Aznam et al., 2022). Dengan memahami interaksi antarvariabel ini, institusi pendidikan dapat merancang dan mengimplementasikan sistem pembelajaran daring yang lebih efektif dan sesuai dengan kebutuhan pengguna.



Gambar 2. 1 Variabel-Variabel TAM

2.3.7 Model Pembelajaran

Model pembelajaran adalah kerangka konseptual yang digunakan untuk merancang aktivitas pembelajaran secara efektif dan efisien. Menurut Rizal et al. (2024) model pembelajaran kolaboratif, mampu meningkatkan hasil belajar siswa dengan mendorong interaksi aktif, berpikir kritis, dan kreativitas. Selain itu menurut Hanifah & Purbosari (2022), memberikan dampak positif terhadap hasil belajar kognitif, afektif, dan psikomotor siswa dengan melibatkan mereka dalam eksplorasi dan pemecahan masalah.

Mengintegrasikan pembelajaran dengan teknologi dan keterampilan abad 21 untuk mempersiapkan siswa menghadapi tantangan global (Samala et al., 2022). Dengan penerapan model yang tepat, kualitas pembelajaran dapat ditingkatkan secara signifikan, sejalan dengan kebutuhan dan tuntutan zaman.

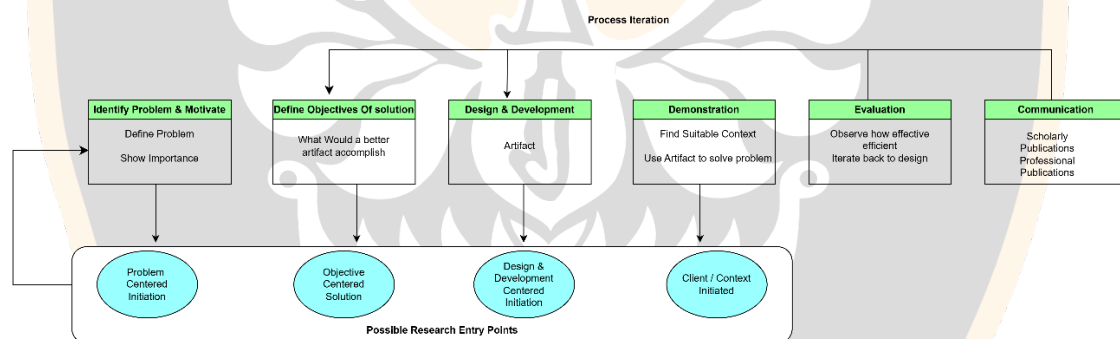
2.3.8 *Desain Science Research Methodology (DSRM)*

Desain Science Research (DSR) adalah paradigma penelitian yang berfokus pada pemecahan masalah melalui pengembangan artefak inovatif yang bertujuan untuk meningkatkan pengetahuan manusia, baik dalam bidang teknologi maupun basis pengetahuan ilmiah. Paradigman ini menghasilkan artefak baru dan pengetahuan desain (*design knowledge*) yang relevan untuk meningkatkan aplikasi pada konteks tertentu (vom Brocke et al., 2020). *Design Science Research Methodology (DSRM)* digunakan sebagai pendekatan sistematis yang mengintegrasikan prinsip, praktik, dan prosedur untuk melaksanakan penelitian. Metodologi ini bertujuan untuk mencapai tiga hal : (1) konsistensi dengan literatur yang relevan, (2) penyediaan proses penelitian nominal, dan (3) pembentukan model mental yang dapat digunakan untuk menyajikan dan mengevaluasi penelitian dalam bidang sistem informasi (Esearch et al., 2004).

Design science research (DSR) menjadi jembatan antara pengembangan artefak inovatif dan kontribusi teoritis yang mendalam, dengan memberikan

pemahaman lebih lanjut mengenai konteks aplikasi relevan (Gregor et al., 2020). Dalam beberapa tahun terakhir, *Design Science Research* (DSR) telah semakin diakui sebagai pendekatan yang signifikan dalam penelitian yang signifikan dalam bidang sistem informasi (Oruç et al., 2022; Peffers et al., 2018). Pendekatan ini memberikan sebuah kerangka yang memungkinkan peneliti untuk mengembangkan solusi inovatif atas masalah praktis, sekaligus memperkaya pengetahuan yang sudah ada.

