

**RANCANG *BANGUN LEARNING MANAGEMENT SYSTEM*
BERBASIS *WEB* DAN *MICROSERVICES***

SKRIPSI



Muhammad Alif Setya Prakasa

H071191040

PROGRAM STUDI SISTEM INFORMASI

DEPARTEMEN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS HASANUDDIN

MAKASSAR

APRIL 2024

**RANCANG BANGUN *LEARNING MANAGEMENT SYSTEM*
BERBASIS *WEB* DAN *MICROSERVICES***

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana
Komputer pada Program Studi Sistem Informasi Departemen Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin
Makassar**

MUHAMMAD ALIF SETYA PRAKASA

H071191040

**PROGRAM STUDI SISTEM INFORMASI
DEPARTEMEN MATEMATIKA**

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS HASANUDDIN

MAKASSAR

APRIL 2024

LEMBAR PERNYATAAN KEOTENTIKAN

Saya yang bertanda tangan di bawah ini menyatakan dengan sungguh-sungguh bahwa skripsi yang saya buat dengan judul:

**RANCANG BANGUN LEARNING MANAGEMENT SYSTEM BERBASIS
WEB DAN MICROSERVICES**

adalah benar hasil karya saya sendiri, bukan hasil plagiat dan belum pernah dipublikasikan dalam bentuk apapun

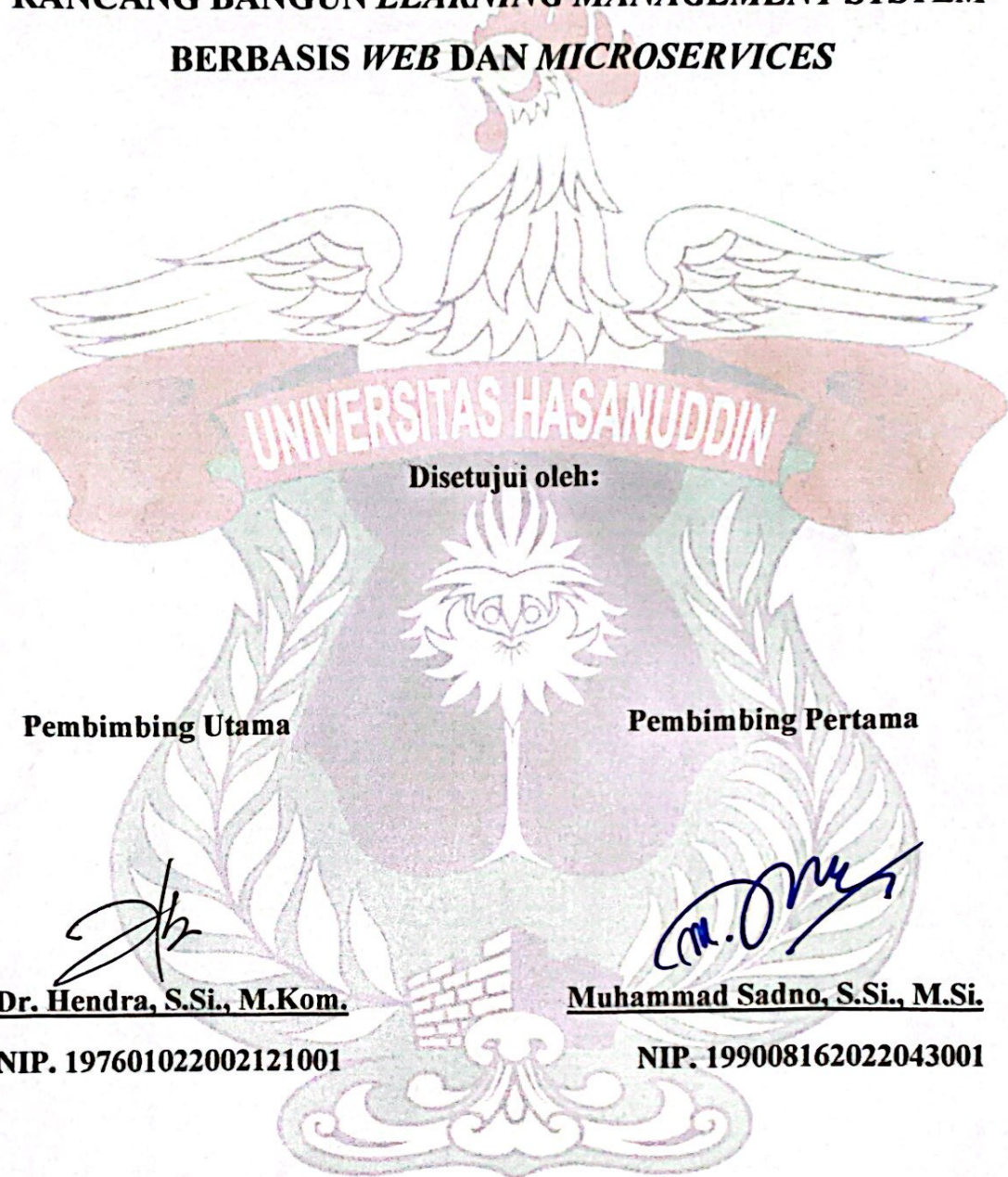
Makassar, 1 Juni 2024



Muhammad Alif Setya Prakasa

NIM. H071191040

RANCANG BANGUN *LEARNING MANAGEMENT SYSTEM* BERBASIS *WEB* DAN *MICROSERVICES*



Disetujui oleh:

Pembimbing Utama

Pembimbing Pertama


Dr. Hendra, S.Si., M.Kom.

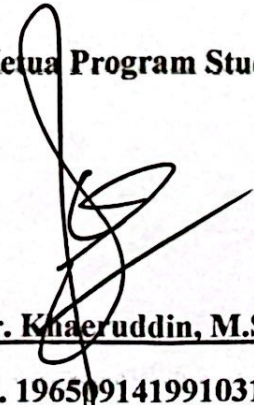
NIP. 197601022002121001


Muhammad Sadno, S.Si., M.Si.

NIP. 199008162022043001

Ketua Program Studi




Dr. Khaeruddin, M.Sc.
NIP. 196509141991031003

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh

Nama : Muhammad Alif Setya Prakasa
NIM : H071191040
Program Studi : Sistem Informasi
Judul Skripsi : RANCANG BANGUN LEARNING MANAGEMENT
SYSTEM BERBASIS WEB DAN MICROSERVICES

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer pada Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin

DEWAN PENGUJI

		Tanda Tangan
1. Ketua	: Dr. Hendra, S.Si., M.Kom.	(.....)
2. Sekretaris	: Muhammad Sadno, S.Si., M.Si.	(.....)
3. Anggota	: Rozalina Amran, S.T., M.Eng.	(.....)
4. Anggota	: Riskawati, S.Si., M.Si.	(.....)

Ditetapkan di : Makassar

Tanggal : 13 Mei 2024

KATA PENGANTAR

Puji Syukur penulis panjatkan ke hadirat Tuhan yang Maha Esa atas segala limpahan Rahmat dan karunia-Nya, yang telah memberikan kesempatan dan kelancaran bagi penulis dalam penyelesaian tugas akhir yang berjudul "RANCANG BANGUN *LEARNING MANAGEMENT SYSTEM* BERBASIS *WEB* DAN *MICROSERVICES* " ini. Dengan berbagai rintangan yang dihadapi saat menyelesaikan tugas ini, tidak lupa untuk penulis mengucapkan terima kasih atas kontribusi dan bantuannya kepada:

1. Rektor Universitas Hasanuddin, **Bapak Prof. Dr. Ir. Jamaluddin Jompa, M.Sc.** beserta jajarannya.
2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam, **Dr. Eng. Amiruddin** beserta jajarannya.
3. Ketua Departemen Matematika Fakultas Matematika Dan Ilmu Pengetahuan Alam Universitas Hasanuddin, **Bapak Prof. Dr. Nurdin, S.Si., M.Si.**, atas seluruh ilmu dan saran-saran yang telah diberikan.
4. Ketua Program Studi Sistem Informasi, Bapak **Dr. Khaeruddin, M.Sc.** atas seluruh ilmu dan saran-saran yang telah diberikan.
5. Ketua Program Studi Sistem Informasi periode sebelumnya, Bapak **Dr. Hendra, S.Si., M.Kom.** yang telah senantiasa membantu dan memberikan arahan selama masa studi penulis.
6. Pembimbing Utama penulis sekaligus Penasehat Akademik Penulis, Bapak **Dr. Hendra, S.Si., M.Kom.** yang telah senantiasa membantu, membimbing, dan memberikan arahan selama masa studi penulis khususnya dalam masa penyusunan skripsi.
7. Pembimbing Pertama penulis **Bapak Muhammad Sadno, S.Si., M.Si.** yang telah senantiasa membantu, membimbing, dan memberikan arahan selama masa studi penulis khususnya dalam masa penyusunan skripsi ini.
8. Kedua dosen penguji, ibu **Rozalina Amran, S.T., M.Eng.**, dan ibu **Riskawati, S.Si., M.Si.**, yang telah memberikan kritik dan masukan yang

bermanfaat dalam penelitian tugas akhir ini sehingga oleh karenanya skripsi ini dapat tersusun dengan baik.

