

**PENGEMBANGAN *WEBSITE* DENGAN IMPLEMENTASI SISTEM  
*WEB SERVICE* MENGGUNAKAN METODE *REST API* SEBAGAI  
MEDIA PENGHUBUNG *MARKETING & CUSTOMER*  
PT. AURA UNGGUL**

**SKRIPSI**



**JONATHAN LIE**

**202010000030**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS SAINS & TEKNOLOGI  
UNIVERSITAS BUDDHI DHARMA  
TANGERANG  
2024**

**PENGEMBANGAN *WEBSITE* DENGAN IMPLEMENTASI SISTEM  
*WEB SERVICE* MENGGUNAKAN METODE *REST API* SEBAGAI  
MEDIA PENGHUBUNG *MARKETING & CUSTOMER*  
PT. AURA UNGGUL**

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk kelengkapan gelar kesarjanaan pada  
Program Studi Teknik Informatika  
Jenjang Pendidikan Strata 1**



**JONATHAN LIE**

**202010000030**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS SAINS & TEKNOLOGI  
UNIVERSITAS BUDDHI DHARMA  
TANGERANG**

**2024**

## LEMBAR PERSEMBAHAN

每一个成功者都有一个开始。勇于开始，才能找到成功

*Měi yī gè chéng gōng zhě dōu yǒu yī gè kāi shǐ. Yǒng yú kāi shǐ, cái néng zhǎo dào chéng gōng*

*“Kesuksesan pasti bermula dari suatu permulaan.*

*Berani memulai, baru akan menemukan kesuksesan.”*

Dengan mengucapkan puji syukur kepada Tuhan Yang Maha Esa, Skripsi ini saya persembahkan untuk:

1. Para Buddha dan Bodhisatva yang selalu menuntun dan menjadi pedoman dalam hidup saya.
2. Kedua Orang Tua yang telah membesarkan dan selalu membimbing, mendukung, serta memotivasi dan memberi apa yang terbaik agar dapat meraih kesuksesan.
3. Adik saya yang telah memberikan dukungan semangat serta dorongan yang senantiasa diberikan.
4. Rekan-rekan seperjuangan bimbingan skripsi yang selalu berjuang Bersama
5. Dewan direksi serta rekan-rekan dari PT. Aura Unggul yang memberikan dorongan dan dukungan
6. Salah satu rekan saya yang sudah memberikan bantuan, *support* dan dorongan sehingga saya dapat menyelesaikan Skripsi saya dengan baik
7. Rekan-rekan Ormawa (Organisasi Mahasiswa), Baik dari SEMA FST, SEMA FB, SEMA FSH, maupun BEM UBD yang selalu mendukung saya dalam menyelesaikan skripsi ini agar dapat selesai tepat pada waktunya

**UNIVERSITAS BUDDHI DHARMA**  
**LEMBAR PERNYATAAN KEASLIAN SKRIPSI**

Yang bertanda tangan di bawah ini:

NIM : 20201000030  
Nama : Jonathan Lie  
Jenjang Studi : Strata 1  
Program Studi : Teknik Informatika  
Perminatan : Networking Specialist

Dengan ini saya menyatakan bahwa:

1. Skripsi ini adalah asli dan belum pernah diajukan untuk mendapat gelar akademik Sarjana atau kelengkapan studi, baik di Universitas Buddhi Dharma maupun di Perguruan Tinggi lainnya.
2. Skripsi ini saya buat sendiri tanpa bantuan dari pihak lain, kecuali arahan dosen pembimbing.
3. Dalam Skripsi ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dan dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan daftar Pustaka.
4. Dalam Skripsi ini tidak terdapat pemalsuan (kebohongan), seperti buku, artikel, jurnal, data sekunder, pengolahan data, dan pemalsuan tanda tangan dosen atau Ketua Program Studi Universitas Buddhi Dharma yang dibuktikan dengan keasliannya.
5. Lembar pernyataan ini saya buat dengan sesungguhnya, tanpa paksaan dan apabila dikemudian hari atau pada waktu lainnya terdapat penyimpangan dan ketidak benaran dalam pernyataan ini, saya bersedia menerima sanksi akademik berupa pencabutan gelar akademik yang telah saya peroleh karena Skripsi ini serta sanksi lainnya sesuai dengan peraturan dan norma yang berlaku.

Tangerang, 29 Juli 2024  
Yang membuat pernyataan,



Jonathan Lie  
20201000030

**UNIVERSITAS BUDDHI DHARMA**  
**LEMBAR PERSETUJUAN PUBLIKASI KARYA ILMIAH**

Yang bertanda tangan di bawah ini:

NIM : 20201000030  
Nama : Jonathan Lie  
Jenjang Studi : Strata 1  
Program Studi : Teknik Informatika  
Perminatan : Networking Specialist

Dengan ini menyetujui untuk memberikan ijin kepada pihak Universitas Buddhi Dharma, Hak Bebas Royalti Non – Eksklusif (*Non-exclusive Royalty-Free Right*) atas karya ilmiah kami yang berjudul: “Pengembangan *Website* Dengan Implementasi Sistem *Web Service* Menggunakan Metode REST API Sebagai Media Penghubung *Marketing & Customer* PT. Aura Unggul”, beserta alat yang diperlukan (apabila ada).