Penelitian ini mengaplikasikan *Design Science Research* untuk mencari solusi terhadap permasalahan yang dihadapi oleh institusi, khususnya dalam konteks lembaga pendidikan. Penelitian dimulai dengan mengidentifikasi isu utama yang perlu diselesaikan, sebagaimana dijelaskan pada latar belakang. Selanjutnya, proses penelitian ini mengikuti serangkaian langkah terstruktur yang mencakup identifikasi masalah, penetapan tujuan solusi, perancangan dan pengembangan artefak, demonstrasi, evaluasi, dan komunikasi hasil (Peffers et al., 2007). Setiap tahap dilakukan untuk memastikan bahwa solusi yang ditemukan tidak hanya aplikatif dalam praktik, tetapi juga memenuhi standar akademik yang ada.



Gambar 2. 2 Design Science Research Methodology

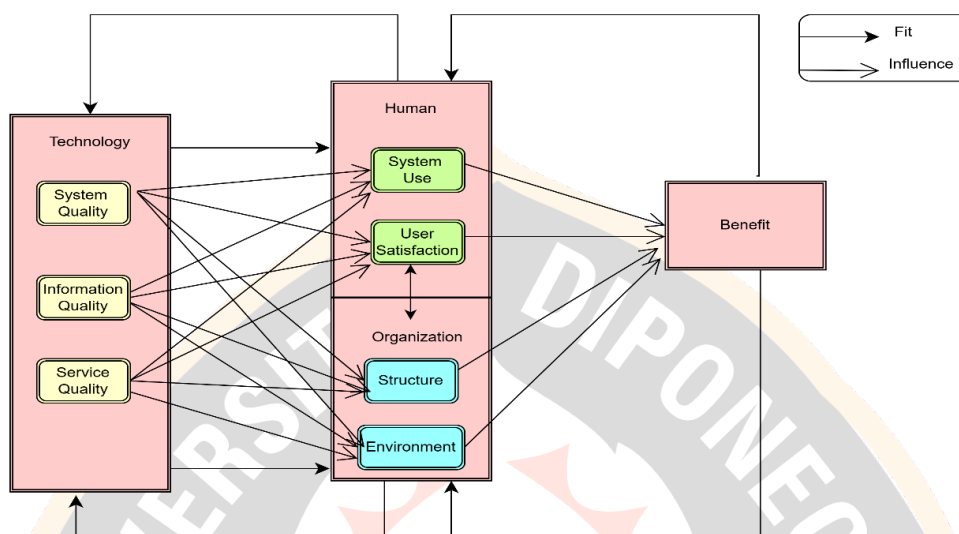
2.3.9 Model Human, Organization, Technology and Benefit (HOT-Fit)

Model *Human-Organization-Technology and Benefit* (HOT-Fit) merupakan kerangka kerja evaluasi yang dirancang untuk menilai kesesuaian antara manusia, organisasi, dan teknologi dalam implementasi sistem informasi. Dikembangkan oleh

Yusof et al. (2008), model ini menawarkan pendekatan holistik yang mampu mengintegrasikan tiga komponen utama untuk mengevaluasi keberhasilan suatu sistem. Salah satu keunggulan utama HOT-Fit adalah kemampuannya untuk mengakomodasi variabel struktur organisasi dan lingkungan organisasi, yang jarang ditemukan dalam model evaluasi lainnya (Erlirianto et al., 2015). Selain itu, kepuasan pengguna (*user satisfaction*) menjadi salah satu variabel *dependen* yang paling penting dalam menentukan tingkat keberhasilan implementasi sistem informasi, termasuk pada sistem *e-learning* (Kodoati & Hartomo, 2022).

Dalam penerapannya, HOT-Fit model memusatkan evaluasi pada tiga aspek utama, yaitu manusia (*human*), organisasi (*organization*), dan teknologi (*technology*). Komponen manusia mencakup kompetensi, kepuasan, serta tingkat penerimaan pengguna terhadap sistem. Komponen organisasi menitikberatkan pada dukungan manajemen, kebijakan dan prosedur, serta kesesuaian budaya organisasi dalam mendukung implementasi teknologi. Sedangkan komponen teknologi meliputi kualitas sistem, kualitas informasi, dan kualitas layanan yang disediakan oleh sistem. Interaksi antara ketiga komponen ini memungkinkan evaluasi yang menyeluruh untuk memastikan keberhasilan implementasi sistem.

Dalam konteks *e-learning*, HOT-Fit sering digunakan untuk menilai bagaimana teknologi dapat mendukung proses pembelajaran daring secara optimal. Model ini membantu mengidentifikasi kekuatan dan kelemahan dalam sistem yang diimplementasikan, sehingga dapat memberikan dasar bagi perbaikan yang berkelanjutan. Dengan pendekatan yang terstruktur dan komprehensif, HOT-Fit menjadi alat evaluasi yang efektif dalam memastikan sistem informasi tidak hanya berfungsi secara teknis, tetapi juga mampu mendukung tujuan organisasi secara menyeluruh.