9. Bapak/Ibu **Dosen Program Studi Sistem Informasi** beserta seluruh tenaga pendidik yang telah memberikan ilmu dan mendidik penulis selama masa perkuliahan. Serta kepada seluruh staf dan pegawai **Departemen Matematika** yang telah membantu penulis terutama dalam segala proses administrasi.
10. Kedua orang tua penulis, Bapak **Jumardi** dan Ibu **Ermila Prabumuli** yang tidak pernah lelah mendoakan, memotivasi, dan memberikan dukungan moral dan materi.
11. Seluruh teman-teman program studi **Sistem Informasi Angkatan 2019** khususnya **Andi Ilhamsyah, Muhammad Ikhsan, Bayu Ajid, Eurico Devon, Muhamad Fajri, Fatwa Anugrah, Fauzi Asham, Muhammad Takdim, Muammar Ahlan, Rafly Fatur, Rafly Masloman, Richard Enrico, Silverius Sony, Sultan, Taufiq Goe, Theodarryl, Yusuf Syam, dan (alm) Aso Ahmad Amin Rais** yang senantiasa memberikan bantuan dan dukungan selama perkuliahan hingga selesainya penulisan skripsi penulis.
12. Teman-teman **KKNT 110 Pengembangan UMKM Bantaeng Posko 9** yang selalu memberikan dukungan dan motivasi kepada penulis dalam menyelesaikan skripsi penulis.
13. Teman-teman Kartjis.Id, **Reski Anugrah, Nizar Sabri** yang telah memberikan keleluasaan dan dukungan bagi penulis dalam mengerjakan skripsi sambil membangun usaha bersama.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna dikarenakan terbatasnya pengalaman dan pengetahuan yang dimiliki penulis. Oleh karena itu, penulis mengharapkan segala bentuk saran serta masukan bahkan kritik yang membangun dari berbagai pihak. Semoga tulisan ini memberikan manfaat kepada semua pihak yang membutuhkan dan terutama untuk penulis.

Makassar, 1 Juni 2024

A handwritten signature in blue ink, appearing to be 'Muhammad Alif', with a stylized, cursive script.

Muhammad Alif

Setya Prakasa

PERSYARATAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai civitas akademik Universitas Hasanuddin, saya yang bertanda tangan dibawah ini :

Nama : Muhammad Alif Setya Prakasa
NIM : H071191040
Program Studi : Sistem Informasi
Departemen : Matematika
Fakultas : Fakultas Matematika dan Ilmu Pengetahuan Alam
Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Hasanuddin **Hak Prediktor Royalti Noneklusif** (*Non-exclusive Royalty-Free Right*) atas tugas akhir saya yang berjudul:

“RANCANG BANGUN LEARNING MANAGEMENT SYSTEM BERBASIS WEB DAN MICROSERVICES”

beserta perangkat yang ada (jika diperlukan). Terkait dengan hal di atas, maka pihak Universitas Hasanuddin berhak menyimpan, mengalih-media/format-kan, mengola dalam bentuk pangkalan data (*database*), merawat dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Dibuat di Makassar pada 1 Juni 2024

Yang menyatakan



Muhammad Alif Setya Prakasa

ABSTRAK

Learning Management System (LMS) merupakan sebuah platform yang penting dalam pendidikan modern untuk mengelola, menyampaikan, dan menilai materi pembelajaran secara efektif. Namun, pengembangan LMS tradisional sering kali menghadapi kendala terkait skalabilitas, fleksibilitas, dan kinerja. Oleh karena itu, penelitian ini bertujuan untuk merancang dan membangun sebuah LMS yang berbasis *web* dan menggunakan arsitektur *microservices* guna mengatasi tantangan tersebut. Metode yang digunakan dalam penelitian ini mencakup analisis kebutuhan fungsional, perancangan arsitektur *microservices*, pemilihan teknologi, implementasi, dan pengujian sistem. Melalui pendekatan ini, LMS yang dihasilkan dapat dengan mudah diperluas, dikonfigurasi, dan diintegrasikan dengan sistem lain. Hasil dari penelitian ini adalah sebuah LMS yang dinamakan *learnyscape* yang memiliki independensi dari segi sistem nya. .Pengujian sistem dengan menggunakan *Black Box Testing* menunjukkan bahwa aplikasi telah memenuhi seluruh kebutuhan fungsional. Selain itu mengetahui kepuasan pengalaman pengguna dilakukan pengujian *User Acceptance Test*, pada *user admin* diperoleh skor sebesar 83.3%, pengujian terhadap *user lecturer* diperoleh skor sebesar 80%.

Kata Kunci : *learning management system, microservice, black box testing, User Acceptance Test*

ABSTRACT

The Learning Management System (LMS) serves as a pivotal platform in modern education, facilitating the management, delivery, and assessment of learning materials effectively. However, traditional LMS development often encounters challenges related to scalability, flexibility, and performance. Therefore, this research aims to design and develop a web-based LMS utilizing microservices architecture to address these challenges. The methodology employed in this research includes functional requirement analysis, microservices architecture design, technology selection, implementation, and system testing. Through this approach, the resulting LMS can be easily expanded, configured, and integrated with other systems. The outcome of this research is a LMS named "learnyscape" that exhibits system independence. System testing using Black Box Testing demonstrates that the application meets all functional requirements. Additionally, User Acceptance Testing was conducted to assess user satisfaction. For admin users, a satisfaction score of 83.3% was achieved, while for lecturer users, a score of 80% was obtained. This research provides a significant contribution to educational system development by presenting a novel approach in designing LMS that leverages the advantages of microservices architecture and web technologies.

Keywords : *learning management system, microservice, black box testing, User Acceptance Test,*

DAFTAR ISI

HALAMAN JUDUL	ii
LEMBAR PERNYATAAN KEOTENTIKAN	iii
HALAMAN PENGESAHAN	v
KATA PENGANTAR.....	vi
ABSTRAK	x
ABSTRACT	xi
DAFTAR ISI.....	xii
DAFTAR GAMBAR.....	xv
DAFTAR TABEL	xviii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	5
BAB II TINJAUAN PUSTAKA.....	6
2.1 Sistem Informasi.....	6
2.2 <i>Framework</i> Penelitian Sistem Informasi	7
2.3 <i>Learning management system (LMS)</i>	7
2.4 React.....	8
2.5 NodeJS.....	9
2.6 <i>Express js</i>	10
2.7 <i>Database</i>	10
2.8 <i>RDBMS MySQL</i>	12
2.9 <i>Black Box Testing</i>	12

2.10	REST API.....	13
2.11	<i>Microservice</i>	14
2.12	<i>Domain Driven Design</i>	15
2.13	Penelitian Relevan	16
BAB III METODE PENELITIAN		21
3.1	Waktu dan Tempat	21
3.2	Tahap Penelitian	21
3.2.1	Analisis Kebutuhan	22
3.2.2	Desain Sistem.....	22
3.2.3	Implementasi Sistem	22
3.2.4	Pengujian Sistem.....	22
3.2.5	Hasil dan Pemeliharaan.....	22
3.3	Desain <i>Science</i> Penelitian.....	22
3.4	Alur Penelitian.....	23
3.5	Arsitektur Sistem	24
3.5.1	<i>Use Case Diagram</i>	25
3.5.2	<i>System Architecture</i>	26
3.6	Perancangan User Interface	27
3.6.1	Halaman <i>Login</i>	27
3.6.2	Halaman Daftar <i>User</i> Mahasiswa dan Dosen	28
3.6.3	Halaman Tambah <i>User</i>	29
3.6.4	Halaman Daftar Mata Kuliah.....	30
3.6.5	Halaman Tambah Mata Kuliah	31
3.6.6	Halaman Daftar Kelas	32
3.6.7	Halaman Tambah Daftar Kelas.....	33
3.6.8	Halaman Tambah Dosen Ke Kelas	34
3.6.9	Halaman Daftar Kelas Dosen.....	35
3.6.10	Halaman Daftar Tugas	36
3.6.11	Halaman Tambah Tugas	37
3.6.12	Halaman Daftar Kuis.....	38
3.6.13	Halaman Tambah Kuis.....	39
3.6.14	Halaman Tambah Soal Kuis	40