Dengan Hak Bebas Royalti Non – Eksklusif ini pihak Universitas Buddhi Dharma berhak menyimpan, mengalih-media atau format-kan, mengelolanya dalam pangkalan data (*database*), mendistribusikannya, dan menampilkan atau mempublikasikannya di *internet* atau media lain untuk kepentingan akademis tanpa perlu meminta ijin dari saya selama tetap mencantumkan nama saya sebagai penulis atau pencipta karya ilmiah tersebut.

Saya bersedia untuk menanggung secara pribadi, tanpa melibatkan pihak Universitas Buddhi Dharma, segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya ilmiah saya ini.

Demikian Pernyataan ini saya buat dengan sebenarnya.

Tangerang, 29 Juli 2024  
Yang membuat pernyataan,



Jonathan Lie  
20201000030

**UNIVERSITAS BUDDHI DHARMA**

**LEMBAR PENGESAHAN PEMBIMBING**

**PENGEMBANGAN *WEBSITE* DENGAN IMPLEMENTASI SISTEM  
*WEB SERVICE* MENGGUNAKAN METODE REST API SEBAGAI  
MEDIA PENGHUBUNG *MARKETING & CUSTOMER***

**PT. AURA UNGGUL**

Dibuat Oleh:

NIM : 20201000030

Nama : Jonathan Lie

Telah disetujui untuk dipertahankan di hadapan Tim Penguji Ujian

Komprehensif

Program Studi Teknik Informatika

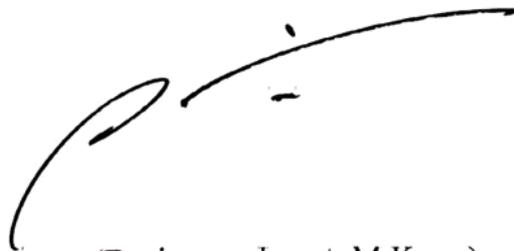
Perminatan Networking Specialist

Tahun Akademik 2023/2024

Disahkan oleh,

Tangerang, 29 Juli 2024

**Pembimbing,**



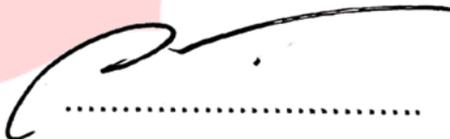
(Desiyanna Lasut, M.Kom.)

NIDN : 0402128601

## LEMBAR PENGESAHAN TIM PENGUJI

Nama : Jonathan Lic  
NIM : 20201000030  
Fakultas : Sains dan Teknologi  
Judul Skripsi : Pengembangan *Website* Dengan Implementasi Sistem *Web Service*  
Menggunakan Metode REST API Sebagai Media Penghubung  
*Marketing & Customer* PT. Aura Unggul

Dinyatakan LULUS setelah mempertahankan di depan Tim Penguji pada hari Senin, 29 Juli 2024

	Nama Penguji:	Tanda Tangan:
Ketua Sidang	: Rino, M.Kom NIDN: 0420058502	 .....
Penguji I	: Susanto Hariyanto, M.Kom NIDN: 0428128601	 .....
Penguji II	: Desiyanna Lasut, M.Kom NIDN: 0402128601	 .....

Mengetahui,  
Dekan Fakultas Sains dan Teknologi



**Dr. Yakub, M.Kom., M.M.**  
NIDN :0304056901

UNIVERSITAS BUDDHI DHARMA

LEMBAR PENGESAHAN SKRIPSI

PENGEMBANGAN *WEBSITE* DENGAN IMPLEMENTASI SISTEM  
*WEB SERVICE* MENGGUNAKAN METODE REST API SEBAGAI  
MEDIA PENGHUBUNG *MARKETING & CUSTOMER*

PT. AURA UNGGUL

Dibuat Oleh:

NIM : 20201000030

Nama : Jonathan Lie

Telah disetujui untuk dipertahankan di hadapan Tim Penguji Ujian

Komprehensif

Program Studi Teknik Informatika

Perminatan Networking Specialist

Tahun Akademik 2023/2024

Disahkan oleh,

Tangerang, 29 Juli 2024

Dekan,

Ketua Program Studi,



Dr. Yakub, M.Kom., M.M.  
NIDN : 0304056901



Hartana Wijaya, M.Kom.  
NIDN : 0412058102

## KATA PENGANTAR

Dengan mengucapkan Puji Syukur kepada Tuhan Yang Maha Esa, yang telah memberikan Rahmat dan Karunia-Nya kepada penulis sehingga dapat Menyusun dan menyelesaikan Skripsi ini dengan judul **Pengembangan Website Dengan Implementasi Sistem Web Service Menggunakan Metode REST API Sebagai Media Penghubung Marketing & Customer PT. Aura Unggul**. Tujuan utama dari pembuatan Skripsi ini adalah sebagai salah satu syarat kelengkapan dalam menyelesaikan program pendidikan Strata 1 Program Studi Teknik Informatika di Universitas Buddhi Dharma. Dalam Penyusunan Skripsi ini penulis banyak menerima bantuan dan dorongan baik moril maupun materil dari berbagai pihak, maka pada kesempatan ini penulis banyak menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Ibu Dr. Limajatini, S.E., M.M., B.K.P, sebagai Rektor Universitas Buddhi Dharma.
2. Bapak Dr. Yakub, M.Kom., M.M, sebagai Dekan Fakultas Sains dan Teknologi.
3. Bapak Hartana Wijaya, S.Kom., M.Kom., sebagai Ketua Program Studi Teknik Informatika.
4. Ibu Desiyanna Lasut, S.Kom., M.Kom. sebagai pembimbing yang telah membantu dan memberikan dukungan serta harapan untuk menyelesaikan penulisan Skripsi ini.
5. Orang tua dan Keluarga yang selalu memberikan dukungan baik moril dan materil.
6. Sahabat dan teman-teman seperjuangan yang selalu membantu dan memberikan semangat.