Gambar 2. 3 Human, Organization, Technology and Benefit (Hot-Fit)

Gambar 2.3 adalah menggambarkan komponen dari HOT-Fit yang terdiri dari teknologi, human, organisasi dan net benefit. Pada komponen teknologi terdiri dari 3 variabel yaitu *System Quality* (kualitas sistem) untuk mengukur reliabilitas, kemudahan penggunaan, dan performa sistem. *High system quality* mendorong *system use* dan *user satisfaction*. *Information quality* (kualitas informasi) menilai akurasi, relevansi, dan kemitakhiran data yang disajikan. Berkontribusi terhadap *system use* dan *user satisfaction*. *Service quality* (kualitas layanan) meliputi dukungan teknis, pelatihan, dan respons layanan. Mempengaruhi *system use* dan *user satisfaction*.

Komponen berikutnya adalah human yang terdiri dari 2 variabel yaitu *System Use* (penggunaan sistem) mewakili intensitas dan frekuensi penggunaan sistem. Memiliki pengaruh langsung terhadap *net benefit*. *User Satisfaction* (kepuasan pengguna) tingkat kepuasan pengguna terhadap sistem. Merupakan determinan kuat terhadap *net benefit*.

Selanjutnya adalah organisasi yang terdiri dari 2 variabel yaitu *structure* (struktur organisasi) menilai dukungan manajemen, kebijakan teknologi informasi dan koordinasi internal. Memberikan kontribusi terhadap *net benefit*. *Environment* (lingkungan organisasi) meliputi sarana pendukung, infrastruktur, dan budaya kerja.

Juga berpengaruh pada net benefit. Terakhir adalah komponen net benefit dihasilkan dari interaksi antara penggunaan sistem dan kepuasan pengguna. System Use dan User Satisfaction kemudian kembali mempengaruhi satu sama lain dalam umpan balik terhadap perspektif keberlanjutan sistem.

2.3.10 *Two Way ANOVA*

Salah satu teknik statistik yang umum digunakan dalam penelitian adalah analisis varian atau Anova. Anova digunakan untuk membandingkan rata-rata kelompok yang berbeda untuk satu atau lebih variabel respon dalam suatu populasi. Tujuan anova adalah untuk mengukur seberapa signifikan perbedaan antara rata-rata kelompok yang berbeda dalam suatu populasi. *Two-way Anova* digunakan ketika terdapat dua faktor atau lebih yang mempengaruhi variabel respon. Langkah-langkah dalam perhitungan ANOVA mencakup:

- Identifikasi nilai: t (jumlah perlakuan), r (jumlah blok)
- Hitung jumlah pengamatan total (n), yaitu: $n=r * t$
- Hitung jumlah kuadrat total dengan rumus:

$$SS_T = \sum (X_{ij})^2 - \frac{(\sum T_j)^2}{n} \quad (2.7)$$

- Hitung jumlah kuadrat perlakuan dengan rumus:

$$SS_p = \sum \frac{\sum (P_1)^2}{r} - \frac{(\sum T_j)^2}{n} \quad (2.8)$$

- Hitung jumlah kuadrat antar blok dengan rumus:

$$SS_B = \sum \frac{\sum (B_1)^2}{t} - \frac{(\sum T_j)^2}{n} \quad (2.9)$$

- Cari nilai F-Hitung dengan menggunakan rumus yang tertera pada tabel berikut:

Tabel 2. 3. Rumus F-hitung

Sources Of Variation	Df	SS	MS	F-Hitung
Between Blocks	$r-1$	SS_B	$MS_B = \frac{SS_B}{r-1}$	$\frac{MS_B}{MS_E}$
Between Treatments	$t-1$	SS_p	$MS_p = \frac{SS_p}{t-1}$	$\frac{MS_p}{MS_E}$
In Treatment (error)	$(n-1) - (k-1) - (t-1)$	$SS_E = SS_T - SS_B - SS_p$	$\frac{SS_E}{(r-1) - (t-1)}$	
Total	$n-1$	SS_T		

- g. Cari nilai F-Hitung dengan mempertimbangkan (1) tingkat signifikansi (α), (2) df1 yaitu df dari MS terbesar, dan (3) df2 yaitu df dari MS terkecil.
- h. Bandingkan nilai F-Hitung dengan F table.
 - (1) Bila F Hitung < F table, maka H_0 diterima, yang berarti rata-rata kedua perlakuan tidak berbeda secara signifikan,
 - (2) Bila F Hitung > F table, maka H_0 ditolak dan H_1 diterima, yang berarti rata-rata kedua perlakuan berbeda secara signifikan