3.6.15	Halaman Detail Kuis	41
3.6.16	Halaman Detail Jawaban Kuis	42
3.6.17	Halaman <i>Feedback</i> Kuis	43
3.6.18	Halaman Daftar Materi	44
3.6.19	Halaman Tambah Materi.....	45
BAB IV HASIL DAN PEMBAHASAN		46
4.1	Implementasi Sistem	46
4.2	Implementasi <i>Database</i>	46
4.2.1	<i>Entity Relationship Diagram</i> (ERD).....	46
4.2.2	Struktur Tabel.....	47
4.2.3	Relasi Antar Tabel.....	57
4.3	Implementasi <i>Activity Diagram</i>	58
4.3.1	<i>Activity Diagram Login</i>	58
4.3.2	<i>Activity Diagram Admin</i>	59
4.3.3	<i>Activity Diagram Lecturer</i>	68
4.4	Implementasi <i>UI/UX</i>	77
4.4.1	Halaman <i>Login</i>	78
4.4.2	Halaman <i>Admin</i>	78
4.4.3	Halaman <i>Lecturer</i>	83
4.5	Pengujian Sistem	89
4.5.1	<i>Black Box Testing</i>	89
4.5.2	<i>User Acceptance Test</i>	122
BAB V KESIMPULAN DAN SARAN		126
5.1	Kesimpulan.....	126
5.2	Saran	126
DAFTAR PUSTAKA		128

DAFTAR GAMBAR

Gambar 2.1 Kerangka Penelitian Sistem Informasi	7
Gambar 2.2 Komponen <i>Database</i>	11
Gambar 2.3 Ilustrasi REST API	14
Gambar 2.4 Ilustrasi Arsitektur <i>microservice</i>	15
Gambar 3.1 Desain <i>Science</i> Penelitian	23
Gambar 3.2 Alur Penelitian	24
Gambar 3.3 <i>Use Case Diagram</i> Admin dan Dosen	25
Gambar 3.4 <i>System Architecture Learnyscape</i>	26
Gambar 3.5 <i>Domain Driven Design Learnyscape</i>	26
Gambar 3.6 Halaman <i>login</i>	27
Gambar 3.7 Halaman Daftar <i>User</i> Mahasiswa dan Dosen	28
Gambar 3.8 Halaman Tambah <i>User</i>	29
Gambar 3.9 Halaman Daftar Mata Kuliah	30
Gambar 3.10 Halaman Tambah Mata Kuliah	31
Gambar 3.11 Halaman Daftar Kelas	32
Gambar 3.12 Halaman Tambah Kelas	33
Gambar 3.13 Halaman Tambah Dosen Ke Kelas	34
Gambar 3.14 Halaman Daftar Kelas Dosen	35
Gambar 3.15 Halaman Daftar Tugas	36
Gambar 3.16 Halaman Tambah Tugas	37
Gambar 3.17 Halaman Daftar Kuis	38
Gambar 3.18 Halaman Tambah Kuis	39
Gambar 3.19 Halaman Tambah Soal Kuis	40
Gambar 3.20 Halaman Detail Kuis	41
Gambar 3.21 Halaman Detail Jawaban Kuis	42
Gambar 3.22 Halaman <i>Feedback</i> Kuis	43
Gambar 3.23 Halaman Daftar Materi	44
Gambar 3.24 Halaman Tambah Materi	45

Gambar 4.1 ERD <i>Learnyscape</i>	47
Gambar 4.2 Relasi Antar Tabel.....	58
Gambar 4.3 <i>Activity Diagram Login</i>	59
Gambar 4.4 <i>Activity Diagram</i> Tambah Mata Kuliah	60
Gambar 4.5 <i>Activity Diagram Edit</i> Mata Kuliah	61
Gambar 4.6 <i>Activity Diagram</i> Hapus Mata Kuliah.....	62
Gambar 4.7 <i>Activity Diagram</i> Tambah user	63
Gambar 4.8 <i>Activity Diagram Edit User</i>	64
Gambar 4.9 <i>Activity Diagram</i> Hapus user	65
Gambar 4.10 <i>Activity Diagram</i> Tambah Kelas.....	66
Gambar 4.11 <i>Activity Diagram Edit</i> Kelas	66
Gambar 4.12 <i>Activity Diagram</i> Hapus Kelas.....	67
Gambar 4.13 <i>Activity Diagram</i> Menugaskan Dosen Pada Kelas.....	67
Gambar 4.14 <i>Activity Diagram Upload</i> Tugas	68
Gambar 4.15 <i>Activity Diagram Edit</i> Tugas.....	69
Gambar 4.16 <i>Activity Diagram</i> Hapus Tugas	69
Gambar 4.17 <i>Activity Diagram Student's Task Submissions</i>	70
Gambar 4.18 <i>Activity Diagram Upload</i> Materi	71
Gambar 4.19 <i>Activity Diagram Edit</i> Materi.....	71
Gambar 4.20 <i>Activity Diagram</i> Hapus Materi	72
Gambar 4.21 <i>Activity Diagram Upload</i> Kuis.....	73
Gambar 4.22 <i>Activity Diagram Edit</i> Kuis.....	73
Gambar 4.23 <i>Activity Diagram</i> Hapus Kuis	74
Gambar 4.24 <i>Activity Diagram</i> Tambah Soal Kuis	75
Gambar 4.25 <i>Activity Diagram Student's Quiz Answers</i>	76
Gambar 4.26 <i>Activity Diagram</i> Memberikan <i>Feedback</i> Pada Kuis.....	77
Gambar 4.27 Implementasi Halaman <i>Login</i>	78
Gambar 4.28 Implementasi Halaman Daftar <i>User Student</i> dan <i>Lecturer</i>	79
Gambar 4.29 Implementasi Halaman Tambah <i>User</i>	79
Gambar 4.30 Implementasi Halaman Daftar Mata Kuliah	80
Gambar 4.31 Implementasi Halaman Tambah Mata Kuliah	81

Gambar 4.32 Implementasi Halaman Daftar Kelas	81
Gambar 4.33 Implementasi Halaman Tambah Kelas	82
Gambar 4.34 Implementasi Halaman Tambah Dosen Ke Kelas.....	83
Gambar 4.35 Implementasi Halaman Daftar Kelas <i>Lecturer</i>	83
Gambar 4.36 Implementasi Halaman Daftar Tugas.....	84
Gambar 4.37 Implementasi Halaman Tambah Tugas.....	84
Gambar 4.38 Implementasi Halaman Daftar Kuis	85
Gambar 4.39 Implementasi Halaman Tambah Kuis	85
Gambar 4.40 Implementasi Halaman Tambah Soal Kuis.....	86
Gambar 4.41 Implementasi Halaman Detail Kuis	86
Gambar 4.42 Implementasi Halaman Detail Jawaban Kuis.....	87
Gambar 4.43 Implementasi Halaman <i>Feedback</i> Kuis.....	88
Gambar 4.44 Implementasi Halaman Daftar Materi.....	88
Gambar 4.45 Implementasi Halaman Tambah Materi	89

DAFTAR TABEL

Tabel 2.1 Perbedaan Penelitian Terkait	18
Tabel 3.1 Jadwal Kegiatan Penelitian	21
Tabel 4.1 Tabel <i>User</i>	48
Tabel 4.2 Tabel <i>Subject</i>	49
Tabel 4.3 Tabel <i>Class</i>	49
Tabel 4.4 Tabel <i>Reference</i>	50
Tabel 4.5 Tabel <i>Task</i>	51
Tabel 4.6 Tabel <i>Task Submission</i>	52
Tabel 4.7 Tabel <i>Announcement</i>	53
Tabel 4.8 Tabel <i>Quiz</i>	53
Tabel 4.9 Tabel <i>Problem</i>	55
Tabel 4.10 Tabel <i>Answer</i>	56
Tabel 4.11 Tabel <i>Feedback</i>	56
Tabel 4.12 Halaman Daftar <i>Task</i>	93
Tabel 4.13 Halaman Tambah <i>Task</i>	94
Tabel 4.14 Halaman Daftar <i>Quiz</i>	94
Tabel 4.15 Halaman Tambah <i>Quiz</i>	94
Tabel 4.16 Halaman Tambah Soal <i>Quiz</i>	94
Tabel 4.17 Halaman Detail <i>Quiz</i>	95
Tabel 4.18 Halaman Detail Jawaban <i>Quiz</i>	95
Tabel 4.19 <i>Modal feedback quiz</i>	95
Tabel 4.20 Halaman Daftar <i>Module</i>	95
Tabel 4.21 Halaman Tambah <i>Module</i>	96
Tabel 4.22 <i>User Login /api/users/login Method POST</i>	97
Tabel 4.23 <i>Authorized User Resource Access</i>	97
Tabel 4.24 <i>Get User Credential /api/users Method GET</i>	97
Tabel 4.25 <i>Edit User Profile /api/users Method PUT</i>	97
Tabel 4.26 <i>Get User's Profile Pic /api/users/pic Method GET</i>	98
Tabel 4.27 <i>Upload Profile's Pic /api/users/pic Method POST</i>	98