Serta semua pihak yang terlalu banyak untuk disebutkan satu-persatu sehingga terwujudnya penulisan ini. Penulis menyadari bahwa penulisan Skripsi ini masih belum sempurna, untuk itu penulis mohon kritik dan saran yang bersifat membangun demi kesempurnaan penulisan di masa yang akan datang.

Akhir kata semoga Skripsi ini dapat berguna bagi penulis khususnya dan bagi para pembaca yang berminat pada umumnya.

Tangerang, 29 Juli 2024

Penulis

PENGEMBANGAN *WEBSITE* DENGAN IMPLEMENTASI SISTEM *WEB SERVICE* MENGGUNAKAN METODE *REST API* SEBAGAI MEDIA PENGHUBUNG *MARKETING & CUSTOMER* PT. AURA UNGGUL

**ABSTRAK**

Teknologi informasi sudah sangatlah sering digunakan untuk membantu dalam banyak hal seperti contohnya pekerjaan yang biasanya akan memakan waktu yang begitu lama. Penggunaan komputer dalam dunia bisnis, pemerintahan, ataupun pribadi menunjukkan peningkatan kebutuhan akan kemajuan ilmu pengetahuan dan teknologi untuk mendukung segala macam aktivitas manusia. Organisasi apapun baik institusi maupun korporasi sangat pasti menginginkan segala sesuatu dilakukan secara mudah, cepat, efektif serta efisien. Maksud dari adanya media penghubung ini yaitu untuk membantu institusi dalam melakukan manajemen pengelolaan pelanggan yang sudah ada dalam suatu sistem untuk dapat menghemat waktu. Seperti contoh, adanya pelanggan yang masih bisa melakukan pemesanan tetapi memiliki tagihan yang sudah melewati batas masa jatuh temponya dan harus melakukan pembayaran, dapat menjadi masalah yang krusial terhadap perputaran ekonomi pada perusahaan. Oleh karena itu, dengan adanya sistem *portal* ini, dapat membantu dalam membatasi piutang pada pelanggan, sehingga apabila terdapat tagihan piutang yang sudah melewati batas jatuh tempo, maka sistem akan secara otomatis memberikan pengingat untuk melakukan pembayaran serta melarang pelanggan untuk melakukan pemesanan sebelum pelanggan tersebut melakukan pembayaran terlebih dahulu. Setelah dilakukan perancangan dan pengujian terhadap sistem yang dibuat, maka dapat ditarik Kesimpulan bahwa sistem manajemen pelanggan menggunakan metode *REST API* yang dibangun pada PT. Aura Unggul, dapat menjadi solusi untuk membantu kedua belah pihak dalam melakukan manajemen hubungan perusahaan dengan pelanggan. Dengan adanya sistem ini dapat mempercepat proses kegiatan transaksi yang berlangsung secara akurat, tegas, dan tertata. Hubungan antara perusahaan dengan pembeli yang dahulu masih dilakukan secara konvensional, kini dapat dilakukan secara terkomputerisasi

**Kata kunci** : Sistem Manajemen Pelanggan, Metode *REST API*, *Web Service*, Teknik Informatika akademik, *Networking Specialist*.

*WEBSITE DEVELOPMENT WITH THE IMPLEMENTATION OF A WEB SERVICE SYSTEM USING THE REST API METHOD AS A MEDIUM FOR CONNECTING MARKETING & CUSTOMERS PT. AURA UNGGUL*

***ABSTRACT***

*Information technology is very often used to help in many ways such as work that will usually take a long time. The use of computers in the world of business, government, or personal shows the increasing need for advances in science and technology to support all kinds of human activities. Every organization, both institutions and companies, must want everything to be done easily, quickly, effectively and efficiently. The purpose of this liaison media is to assist institutions in managing existing customer management in a system to save time. For example, the existence of customers who can still place orders but have bills that have passed the due date and must make payments, can be a crucial problem for the company's economic turnover. Therefore, with this portal system, it can help in limiting receivables to customers, so that if there is a receivable bill that has passed the due date, the system will automatically provide a reminder to make a payment and prohibit customers from placing orders before the customer makes a payment first. After designing and testing the system created, it can be concluded that the customer management system using the REST API method built at PT Aura Unggul can be a solution to help both parties in managing company relationships with customers. With this system, it can speed up the process of transaction activities that take place accurately, firmly, and regularly. The relationship between the company and the buyer, which was still done conventionally, can now be done computerized.*

***Keywords:*** *Customer Management System, REST API Method, Web Service, Academic Informatics Engineering, Networking Specialist.*