2.3.11 Usability Testing

Usability testing merupakan pendekatan evaluasi yang bertujuan untuk mengukur tingkat efektivitas, efisiensi, dan kepuasan pengguna dalam menggunakan suatu sistem atau aplikasi. Proses ini penting dalam memastikan bahwa antarmuka yang dirancang tidak hanya memenuhi kebutuhan teknis tetapi juga memberikan pengalaman pengguna yang optimal. Dalam penelitian oleh Subiyakto et al. (2021), usability testing pada situs repositori institusi pendidikan tinggi menunjukkan bahwa perbaikan antarmuka mengurangi tingkat *bounce rate* dan meningkatkan kepuasan pengguna. Studi ini menyoroti pentingnya melibatkan pengguna dalam evaluasi untuk mengidentifikasi permasalahan yang dialami, seperti navigasi yang membingungkan dan hasil pencarian yang tidak relevan.

Selain itu, penelitian oleh Jiménez et al. (2017) menekankan pentingnya evaluasi heuristik yang disesuaikan untuk meningkatkan kualitas pengalaman pengguna. *Usability testing* untuk mengevaluasi kepuasan pengguna menunjukkan bahwa faktor kemudahan navigasi dan keamanan berkontribusi signifikan terhadap tingkat kepuasan pengguna (Pitri & Abdillah, 2022).

Penelitian-penelitian ini menunjukkan bahwa *usability testing* tidak hanya membantu mengidentifikasi kelemahan sistem tetapi juga memberikan panduan strategies untuk pengembangan teknologi yang lebih responsif terhadap kebutuhan pengguna. Dengan menerapkan metodologi yang terstruktur dan melibatkan partisipasi langsung pengguna, *usability testing* menjadi alat penting dalam menciptakan sistem yang lebih intuitif dan memuaskan.

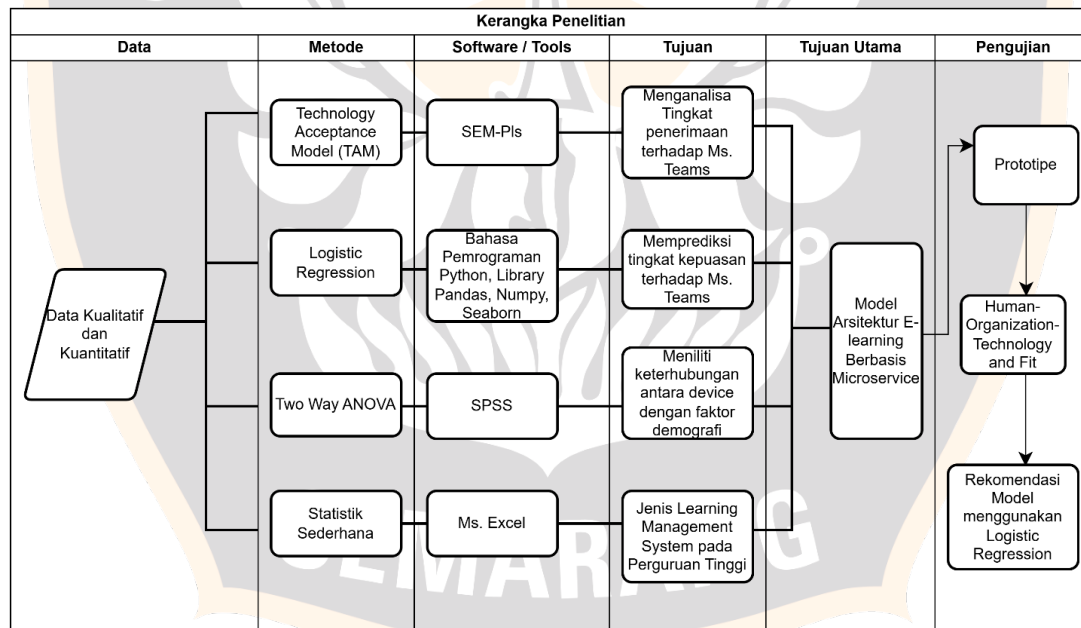
2.3.12 Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah bahasa standar yang digunakan untuk merancang dan memodelkan sistem perangkat lunak. UML menyediakan berbagai jenis diagram yang digunakan untuk menggambarkan berbagai aspek sistem, termasuk statis dan perilaku dinamisnya. UML diagram adalah standar yang dapat diterima untuk menggambarkan model desain berorientasi objek sehingga mempermudah untuk mengidentifikasi persyaratan, ruang lingkup sistem dan aplikasi dengan menyediakan model visual (Koç et al., 2021). Memberikan dasar matematis untuk memastikan konsistensi semantik dari berbagai diagram UML. Hal ini memungkinkan pengembangan alat bantu yang mendukung validasi dan verifikasi sistem secara sistematis (Evans et al., 1999). Selain itu, Wang et al. (2023) mengembangkan kerangka kerja berbasis algebra yang mengintegrasikan struktur dan perilaku UML. Pendekatan ini memungkinkan representasi yang lebih terpadu dari berbagai aspek sistem, sehingga meningkatkan efisiensi pengembangan perangkat lunak.