Tabel 4.28 Daftar User /api/users/master Method GET	98
Tabel 4.29 Membuat User /api/users Method POST	99
Tabel 4.30 Membuat Subject /api/subjects Method POST.....	99
Tabel 4.31 Edit Subject /api/subjects/{subjectID} Method PUT	100
Tabel 4.32 Menghapus Subject /api/subjects/{subjectID} Method DELETE	100
Tabel 4.33 Membuat Class /api/classes Method POST	100
Tabel 4.34 Edit Class /api/classes/{classID} Method PUT	101
Tabel 4.35 Menghapus Class /api/classes/{classID} Method DELETE	102
Tabel 4.36 Menugaskan Lecturer Ke Kelas /api/classes/{classID} Method PUT....	102
Tabel 4.37 Student Register To Class /api/users/classes Method PUT	103
Tabel 4.38 Get User's Joined Classes /api/users/classes Method GET	103
Tabel 4.39 Lecturer View Student's Waiting List /api/users/classes/{classID}/waiting-list Method GET	103
Tabel 4.40 Get User's Today Schedules	104
Tabel 4.41 Viewing Students Of Class /api/users/classes/{classID}/students Method GET	104
Tabel 4.42 View Today's Schedules /api/users/schedules Method GET.....	104
Tabel 4.43 Viewing List of Students of a Class /api/users/classes/{classId}/students Method GET	104
Tabel 4.44 Get All Subjects /api/subjects Method GET.....	105
Tabel 4.45 Get All Classes /api/classes Method GET	105
Tabel 4.46 Get Class Activity History Such As Announcements, Tasks, Quizzes, and Modules /api/classes/{classId}/histories Method GET	105
Tabel 4.47 Accept Or Reject Student Request to Class /api/waiting- lists/{waitingListId} Method PUT	106
Tabel 4.48 Lecturer Uploads Modules /api/references Method POST	106
Tabel 4.49 Get Modules Uploaded In Class /api/references/classes/{classId} Method GET	107
Tabel 4.50 Get Module Detail /api/references/{referenceId} Method GET	108
Tabel 4.51 Get Modules Attachment By Id /api/references/attachments/{attachmentId} Method GET	108

Tabel 4.52 Delete Module /api/references/{referenceId} Method DELETE	109
Tabel 4.53 Lecturer Uploads Announcements /api/announcements Method POST	109
Tabel 4.54 Get Announcements Uploaded In Class /api/announcements/classes/{classId} Method GET.....	110
Tabel 4.55 Get Announcements Detail /api/announcements/{announcementId} Method GET	110
Tabel 4.56 Get Announcment Attachment By Id /api/announcements/attachments/{attachmentId} Method GET	111
Tabel 4.57 Delete Announcements /api/announcements/{announcementId} Method DELETE	111
Tabel 4.58 Lecturer Uploads Task /api/tasks Method POST.....	112
Tabel 4.59 Get Tasks Uploaded In Class /api/tasks/classes/{classId} Method GET	113
Tabel 4.60 Get Tasks Detail /api/tasks/{taskId} Method GET.....	113
Tabel 4.61 Get Task Attachment By Id /api/tasks/attachments/{attachmentId} Method GET	114
Tabel 4.62 Upload Task Submission /api/tasks/submissions Method POST	114
Tabel 4.63 Get Task Submission /api/tasks/submissions/{submissionId} Method GET	115
Tabel 4.64 Turn In Task Submission /api/tasks/submissions/{submissionId}/turn- in Method PUT.....	115
Tabel 4.65 Get Task Submission Detail /api/tasks/submissions/{submissionId} Method GET	116
Tabel 4.66 Delete Task Submission /api/tasks/submissions/{submissionId} Method DELETE	116
Tabel 4.67 Get Task Submission Attachment /api/tasks/submission/attachments/{attachmentId} Method GET	117
Tabel 4.68 Add Quiz To Class /api/quizzes Method POST.....	117
Tabel 4.69 Get Quiz In Class /api/quizzes/classes/{classId} Method GET	118
Tabel 4.70 Get Quiz Detail /api/quizzes/{quizId} Method GET	118
Tabel 4.71 Get Quiz Problems /api/quizzes/{quizId}/problems Method GET	119
Tabel 4.72 Add Quiz Problems /api/quizzes/{quizId}/problems Method POST	119

Tabel 4.73 <i>Submit Quiz Multiple Choice Answer</i>	
<i>/api/quizzes/{quizId}/problems/choice Method PUT</i>	120
Tabel 4.74 <i>lecturer View Students Answers Of Quiz</i>	
<i>/api/quizzes/{quizId}/students/{userId}/answers</i>	121
Tabel 4.75 <i>Get All Students Quiz Submissions</i> <i>/api/quizzes/{quizId}/students</i>	
<i>Method GET</i>	121
Tabel 4.76 <i>Bobot Jawaban UAT</i>	122
Tabel 4.77 <i>Indikator Tingkat Keberhasilan UAT</i>	123
Tabel 4.78 <i>Pertanyaan Kuisisioner Role Admin</i>	123
Tabel 4.79 <i>Pertanyaan Kuisisioner Role Lecturer</i>	124

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pendidikan merupakan aspek penting dalam perkembangan suatu negara. Dalam era digital, teknologi semakin mempengaruhi berbagai sektor, termasuk sektor pendidikan. *Learning management system (LMS)* merupakan teknologi yang sangat berperan dalam dunia pendidikan. Penggunaan *Learning management system (LMS)* tidak hanya digunakan pada jenjang universitas saja sudah banyak sekolah menengah atas yang menerapkan *learning management system (LMS)* sebagai alat bantu para guru dan murid.

Learning management system (LMS) adalah sebuah perangkat lunak untuk membuat materi pembelajaran berbasis *web* yang mengelola kegiatan pembelajaran beserta hasilnya dan memfasilitasi interaksi antara dosen dan pembelajar, antara dosen dan dosen, dan antara pembelajar dan pembelajar (Permana, 2013). *Learning management system (LMS)* merupakan sebuah sistem terintegrasi yang bersifat komprehensif yang tidak hanya sekadar memasukkan bahan ajar, namun juga dapat mengakomodasi sistem pembelajaran yang mengatur peran dosen, mahasiswa, pemanfaatan sumber belajar, pengelolaan pembelajaran, sistem evaluasi dan monitoring pembelajaran (Munir, 2010).

Berdasarkan penelitian yang dilakukan, menurut Sumarna sebuah *learning management system* tidak hanya digunakan untuk menyimpan sebuah konten pembelajaran dan informasi siswa/mahasiswa melainkan juga digunakan sebagai pengingat kewajiban yang harus dilakukan oleh siswa/mahasiswa, mengarahkan siswa/mahasiswa ke materi yang tepat, serta dapat digunakan untuk memantau progres siswa/mahasiswa. Menurut jurnal tersebut sudah banyak institusi pendidikan di Malaysia yang menggunakan *LMS* sebagai platform yang digunakan sebagai *course enhancement*, *hybrid learning* dan *virtual classrooms*. Adopsi *learning management system* yang telah dilakukan menggunakan berbagai metode penelitian seperti ADDIE dan SCRUM seperti yang ditulis oleh (Junianto K et al., 2021). Kebutuhan *learning management system* sangat penting tapi pun

tidak kalah pentingnya sebuah konsep *e-learning* yang menjadi konten dari *learning management system* itu sendiri (Elis Hernawati *et al.*, 2016).

Saat ini transisi dari masa - masa pandemik covid 19 menuju *new normal* berdampak pada salah satu bidang yaitu pendidikan. Sebuah sistem yang dapat mengelola kebutuhan siswa/mahasiswa dan guru/dosen yang *reliable* sangat dibutuhkan salah satunya adalah *learning management system* yang terintegrasi (Sintha *et al.*, 2020). Dalam hal ini terintegrasi tidak hanya sebatas fitur dosen dan mahasiswa tapi juga dapat diintegrasikan dengan mudah oleh *developer* yang berniat membuat sebuah *3rd party application* yang membutuhkan *service* sebuah *learning management system* atau dengan kata lain *developer* yang ingin membuat sebuah aplikasi (seperti aplikasi *mobile* atau *website*) *LMS* tanpa harus membuat semuanya dari awal. Sehingga dapat memudahkan sebuah institusi untuk membangun aplikasi nya sendiri dengan pondasi yang sudah ada.

Sebuah sistem berkembang karena kebutuhan bisnis, arsitektur monolitik menjadi arsitektur yang digunakan untuk membuat perangkat lunak karena merupakan cara termudah dan tercepat. Akan tetapi setelah bertahun – bertahun, untuk menambahkan fitur baru menjadi lebih sulit karena kompleksnya sistem dan struktur dari perangkat lunak yang dibuat, sehingga merubah monolitik menjadi *microservice* memungkinkan perangkat lunak menjadi lebih otonom sehingga sistem dapat bekerja khusus dalam tugas tertentu (Khoirunnisa, 2019). Munculnya *microservice* dapat menggantikan aplikasi monolitik yang kompleks.