# DAFTAR ISI

LEMBAR JUDUL LUAR SKRIPSI

LEMBAR JUDUL DALAM SKRIPSI

LEMBAR PERSEMBAHAN

LEMBAR PERNYATAAN KEASLIAN SKRIPSI

LEMBAR PERSETUJUAN PUBLIKASI KARYA ILMIAH

LEMBAR PENGESAHAN PEMBIMBING SKRIPSI

LEMBAR PENGESAHAN PENGUJI SKRIPSI

LEMBAR PENGESAHAN SKRIPSI

KATA PENGANTAR.....	i
ABSTRAK.....	ii
<i>ABSTRACT</i> .....	iii
DAFTAR ISI.....	iv
DAFTAR TABEL.....	viii
DAFTAR GAMBAR.....	x
DAFTAR LAMPIRAN.....	xii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Identifikasi Masalah.....	2
1.3 Ruang Lingkup.....	3
1.4 Tujuan & Manfaat Penulisan.....	3
1.4.1 Tujuan Penulisan.....	3
1.4.2 Manfaat Penulisan.....	4
1.5 Teknik Pengumpulan Data.....	5
1.6 Sistematika Dalam Penulisan.....	6
BAB II LANDASAN TEORI.....	8
2.1 Teori Umum.....	8
2.1.1 Data.....	8
2.1.2 Informasi.....	8

2.1.3.	Sistem .....	9
2.1.4.	Aplikasi.....	10
2.1.5.	<i>Internet</i> .....	12
2.2	Teori Khusus.....	13
2.2.1.	<i>Web Service</i> .....	13
2.2.2.	<i>REST</i> .....	14
2.2.3.	<i>API</i> .....	15
2.2.4.	<i>Website</i> .....	16
2.2.5.	<i>CORS (Cross-Origin Resource Sharing)</i> .....	18
2.2.6.	<i>SRM &amp; CRM</i> .....	19
2.3	Teori Analisa & Perancangan .....	21
2.3.1.	Teori Metodologi Penelitian <i>Waterfall</i> .....	21
2.3.2.	Teori Pemodelan Objek .....	23
2.3.3.	Teori Aplikasi .....	30
2.4	Teori Pengujian <i>Blackbox</i> .....	37
2.5	Tinjauan Studi.....	38
2.6	Kerangka Pemikiran.....	70
BAB III METODOLOGI PENELITIAN .....		71
3.1.	Gambaran Umum Perusahaan.....	71
3.1.1.	Sejarah Perusahaan.....	71
3.1.2.	Visi dan Misi PT. Aura Unggul.....	72
3.1.3.	Struktur Organisasi PT. Aura Unggul .....	72
3.1.4.	Tugas dan Wewenang.....	72
3.2.	Prosedur Sistem Berjalan.....	79
3.2.1.	Prosedur Penambahan <i>Customer</i> Baru .....	79
3.2.2.	Prosedur Pemesanan .....	80
3.2.3.	Prosedur Penagihan .....	80