Penelitian ini mendukung relevansi UML sebagai alat penting dalam rekayasa perangkat lunak, baik dalam hal visualisasi sistem yang kompleks maupun dalam memastikan keselarasan antara berbagai elemen desain. Dengan demikian, UML tetap menjadi pilihan utama untuk mendukung kebutuhan komunikasi, desain, dan dokumentasi dalam proyek pengembangan perangkat lunak modern.

2.3.13 Kerangka pemikiran

Kerangka penelitian disusun dengan menggunakan pendekatan *design science research methodology* (DSRM) untuk memberikan arah dan fokus yang jelas dalam mencapai tujuan penelitian. Kerangka ini menghubungkan teori, metode, dan data secara sistematis, mempermudah analisis, menguji hipotesis, serta memastikan penelitian berjalan efektif dan sesuai dengan tujuan yang ditetapkan. Berikut adalah usulan kerangka pemikiran dalam penelitian ini.



Gambar 2. 4. Kerangka Penelitian Model Arsitektur Sistem E-Learning Berbasis Microservices di Perguruan Tinggi

Gambar 2.4 di atas menunjukkan kerangka penelitian yang mengintegrasikan berbagai metode dan tools untuk mencapai penelitian terkait pengembangan model arsitektur *e-learning* berbasis *microservices*. Penelitian ini memanfaatkan data kualitatif dan kuantitatif untuk mendapatkan gambaran yang komprehensif.

Beberapa metode yang dipilih antara lain *Technology Acceptance Model* (TAM) menggunakan SEM-Pls untuk menganalisis tingkat penerimaan terhadap *Microsoft Teams*. Metode ini dipilih karena TAM efektif dalam mengukur adopsi teknologi oleh pengguna, sementara SEM-Pls mampu menangani model yang kompleks dengan ukuran sampel kecil hingga menengah.

Untuk memprediksi tingkat kepuasan terhadap *Microsoft Teams*, penelitian menggunakan *Logistic Regression* dengan dukungan bahasa pemrograman *Python* dan *library* seperti *Pandas*, *Numpy*, dan *Seaborn*. *Logistic Regression* dipilih karena mampu memodelkan hubungan antara variabel *independen* dan *dependen* kategorikal, sementara *Python* dan *library* terkait memberikan fleksibilitas dalam pengolahan dan visualisasi data.

Metode *Two Way ANOVA* digunakan dengan bantuan SPSS untuk meneliti keterhubungan antara jenis perangkat yang digunakan dengan faktor demografi. *Two Way ANOVA* cocok untuk menguji pengaruh dua variabel *independen* terhadap satu variabel *dependen* sekaligus melihat interaksinya. SPSS dipilih karena kemampuannya dalam mengelola dan menganalisis data statistik dengan antarmuka yang ramah pengguna.

Analisis statistik sederhana menggunakan *Microsoft Excel* juga dimanfaatkan untuk mengidentifikasi jenis *Learning Management System* (LMS) yang digunakan di perguruan tinggi. *Microsoft Excel* dipilih karena kemudahan dalam pengolahan data sederhana dan visualisasi awal.

Tujuan utama penelitian ini adalah mengembangkan model arsitektur *e-learning* berbasis *microservices* yang dievaluasi dengan pendekatan *Human-Organization-Technology and Benefit (HOT-Fit)*. Dari sisi arsitektur *microservices*nya terbagi menjadi 3 pengujian yang pertama dengan usability testing untuk mengukur

kemudahan penggunaan rancangan desain antarmuka dari prototipe menggunakan tools maze.design untuk mahasiswa dan dosen, system usability scale (SUS) untuk admin program studi dan IT support perguruan tinggi, kinerja website Skuring.com menggunakan GTmetrix dan menggunakan K6 untuk mengukur performa, skalabilitas, reliabilitas serta fleksibilitas dari arsitektur yang dihasilkan. Dan untuk mendapatkan rekomendasi prototipe sebagai alternatif pembelajaran daring menggunakan *Logistic Regression* untuk memastikan model yang dihasilkan relevan dan efektif.