Sudah banyak *Learning management system (LMS)* yang eksis di internet seperti *chamilo*. Keberadaan *Learning management system (LMS)* yang sudah ada sekarang pun sudah dapat dengan mudah digunakan sesuai kebutuhan kampus atau instansi pendidikan lainnya. Kekurangan yang sangat terlihat pada *learning management system (LMS)* yang ada sekarang yaitu tidak adanya kemampuan aplikasi untuk dapat diintegrasikan secara langsung dengan *3rd party application* seperti aplikasi *mobile* maupun *desktop*, sehingga fitur – fitur *learning management system (LMS)* yang digunakan mungkin berlebihan dengan kata lain banyak fitur - fitur yang tidak diperlukan. Dengan adanya sebuah *learning management system (LMS)* yang dapat diintegrasikan membuat para *developer*

3rd party application dengan mudah membuat aplikasi yang sesuai kebutuhan, serta pembaruan pada aplikasi juga dengan mudah didapatkan. **Penelitian ini bertujuan untuk membuat sebuah *learning management system* berbasis web dan *services* dengan nama *learnyscape* yang dapat diintegrasikan oleh *developer*.**

Berdasarkan pembahasan *learning management system* yang disebutkan di atas, maka penulis akan menyusun tugas akhir dengan judul **“RANCANG BANGUN *LEARNING MANAGEMENT SYSTEM* BERBASIS WEB DAN *MICROSERVICES*”**. Diharapkan penelitian ini tidak hanya mempermudah institusi pendidikan yang memakainya tetapi juga mempermudah *developer* aplikasi dalam membuat *learning management system* tanpa membuat konsepnya dari awal.

1.2 Rumusan Masalah

Berdasarkan uraian pada latar belakang masalah di atas, dapat dirumuskan masalah:

1. Bagaimana cara merancang dan membangun *learning management system (LMS)* berbasis web ?
2. Bagaimana menganalisis kebutuhan fungsional dari *learning management system* yang akan dibuat ?
3. Bagaimana implementasi arsitektur *microservices* pada *learnyscape* ?

1.3 Batasan Masalah

Batasan masalah pada penelitian ini adalah:

1. *Website* yang akan dibuat hanya akan digunakan oleh *admin* dan dosen bukan mahasiswa.
2. *Service* yang akan dibuat hanya akan menunjang kebutuhan dosen, mahasiswa, *admin* bukan perangkat pejabat lain di suatu universitas (*admin* fakultas, kaprodi, dsb).

3. Kebutuhan fungsional yang akan dibuat hanya mencakup manajemen materi, kuis, tugas dan pengumuman tidak mencakup ruang diskusi atau *chat room*.
4. Penelitian ini tidak akan membahas tentang evaluasi efektivitas sistem atau evaluasi penggunaan.
5. Penelitian ini hanya akan membahas penggunaan *learning management system* di lingkungan pembelajaran formal sekolah atau universitas, sehingga tidak membahas penggunaan di lingkungan informal atau bisnis.
6. Penelitian ini akan membahas teknologi yang digunakan dalam pengembangan *learning management system* berbasis *web* seperti bahasa pemrograman, *framework*, dan *database*.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah, maka tujuan dari penelitian ini adalah:

1. Merancang konsep *learning management system* di lingkup pendidikan formal (sekolah atau universitas).
2. Membangun kebutuhan fungsional pada *LMS* yang dapat memudahkan proses pembelajaran.
3. Membangun *learning management system* berbasis *web* dan *service* yang dapat diintegrasikan.

1.5 Manfaat Penelitian

Berdasarkan tujuan penelitian di atas, maka manfaat dari penelitian ini adalah sebagai berikut:

1. Memudahkan *admin* sekolah atau universitas dalam mengelola sistem pendidikan formal seperti manajemen kelas, mata kuliah, mahasiswa, dosen pengampu dll.
2. Memudahkan *developer* aplikasi yang ingin membuat *LMS* dengan mengintegrasikan *service* yang akan dibuat tanpa perlu memikirkan konsep *LMS* yang dibutuhkan seperti manajemen mahasiswa dan dosen, manajemen kelas, tugas dan kuis.
3. Memudahkan dosen dan mahasiswa untuk melakukan kegiatan ajar mengajar seperti, pengelolaan bahan ajar, penilaian, pembuatan tugas dan sebagainya.

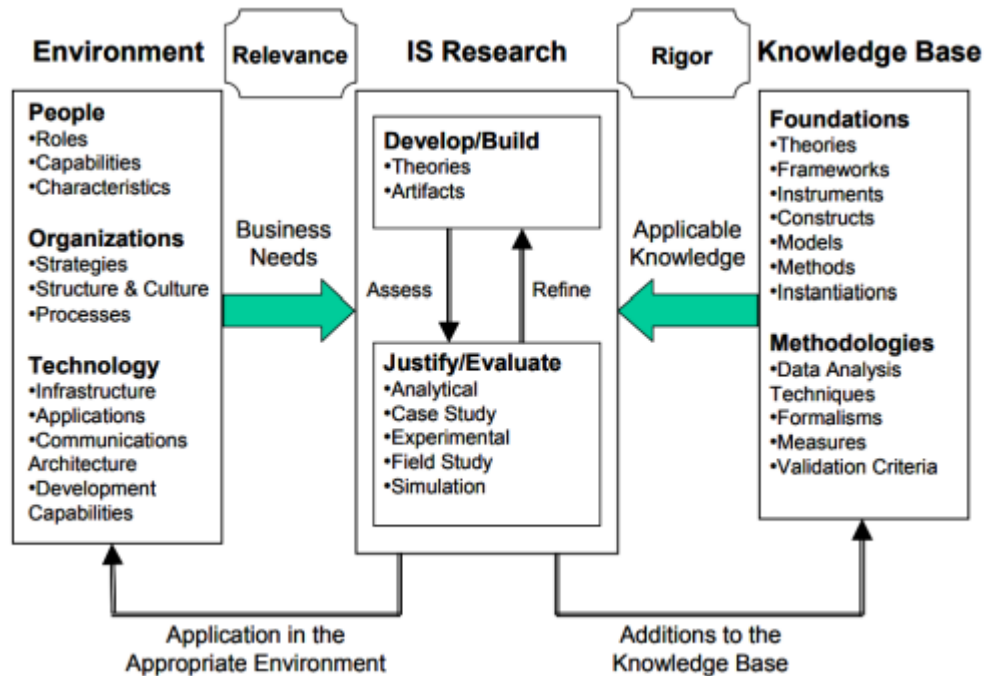
BAB II

TINJAUAN PUSTAKA

2.1 Sistem Informasi

Sistem adalah kumpulan orang yang saling bekerja sama dengan ketentuan aturan yang sistematis dan terstruktur untuk membentuk satu kesatuan yang melaksanakan suatu fungsi untuk mencapai tujuan. Sedangkan informasi adalah data yang diolah menjadi lebih berguna dan berarti bagi penerimanya, serta 5 untuk mengurangi ketidakpastian dalam proses pengambilan keputusan mengenai suatu keadaan. Sistem informasi merupakan suatu kombinasi teratur dari orang, *hardware*, *software*, jaringan komunikasi dan sumber daya data yang mengumpulkan, mengolah dan menyebarkan informasi dalam sebuah organisasi. Sistem informasi dalam suatu organisasi dapat dikatakan sebagai suatu sistem yang menyediakan informasi bagi semua tingkatan dalam organisasi tersebut kapan saja diperlukan. Sistem ini menyimpan, mengambil, mengubah, mengolah dan mengkomunikasikan informasi yang diterima dengan menggunakan sistem informasi atau peralatan sistem lainnya (Aggraeni & Irviani, 2017).

2.2 Framework Penelitian Sistem Informasi



Gambar 2.1 Kerangka Penelitian Sistem Informasi

Sumber: www.researchgate.net/figure/Research-framework-source-adapted-from-Hevner-et-al-2004_fig1_355698973

Pada Gambar 2.1 menyajikan *framework* untuk memahami, melaksanakan, dan mengevaluasi penelitian sistem informasi yang menggabungkan *behavioral science* dan paradigma *design-science*. *Behavioral-science* membahas penelitian melalui pengembangan dan pembenaran teori yang menjelaskan atau memprediksi fenomena yang terkait dengan kebutuhan bisnis yang teridentifikasi. *Design-science* membahas penelitian melalui pembangunan dan evaluasi *artifact* yang dirancang untuk memenuhi kebutuhan bisnis yang teridentifikasi. Tujuan penelitian *behavioral-science* adalah *truth* (kebenaran). Tujuan dari penelitian *design-science* adalah *utility* (kegunaan) (Hevner et al, 2004).

2.3 Learning management system (LMS)

Learning management system (LMS) adalah sebuah infrastruktur yang menyediakan dan mengelola konten, mengidentifikasi dan menilai proses pembelajaran perorangan maupun organisasi secara utuh, mengawasi progres

belajar untuk mencapai tujuan pembelajaran, mengumpulkan dan mempresentasikan data untuk mengawasi proses belajar suatu organisasi secara utuh (Szabo, 2002). *LMS* tidak hanya menyediakan konten tapi juga menangani pembuatan kursus dan administrasinya, menganalisa perbedaan kemampuan individu, mengawasi dan melaporkan proses belajar (Gilhooly, 2001). Beberapa fungsi *LMS* (Noveandini & Wulandari, 2021).