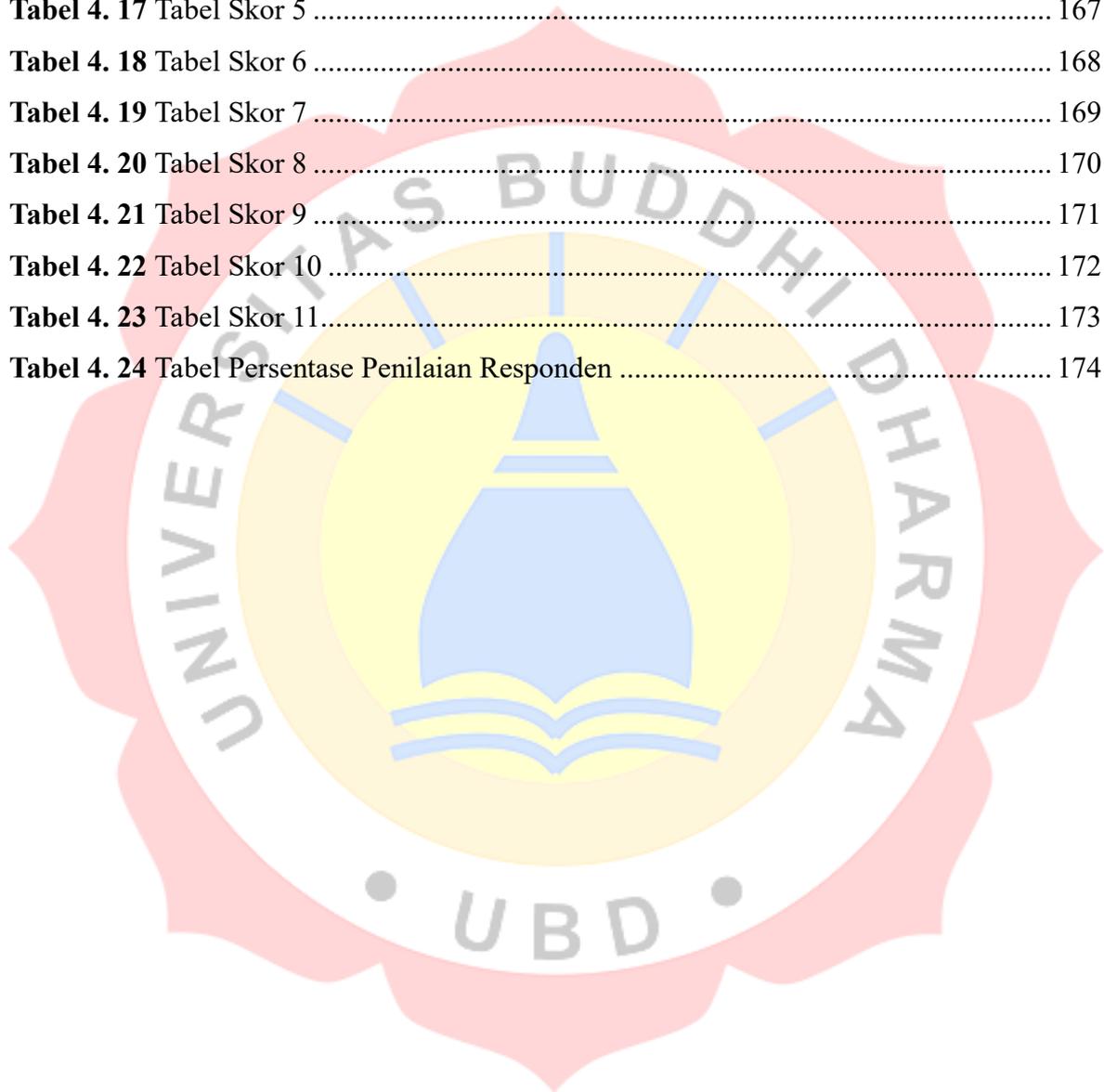
3.3.	<i>Activity Diagram</i> .....	81
3.3.1.	Sistem Penambahan <i>Customer</i> Baru.....	81
3.3.2.	Sistem Pemesanan .....	82
3.3.3.	Sistem Penagihan.....	82
3.4.	Dokumentasi <i>Input</i> dan <i>Output</i> .....	83
3.4.1.	Dokumen Masukan.....	83
3.4.2.	Dokumen Keluaran.....	84
3.5.	Analisa Masalah.....	85
3.6.	Identifikasi Kebutuhan Sistem.....	86
3.7.	Metode REST API.....	87
3.7.1.	<i>Session Login</i> .....	88
3.7.2.	Mengambil <i>userid</i> pengguna.....	88
3.7.3.	Mengambil Hak Akses Pengguna.....	89
3.7.4.	Mengambil Biodata <i>Customer</i> .....	89
3.7.5.	Pengisian Biodata <i>Customer</i> .....	90
3.7.6.	Memperbarui Biodata <i>Customer</i> .....	90
3.7.7.	Menghapus Biodata <i>Customer</i> .....	91
3.7.8.	Mengambil Data Pemesanan .....	91
3.7.9.	Menambahkan Data Pemesanan.....	92
3.7.10.	Memperbarui Data Pemesanan .....	92
3.7.11.	Menghapus Data Pemesanan .....	93
3.7.12.	Mengambil Data Penagihan.....	93
3.7.13.	Menambahkan Data Penagihan.....	94
3.7.14.	Memperbarui Data Penagihan.....	94
3.7.15.	Menghapus Data Penagihan.....	95
3.7.16.	Melakukan Proses <i>Logout</i> .....	95
3.8.	<i>Requirement Elicitation</i> .....	97

3.9. Jadwal Penelitian .....	98
BAB IV PENGUJIAN DAN IMPLEMENTASI .....	99
4.1. Prosedur Sistem Usulan .....	99
4.2. Rancangan Sistem Usulan.....	100
4.2.1. <i>Use Case Diagram</i> .....	100
4.2.2. <i>Use Case Diagram</i> Skenario .....	102
4.2.3. <i>Activity Diagram</i> .....	107
4.2.4. <i>Sequence Diagram</i> .....	121
4.3. Rancangan <i>Database</i> .....	126
4.3.1. <i>Class Diagram</i> .....	126
4.3.2. Struktur File.....	126
4.4. Rancangan Tampilan Program.....	132
4.4.1. Halaman Utama .....	132
4.4.2. Halaman Masukan .....	133
4.4.3. Halaman Keluaran .....	135
4.5. Implementasi Sistem.....	136
4.5.1. Tampilan Program .....	136
4.5.2. Spesifikasi <i>Hardware</i> dan <i>Software</i> .....	148
4.5.3. Pengujian Sistem .....	149
BAB V KESIMPULAN & SARAN.....	177
5.1. Kesimpulan .....	177
5.2. Saran .....	178
DAFTAR PUSTAKA.....	179

## DAFTAR TABEL

<b>Tabel 2. 1</b> Komponen <i>Usecase Diagram</i> .....	24
<b>Tabel 2. 2</b> Simbol <i>Activity Diagram</i> .....	26
<b>Tabel 2. 3</b> Simbol <i>Class Diagram</i> .....	27
<b>Tabel 2. 4</b> Simbol-simbol <i>Sequence Diagram</i> .....	29
<b>Tabel 2. 5</b> Tinjauan Studi 1 .....	38
<b>Tabel 2. 6</b> Tinjauan Studi 2 .....	40
<b>Tabel 2. 7</b> Tinjauan Studi 3 .....	41
<b>Tabel 2. 8</b> Tinjauan Studi 4 .....	44
<b>Tabel 2. 9</b> Tinjauan Studi 5 .....	46
<b>Tabel 2. 10</b> Tinjauan Studi 6 .....	48
<b>Tabel 2. 11</b> Tinjauan Studi 7 .....	51
<b>Tabel 2. 12</b> Tinjauan Studi 8 .....	53
<b>Tabel 2. 13</b> Tinjauan Studi 9 .....	56
<b>Tabel 2. 14</b> Tinjauan Studi 10 .....	58
<b>Tabel 2. 15</b> Tinjauan Studi 11 .....	60
<b>Tabel 2. 16</b> Tinjauan Studi 12 .....	62
<b>Tabel 2. 17</b> Tinjauan Studi 13 .....	64
<b>Tabel 2. 18</b> Tinjauan Studi 14 .....	65
<b>Tabel 2. 19</b> Tinjauan Studi 15 .....	67
<b>Tabel 2. 20</b> Kerangka Pemikiran.....	70
<b>Tabel 3. 1</b> Tabel <i>Requirement Elicitation</i> .....	97
<b>Tabel 3. 2</b> Tabel Jadwal Penelitian ( <i>Gant Chart</i> ).....	98
<b>Tabel 4. 1</b> Tabel Data <i>Users</i> .....	127
<b>Tabel 4. 2</b> Tabel Data <i>Sessions</i> .....	128
<b>Tabel 4. 3</b> Tabel Data <i>Order</i> .....	129
<b>Tabel 4. 4</b> Tabel Data <i>Invoice</i> .....	130
<b>Tabel 4. 5</b> Tabel Data <i>Customer</i> .....	131
<b>Tabel 4. 6</b> Hasil Uji <i>Blackbox</i> pada <i>menu login</i> .....	149
<b>Tabel 4. 7</b> Hasil Uji <i>Blackbox</i> pada <i>sessions user</i> .....	151
<b>Tabel 4. 8</b> Hasil Uji <i>Blackbox</i> pada <i>biodata customer</i> .....	152
<b>Tabel 4. 9</b> Hasil Uji <i>Blackbox</i> Pada <i>Menu Pemesanan</i> .....	155
<b>Tabel 4. 10</b> Hasil Uji <i>Blackbox</i> Pada <i>Menu Invoice Penagihan</i> .....	159

<b>Tabel 4. 11</b> Hasil Uji <i>Blackbox</i> Pada Menu Ganti Password .....	161
<b>Tabel 4. 12</b> Tabel Skor Ideal .....	163
<b>Tabel 4. 13</b> Tabel Skor 1 .....	164
<b>Tabel 4. 14</b> Tabel Skor 2 .....	164
<b>Tabel 4. 15</b> Tabel Skor 3 .....	165
<b>Tabel 4. 16</b> Tabel Skor 4 .....	166
<b>Tabel 4. 17</b> Tabel Skor 5 .....	167
<b>Tabel 4. 18</b> Tabel Skor 6 .....	168
<b>Tabel 4. 19</b> Tabel Skor 7 .....	169
<b>Tabel 4. 20</b> Tabel Skor 8 .....	170
<b>Tabel 4. 21</b> Tabel Skor 9 .....	171
<b>Tabel 4. 22</b> Tabel Skor 10 .....	172
<b>Tabel 4. 23</b> Tabel Skor 11.....	173
<b>Tabel 4. 24</b> Tabel Persentase Penilaian Responden .....	174