1. Memusatkan dan otomatisasi administrasi
2. Mampu menyediakan layanan dan panduan yang dapat dilakukan oleh *user* sendiri tanpa melibatkan orang lain
3. Merangkai dan mempresentasikan konten pembelajaran secara berkala
4. Mendukung kemudahan portabilitas dan standarisasi yang lebih baik
5. Mengatur materi pembelajaran yang sifatnya *reuse*.

2.4 React

React adalah *library* JavaScript yang bersifat *open-source* yang dibuat oleh *facebook* untuk untuk membangun antarmuka pengguna (UI) pada aplikasi *web* (Rawat & Mahajan, 2020). *Developer* menggunakan React dengan membuat komponen-komponen *reusable* yang mewakili bagian-bagian dari *User Interface* (UI). Komponen ini menerima data melalui *props* dan dapat memiliki *state internal*. React juga memanfaatkan *Virtual DOM* untuk mengoptimalkan pemutakhiran tampilan dengan hanya memperbarui bagian yang berubah.

JSX (JavaScript XML) adalah sintaks ekstensi *javaScript* yang digunakan untuk menggabungkan kode *javaScript* dengan bahasa *markup HTML*. *LifeCycle* komponen dan *hooks* membantu dalam mengelola tindakan yang terjadi pada komponen. Dukungan dari *library* pendukung dan komunitas yang aktif membuat pengembangan dengan react lebih efisien. Fitur-fitur seperti *hot reloading* memungkinkan pengembangan *real-time*. Dengan fitur – fitur ini, ReactJS memudahkan *developer* dalam membangun aplikasi *web* interaktif dan responsif.

2.5 NodeJS

Node.JS adalah sistem perangkat lunak yang didesain untuk pengembangan aplikasi *web*. Node.JS dapat juga disebut sebagai *runtime environment*. Aplikasi ini ditulis dalam campuran Bahasa C++ dan juga JavaScript, mempunyai model *event driven* (basis *event*) dan *asynchronous I/O*. Tidak seperti kebanyakan bahasa JavaScript yang dijalankan pada *web browser*, Node.JS dieksekusi sebagai aplikasi server. Node.JS dapat berjalan di server karena dukungan dari *V8 Engine* buatan Google dan beberapa modul bawaan yang terintegrasi seperti modul *http*, modul *filesystem*, modul *security* dan beberapa modul penting lainnya (Fajrin, 2017). Berdasarkan artikel Muhammad Devano Zaidan pada blog Teknik Informatika Universitas Islam Indonesia NodeJS memiliki beberapa kelebihan sebagai berikut (Zaidan, 2022) :

1. Bersifat *Non-Blocking*

Non-blocking maksudnya adalah sebuah baris kode tidak akan melakukan *blocking* terhadap baris kode setelahnya. Artinya, sebuah kode dapat dieksekusi meskipun proses pada eksekusi sebelumnya belum selesai sehingga memungkinkan kode berjalan secara *asynchronous*. *Asynchronous* berarti program tidak perlu dieksekusi baris per baris. Misal adalah ketika kita melakukan sebuah *HTTP request* kepada server. Kode akan berjalan secara *asynchronous* yang artinya proses eksekusi setelahnya akan dilakukan meskipun proses *request* belum selesai atau belum mendapatkan respon.

2. *Fast Execution*

Kode Javascript yang dieksekusi terbilang mempunyai performa yang bagus dan cepat. Hal ini dimungkinkan karena kode Javascript pada Node.js dijalankan menggunakan *V8 engine* yang ditulis menggunakan bahasa C++. Dengan demikian, Javascript juga akan mendapatkan *behavior* dari C++ yaitu kecepatan dan keamanan yang dapat diandalkan.

3. *Single Threaded*

Node.js memiliki karakteristik *single threaded*. Artinya, Node.js berjalan pada satu proses pada waktu yang sama. Node.js hanya menggunakan satu *thread* saja pada prosesor. Oleh karena itu, program yang dibuat dengan

Node.js bisa dibilang ringan. Akan tetapi, Node.js menyediakan fitur *asynchronous* di mana beberapa *request* dapat berjalan secara bersamaan dan mencegah kode pada program melakukan *blocking*. Node.js dapat menangani ribuan koneksi secara bersamaan dengan server tunggal

2.6 Express js

Express js merupakan sebuah *framework* NodeJS yang berguna untuk mempermudah pembuatan aplikasi berbasis NodeJS. Dengan menggunakan *design pattern* yang dapat disesuaikan dan sangat fleksibel, Express js juga merupakan *framework* yang sangat ringan dan cocok untuk membuat aplikasi *web* dan API (Rachmat, 2017).

Express js dibangun di atas bahasa pemrograman *JavaScript* dan berjalan pada lingkungan *runtime* NodeJS. Pada penelitian ini, Express js digunakan untuk membuat *server site* yang bertanggung jawab untuk mengatur proses pemesanan dan mengelola data di server.

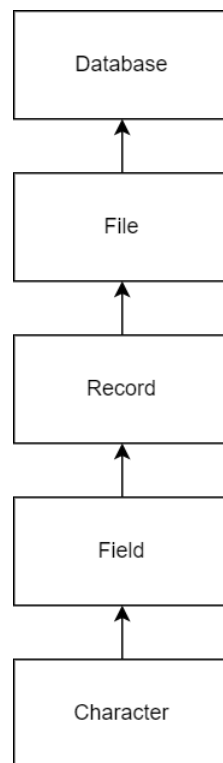
2.7 Database

Database secara umum dapat diartikan sebuah tempat penyimpanan data sebagai pengganti dari sistem konvensional yang berupa dokumen *file*. *Database* didefinisikan kumpulan data yang dihubungkan secara bersama-sama, dan gambaran dari data yang dirancang untuk memenuhi kebutuhan informasi dari suatu organisasi. Berbeda dengan sistem *file* yang menyimpan data secara terpisah, pada *database* data tersimpan secara terintegrasi. Ada beberapa istilah umum yang sering dipakai pada *database*, yaitu sebagai berikut (Rahmayati 2022):

1. *Field*, yaitu sekumpulan kecil dari kata atau sebuah deretan angka-angka.
2. *Record*, yaitu kumpulan dari *field* yang berelasi secara logis.
3. *File*, yaitu kumpulan dari *record* yang berelasi secara logis.
4. *Entity*, yaitu orang, tempat, benda, atau kejadian yang berkaitan dengan informasi yang disimpan.
5. *Attribute*, yaitu setiap karakteristik yang menjelaskan suatu *entity*.

6. *Primary key*, yaitu sebuah *field* yang nilainya unik yang tidak sama antara satu *record* dengan *record* yang lain.
7. *Foreign key*, yaitu sebuah *field* yang nilainya berguna untuk menghubungkan *primary key* yang berada pada tabel yang berbeda

Komponen jenjang data sampai membentuk suatu *database* pada Gambar 2.1 sebagai berikut :



Gambar 2.2 Komponen *Database*

Sumber: wiwitseptiani.wordpress.com/catatan-kuliah/sistem-basis-data-terdistribusi/struktur-atau-jenjang-data/

1. *Character* merupakan bagian yang paling kecil, dapat berupa angka, huruf maupun karakter karakter khusus.
2. *Field* merupakan representasi dari *record* yang berfungsi untuk menunjukkan *item* data seperti nama, jenis kelamin dll.
3. *Record* merupakan kumpulan dari *field*. Menggambarkan sebuah unit data yang berasal dari individu tertentu.

4. *File* merupakan kumpulan dari *record* yang menggambarkan satu kesatuan data yang sejenis.

2.8 RDBMS MySQL

RDBMS (Relational Database Management System) adalah sebuah program *database server* yang mampu menerima dan mengirimkan datanya sangat cepat, multi *user* serta menggunakan perintah dasar *Structured Query Language (SQL)*. MySQL Merupakan sebuah *database server* yang *free*, artinya kita bebas menggunakan *database* ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya. MySQL pertama kali dirintis oleh seorang *programmer database* bernama Michael Widenius. Selain *database server*, MySQL juga merupakan program yang dapat mengakses suatu *database MySQL* yang berposisi sebagai server, yang berarti program kita berposisi sebagai *client*.

MySQL adalah sebuah *database* yang dapat digunakan sebagai *client* maupun server. *Database MySQL* merupakan suatu perangkat lunak *database* yang berbentuk *database* relasional atau disebut *Relational Database Management System (RDBMS)* yang menggunakan suatu bahasa permintaan yang bernama *Structured Query Language (SQL)* (Rahmayati, 2022).