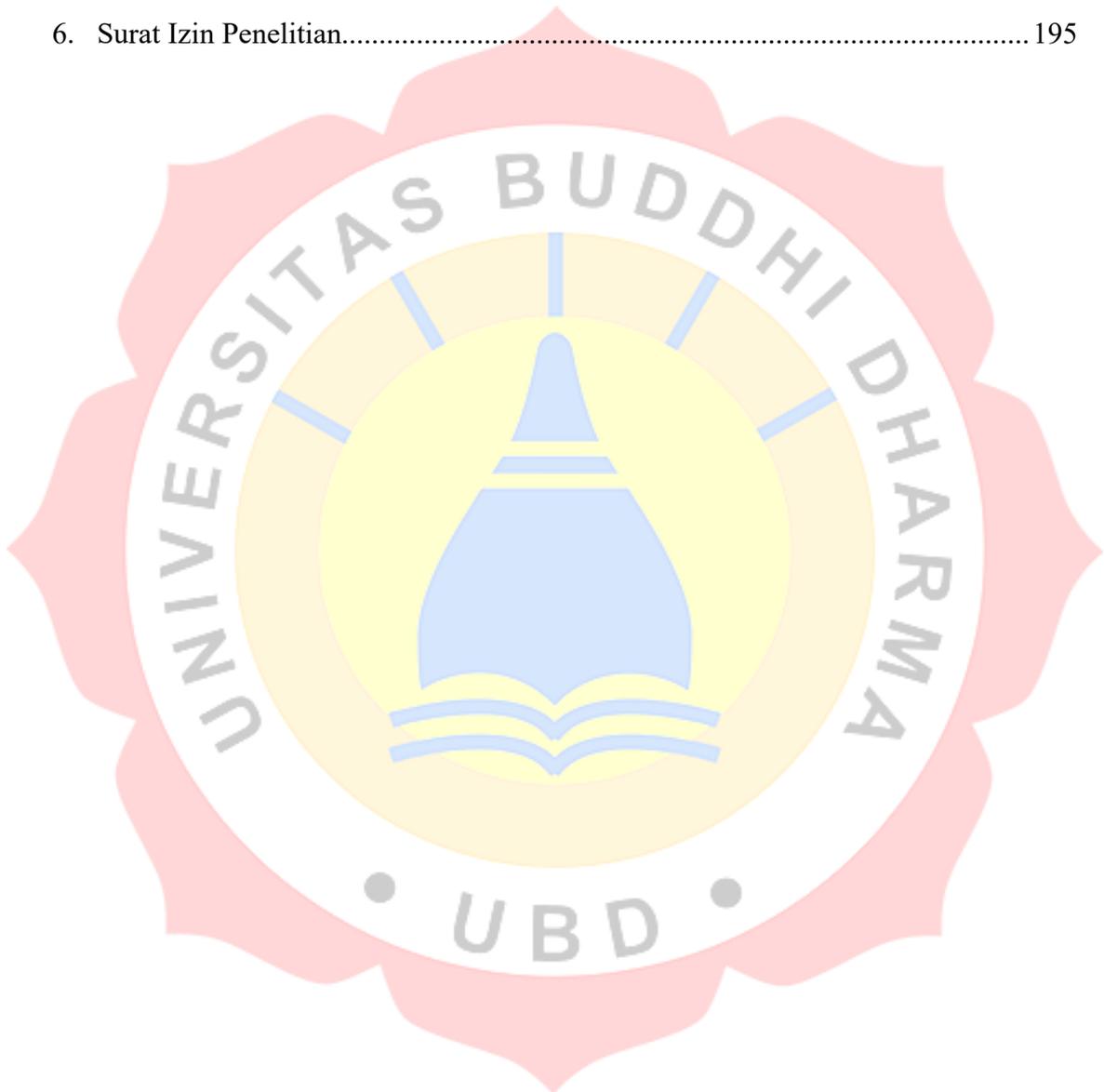
## DAFTAR GAMBAR

<b>Gambar 2. 1</b> <i>Web Service Schema</i> (Sumber: Intern, 2021) .....	13
<b>Gambar 2. 2</b> Contoh cara kerja <i>REST API</i> (Sumber: Team, 2024) .....	14
<b>Gambar 2. 3</b> Ilustrasi cara kerja <i>API</i> dalam jaringan (Sumber: Rawat, 2021).....	15
<b>Gambar 2. 4</b> Ilustrasi <i>CORS</i> (Sumber: Priwanto, 2023).....	18
<b>Gambar 2. 5</b> Metode <i>SDLC</i> (Sumber: Hartono, 2021) .....	21
<b>Gambar 3. 1</b> Struktur Organisasi <i>Aura</i> (Sumber: Direktur PT. <i>Aura Unggul</i> , 2023).....	72
<b>Gambar 3. 2</b> <i>Activity Diagram</i> sistem penambahan <i>customer</i> baru.....	81
<b>Gambar 3. 3</b> <i>Activity Diagram</i> sistem pemesanan barang .....	82
<b>Gambar 3. 4</b> <i>Activity Diagram</i> sistem penagihan.....	82
<b>Gambar 4. 1</b> <i>Use Case Diagram Administrator Login</i> .....	101
<b>Gambar 4. 2</b> <i>Use Case Diagram Marketing Login</i> .....	101
<b>Gambar 4. 3</b> <i>Use Case Diagram Customer Login</i> .....	102
<b>Gambar 4. 4</b> <i>Activity Diagram Login</i> .....	107
<b>Gambar 4. 5</b> <i>Activity Diagram Index Getsessions</i> .....	108
<b>Gambar 4. 6</b> <i>Activity Diagram Administrator biodata customer</i> .....	109
<b>Gambar 4. 7</b> <i>Activity Diagram marketing biodata customer</i> .....	110
<b>Gambar 4. 8</b> <i>Activity Diagram customer biodata customer</i> .....	111
<b>Gambar 4. 9</b> <i>Activity Diagram administrator menu order</i> .....	112
<b>Gambar 4. 10</b> <i>Activity Diagram marketing menu order</i> .....	113
<b>Gambar 4. 11</b> <i>Activity Diagram customer menu order</i> .....	114
<b>Gambar 4. 12</b> <i>Activity Diagram administrator menu invoice</i> .....	115
<b>Gambar 4. 13</b> <i>Activity Diagram marketing menu invoice</i> .....	116
<b>Gambar 4. 14</b> <i>Activity Diagram customer menu invoice</i> .....	117
<b>Gambar 4. 15</b> <i>Activity Diagram Administrator Logout</i> .....	118
<b>Gambar 4. 16</b> <i>Activity Diagram Customer Logout</i> .....	119
<b>Gambar 4. 17</b> <i>Activity Diagram marketing Logout</i> .....	120
<b>Gambar 4. 18</b> <i>Sequence Diagram Login</i> .....	121
<b>Gambar 4. 19</b> <i>Sequence Diagram Biodata Pelanggan</i> .....	121
<b>Gambar 4. 20</b> <i>Sequence Diagram menu pemesanan</i> .....	122
<b>Gambar 4. 21</b> <i>Sequence Diagram menu invoice</i> .....	123
<b>Gambar 4. 22</b> <i>Sequence Diagram menu laporan tagihan</i> .....	124
<b>Gambar 4. 23</b> <i>Sequence Diagram Logout</i> .....	125