2.9 Black Box Testing

Black Box testing adalah metode pengujian perangkat lunak yang berfokus pada spesifikasi fungsional tanpa adanya pengujian desain dan kode program itu sendiri. *Black box testing* hanya membutuhkan batas bawah dan batas atas pada data yang diharapkan (Cholifah & Sagita, 2018). *Black box testing* berfokus untuk memastikan bahwa setiap proses yang ada dalam sistem berfungsi sebagaimana mestinya dengan tujuan untuk menemukan error yang nantinya akan diperbaiki sehingga sistem dapat dikatakan layak untuk digunakan (Wijaya & Astuti, 2021). *Black box testing* menguji sistem tanpa pengetahuan sebelumnya tentang cara kerja internalnya. Penguji memberikan masukan, dan mengamati keluaran yang dihasilkan oleh sistem yang diuji. Hal ini memungkinkan untuk mengidentifikasi

bagaimana sistem merespons tindakan pengguna yang diharapkan dan tidak diharapkan, waktu respon, masalah kegunaan, dan masalah keandalan.

2.10 REST API

REST API merupakan salah satu dari desain arsitektur yang terdapat di dalam API itu sendiri dan cara kerja dari RESTful API yaitu REST *client* akan melakukan akses pada data/resource pada REST server masing-masing resource. Atau data/resource tersebut akan dibedakan oleh sebuah global ID atau URIs (*Universal Resource Identifiers*).

REST API memungkinkan berbagai sistem untuk berkomunikasi dan mengirim / menerima data dengan cara yang sangat sederhana. Setiap *request* ke API, REST memiliki hubungan antara kata kerja HTTP dan URL. Sumber daya dalam database dalam suatu aplikasi dapat dipetakan dengan *endpoint* (Rizki & Adil, 2018).

Beberapa perintah HTTP antara lain fungsi *GET*, *POST*, *PUT*, *UPDATE* atau *DELETE*. Supaya informasi yang diterima lebih mudah dibaca dan di-*parsing* di sisi *client*, maka respons yang dikirimkan oleh API server dalam bentuk JSON (Choirudin & Adil, 2019).

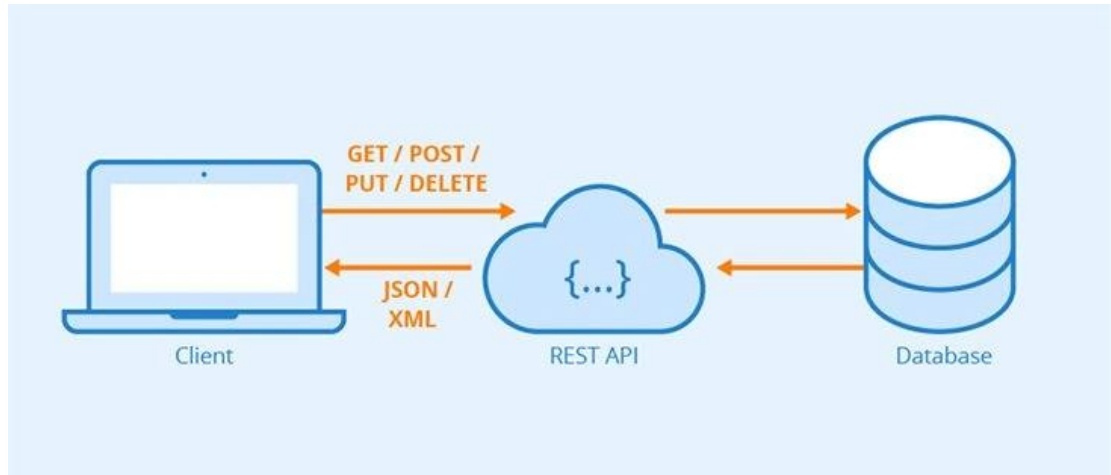
Representasi sumber daya yang dikirim ke klien tergantung pada permintaan dan bagaimana server mengirim data (Indrawan & Adil, 2016).

Ada empat prinsip utama teknologi yang mendasari metode REST (Indrawan & Adil, 2016) yaitu:

1. *Resource identifier* melalui *Uniform Resource Identifier* (URI), yaitu sekumpulan sumber daya yang interaksi antar *client* diidentifikasi oleh REST *web service*.
2. *Uniform interface*, menggunakan operasi *PUT*, *GET*, *POST* dan *DELETE* untuk memanipulasi CRUD (*Create, Read, Update, Delete*).
3. *Uniform interface*, menggunakan operasi *PUT*, *GET*, *POST* dan *DELETE* untuk memanipulasi CRUD (*Create, Read, Update, Delete*) format konten (HTML, XML, PDF, JPEG, *Plain Text* dan lainnya) dapat diakses.

4. *Stateful interactions*, dapat berinteraksi dengan suatu sumber daya bersifat *stateless*, dengan *request messages* tergantung jenis kontennya melalui *hyperlink*.

Pada Gambar 2.2 merupakan ilustrasi interaksi *client* dan server pada arsitektur REST API



Gambar 2.3 Ilustrasi REST API

Sumber: course-net.com/blog/restful-api-adalah

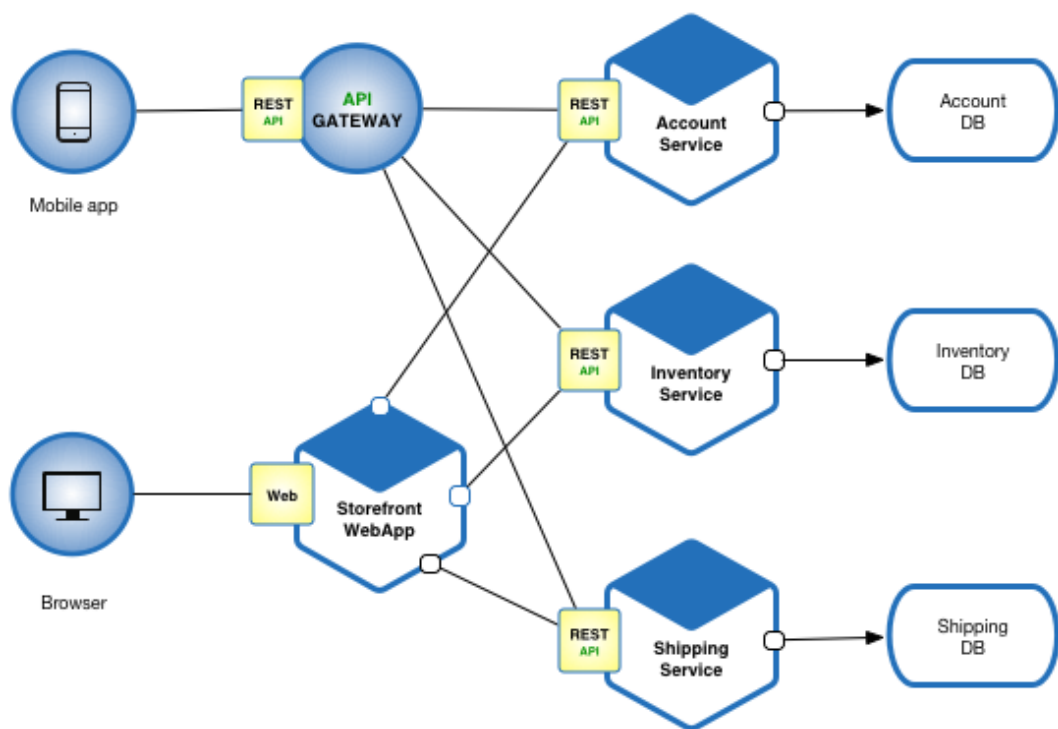
2.11 *Microservice*

Microservices adalah komponen kecil perangkat lunak yang khusus dalam satu tugas tertentu dan bekerja sama untuk mencapai tugas tingkat tinggi (Gonzalez, 2016). Aplikasi diatur sedemikian rupa sehingga saling terpisah menjadi *service-service* kecil yang independen, berfungsi spesifik (*high cohesion*) dan tidak saling bergantung pada komponen program lainnya (*loose coupling*), dengan antarmuka API (*Application Programming Interface*) (Newman, 2015). Secara sederhana arsitektur aplikasi *microservice* menggunakan desain yang memecah aplikasi berdasarkan fungsinya secara spesifik. Tidak sekedar memisahkan berdasarkan *user-role* atau *subdomain* saja, tetapi aplikasi akan di *breakdown* lebih rinci dari segi fungsionalitasnya. Aplikasi dirancang agar setiap fungsi bekerja secara independen dan setiap fungsi dapat menggunakan teknologi sesuai dengan kebutuhan yang berarti akan ada teknologi yang berbeda dalam satu aplikasi besar.

Dalam *microservice* menurut penulis Mike Amundsen dkk mengemukakan sifat aplikasi *microservice* adalah sebagai berikut:

1. Ukurannya kecil
2. Dibatasi oleh konteks
3. Berkembang secara independen
4. Dapat di *deploy* secara independen
5. Desentralisasi
6. Dibangun dan dirilis dengan proses otomatis

Pada gambar 2.3 terlihat konsep dari *microservice* yaitu *service – service* kecil yang independen, *service* dapat berjalan sendiri dan dapat mengelola sendiri *service* yang ada tanpa berkaitan dengan *service* lain.



Gambar 2.4 Ilustrasi Arsitektur *microservice*

Sumber: terralogiq.com/category/teknologi/page/8/

2.12 Domain Driven Design

DDD adalah pendekatan yang digunakan dalam pengembangan aplikasi, *domain model* menjadi pusat masalah yang diselesaikan. Arsitek perangkat lunak dan *developer* menggunakan *domain model* sebagai acuan utama dalam

mendesain perangkat lunak. Lebih jauh, DDD berfokus pada *business logic* di ruang lingkup *customer* dan mengabaikan aspek teknis (Hippchen et al., 2017).

2.13 Penelitian Relevan

Penelitian berjudul “Rancang Bangun *Learning management system* Menggunakan *Framework CodeIgniter* Pada PT. Rekayasa Industri” yang dilakukan oleh Sumarna dkk yang dilakukan pada tahun 2021. Pada penelitian ini dibuat sebuah *Learning management system* menggunakan *framework codeigniter*. Penelitian ini menggunakan model pengembangan *SCRUM* yang merupakan bagian dari *agile development*. Tahapan - tahapan yang dilakukan yaitu, *product backlog* pada tahap ini penulis mengumpulkan semua kebutuhan yang wajib diprioritaskan untuk diselesaikan, *sprint backlog* pada tahap ini dilakukan pengerjaan atau *development* aplikasi, *sprint* pada tahap ini akan dilakukan presentasi kepada pihak perusahaan tentang aplikasi yang akan dibuat sesuai kebutuhan pengguna, *Working Increment of The Software* adalah tahap pengembangan *LMS* berdasarkan kebutuhan perusahaan. Berdasarkan rancangan yang telah dibuat penelitian ini menyimpulkan terdapat beberapa keuntungan dibuatnya *LMS* antara lain, mempermudah peserta maupun *trainer*, mempermudah peserta untuk mengingat kembali materi yang telah disampaikan oleh *trainer*, materi yang disimpan menjadi lebih terstruktur, mempermudah *trainer* dalam mengoreksi *test* (Sumarna et al., 2021).

Penelitian berjudul “Rancang Bangun *Learning management system (LMS)* Berbasis Web Di Sekolah Menengah Kejuruan (SMK) Tunas Harapan Kabupaten Pati” oleh Junianto K, Febrian Murti Dewanto, Rahmat Robi Waliyansyah pada tahun 2021. Penelitian ini bertujuan untuk membuat sebuah *learning management system (LMS)* yang akan digunakan untuk Sekolah Menengah Kejuruan (SMK) Tunas Harapan Kabupaten Pati sebagai media pembelajaran *e-learning* dan akan dilengkapi dengan fitur kuis *online* secara *real-time*. Pada penelitian perancangan *LMS* akan menggunakan model *Analysis, Design, Development, Implementation, and Evaluation (ADDIE)*. Hasil dari *LMS* yang dibuat dilakukan pengujian menggunakan metode *black box* pada tahap ini dilakukan 2 kondisi pengujian

yaitu kondisi normal aplikasi dan kondisi tidak normal. Pada pengujian dalam kondisi normal diberikan 20 *test case* dan diperoleh 20/20 *test case* valid dan 0/20 *test case* gagal mulai dari membuka aplikasi, *login*, manajemen siswa, manajemen kelas, dst. Pada pengujian dalam kondisi tidak normal diberikan 6 *test case* dan diperoleh 6/6 *test case* valid dan 0/6 *test case* gagal (Junianto K et al., 2021).

Penelitian berjudul “Perancangan dan Penerapan Konten *E-Learning* melalui *Learning Management System* dalam Meningkatkan Motivasi Belajar” yang dilakukan oleh Elis Hernawati, Pramuko Aji pada tahun 2016. Penelitian ini ditujukan untuk merancang dan membangun sebuah *E-Learning* terkhusus untuk mata kuliah basis data pada program studi D3 Manajemen Informatika Fakultas Ilmu Terapan Universitas Telkom. Pada mata kuliah basis data masih dilakukan secara konvensional yang membuat mahasiswa cepat untuk jenuh dan bosan sehingga dibangun *E-Learning* yang dapat meningkatkan motivasi mahasiswa. *E-Learning* yang akan dibuat akan diintegrasikan dengan *LMS* Universitas Telkom yang bernama *iDea*. Penelitian ini menghasilkan konten *e-learning* untuk mata kuliah pemrograman basis data yang diintegrasikan pada aplikasi *e-learning* *iDea* Universitas Telkom. Konten *e-learning* mata kuliah pemrograman yang telah dibangun akan diterapkan dalam sistem pembelajaran mata kuliah pemrograman basis data pada mahasiswa program studi D3 Manajemen Informatika. Berdasarkan hasil pengujian dan pengukuran konten *e-learning* mata kuliah pemrograman basis data yang dibangun mampu memberikan solusi terhadap permasalahan yang terjadi yaitu dengan adanya konten *e-learning* mata kuliah pemrograman basis data yang diintegrasikan pada aplikasi *e-learning* *iDea* dapat meningkatkan minat belajar mahasiswa sebesar 87% dan 83% memahami materi pemrograman basis data. Sedangkan hasil pengujian terhadap ketertarikan konten materi *e-learning* rata-rata menunjukkan respon positif (Hernawati&Aji, 2016). Penjelasan perbedaan penelitian terkait di atas dengan penelitian penulis dapat dilihat pada Tabel 2.1 di bawah ini.

Tabel 2.1 Perbedaan Penelitian Terkait

No.	Penelitian Terkait	Hasil Penelitian	Perbedaan Penelitian
1.	Rancang Bangun <i>Learning management system</i> Menggunakan <i>Framework CodeIgniter</i> Pada PT. Rekayasa Industri (Sumarna et al., 2021)	Pada penelitian yang dilakukan oleh Sumarna, Mohammad Roffi Suhendry, Eri Riana, Verry Riyanto, Hafis Nurdin pada tahun 2021 menghasilkan sebuah <i>learning management system</i> yang dibangun menggunakan <i>framework codeigniter</i> . Pada tahap pengembangannya penulis menggunakan metode <i>SCRUM</i> yang merupakan bagian dari metode pengembangan <i>agile</i> .	Perbedaan penelitian Sumarna dkk dengan penelitian penulis adalah teknologi yang digunakan untuk membangun <i>learning management system</i> dan metode tahapan pengembangan/penelitian. Penulis menggunakan <i>nodejs</i> dan <i>express js</i> sebagai <i>framework</i> untuk membangun <i>learning management system</i> dan metode <i>waterfall</i> sebagai metode penelitian/pengembangan, sedangkan Sumarna dkk menggunakan <i>framework codeigniter</i> dan metode <i>SCRUM</i> pada penelitian terkait.
2.	Rancang Bangun <i>Learning</i>	Penelitian yang dilakukan oleh Junianto K,	Perbedaan penelitian terkait

No.	Penelitian Terkait	Hasil Penelitian	Perbedaan Penelitian
	<i>management system (LMS) Berbasis Web Di Sekolah Menengah Kejuruan (SMK) Tunas Harapan Kabupaten Pati</i> (Junianto K et al., 2021)	Febrian Murti Dewanto, Rahmat Robi Waliyansyah pada tahun 2021, menghasilkan sebuah <i>learning management system</i> berbasis <i>web</i>. Penelitian ini menggunakan model <i>Analysis, Design, Development, Implementation, and Evaluation (ADDIE)</i>. Pada <i>learning management system</i> yang dihasilkan terdapat fitur kuis online secara <i>realtime</i>.	dengan penelitian ini adalah pada penelitian yang dilakukan Junianto Khoirul dkk <i>learning management system</i> yang dihasilkan hanya berbasis <i>web</i> sedangkan keluaran pada penelitian ini adalah <i>learning management system</i> berbasis <i>web</i> dan <i>microservices</i>.
3.	Perancangan dan Penerapan Konten <i>e-Learning</i> melalui <i>Learning management system</i> dalam Meningkatkan Motivasi Belajar (Hernawati&Aji,.2016)	Penelitian yang dilakukan oleh Elis Hernawati, Pramuko Aji pada tahun 2016 merancang sebuah <i>e-learning</i> yang akan digunakan untuk mata kuliah sistem basis data. <i>E-learning</i> yang dibangun kemudian akan diintegrasikan oleh <i>learning management system</i> Universitas Telkom yang bernama <i>iDea</i>.	Perbedaan penelitian yang dilakukan oleh Elis Hernawati dkk dengan penelitian ini adalah konsep sistem yang akan dibangun. Pada penelitian yang dilakukan oleh Elis Hernawati dkk konsep sistem yang dibangun adalah sebuah <i>e-learning</i> lalu kemudian

No.	Penelitian Terkait	Hasil Penelitian	Perbedaan Penelitian
			dapat diintegrasikan oleh <i>learning management system</i> Universitas Telkom yang bernama <i>iDea</i> sedangkan pada penelitian ini penulis membangun sebuah <i>learning management system</i> yang mengakomodir semua mata kuliah / subjek terkait pada suatu instansi pendidikan.