<b>Gambar 4. 24</b> <i>Class Diagram Database</i> .....	126
<b>Gambar 4. 25</b> Tampilan <i>Login</i> .....	132
<b>Gambar 4. 26</b> Tampilan <i>Dashboard</i> .....	132
<b>Gambar 4. 27</b> Tampilan Tambah Biodata <i>Customer</i> .....	133
<b>Gambar 4. 28</b> Tampilan Tambah Data <i>Order</i> .....	133
<b>Gambar 4. 29</b> Tampilan Tamhah Data <i>Invoice</i> .....	134
<b>Gambar 4. 30</b> Tampilan Ganti <i>Password</i> .....	134
<b>Gambar 4. 31</b> Tampilan <i>Menu Filter Laporan</i> .....	135
<b>Gambar 4. 32</b> Tampilan <i>Printout Laporan Piutang Pelanggan</i> .....	135
<b>Gambar 4. 33</b> Halaman <i>Login</i> .....	136
<b>Gambar 4. 34</b> Halaman <i>Dashboard useraccess : administrator</i> .....	136
<b>Gambar 4. 35</b> Halaman data pelanggan <i>useraccess : administrator</i> .....	137
<b>Gambar 4. 36</b> Halaman data pelanggan <i>useraccess : customer</i> .....	137
<b>Gambar 4. 37</b> Form tambah data pelanggan <i>useraccess : administrator</i> .....	138
<b>Gambar 4. 38</b> Form tambah data pelanggan <i>useraccess : customer</i> .....	138
<b>Gambar 4. 39</b> Form <i>edit</i> data pelanggan <i>useraccess : administrator</i> .....	139
<b>Gambar 4. 40</b> Konfirmasi Hapus Data Pelanggan <i>useraccess : administrator</i> .....	139
<b>Gambar 4. 41</b> Halaman data pemesanan <i>useraccess : administrator</i> .....	140
<b>Gambar 4. 42</b> Halaman data pemesanan <i>useraccess : customer</i> .....	140
<b>Gambar 4. 43</b> Form tambah data pemesanan <i>useraccess : customer</i> .....	141
<b>Gambar 4. 44</b> Form <i>edit</i> data pemesanan <i>useraccess : administrator</i> .....	141
<b>Gambar 4. 45</b> Form <i>edit</i> data pesanan <i>useraccess : customer</i> .....	142
<b>Gambar 4. 46</b> Konfirmasi Hapus Data Pesanan <i>useraccess : administrator</i> .....	142
<b>Gambar 4. 47</b> Halaman data <i>invoice useraccess : administrator</i> .....	143
<b>Gambar 4. 48</b> Halaman data <i>invoice useraccess : customer</i> .....	143
<b>Gambar 4. 49</b> Form tambah <i>invoice useraccess : administrator</i> .....	144
<b>Gambar 4. 50</b> Form ubah <i>invoice useraccess : administrator</i> .....	144
<b>Gambar 4. 51</b> Konfirmasi hapus data <i>invoice useraccess : administrator</i> .....	145
<b>Gambar 4. 52</b> Halaman laporan tagihan <i>useraccess : administrator</i> .....	145
<b>Gambar 4. 53</b> Cetak Laporan Tagihan .....	146
<b>Gambar 4. 54</b> Pesan kesalahan saat cetak lampiran <i>order</i> oleh <i>marketing</i> .....	146
<b>Gambar 4. 55</b> Gagal menambahkan <i>order</i> oleh <i>customer</i> , biodata belum ACC.....	147
<b>Gambar 4. 56</b> Gagal <i>order</i> oleh <i>customer</i> karena <i>piutang</i> jatuh tempo .....	147

## DAFTAR LAMPIRAN

1. Kartu Bimbingan Skripsi.....	184
2. Hasil Wawancara .....	185
3. <i>Requirement Elicitation</i> .....	188
4. Hasil Kuisisioner.....	192
5. Listing Program .....	193
6. Surat Izin Penelitian.....	195



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

*Web Service* dapat di definisikan dengan sebuah teknologi yang membuat internet dapat saling komunikasi secara berpola *programed to programed*, bertujuan agar client dari berbagai *platform (desktop, website, mobileapps)* dapat mengakses seacara langsung layanan pada *web service* (Nugroho & Primadewi, 2021a). *Web Service* adalah layanan yang dapat diakses melalui internet menggunakan protokol HTTP dan XML. Pengguna bisa mengaksesnya dengan berbagai bahasa pemrograman, perangkat keras, dan sistem operasi berbeda. Layanan *web service* ini menggunakan metode REST API sehingga memungkinkan antarmuka pengguna dengan *server* penyedia layanan dapat saling berkomunikasi. API adalah seperangkat teknik dasar yang digunakan oleh *programmer* untuk membuat perangkat lunak di sistem operasi tertentu, yang berisikan aturan, fungsi serta protokol cara melakukan suatu tugas atau pekerjaan. (Baharuddin, Wakkang, & Irianto, 2022)

Peran *Website* berbasis *Web Service REST API* sangat diperlukan di dalam dunia bisnis, sebab untuk memberikan sarana kemudahan, kenyamanan, serta keamanan dalam berbisnis. Selain untuk memberikan kelancaran berbisnis, hal ini juga mempermudah adanya kejelasan dalam memberikan pertukaran informasi ketika sebuah perusahaan atau bidang usaha menemukan masalah dalam melakukan penjualan terhadap konsumen, misalkan seperti menambahkan data konsumen baru agar langsung terjadi interaksi secara langsung dan akurat antara konsumen baru dengan *marketing* tanpa harus mengganggu jalannya bisnis yang ada, proses ini juga didukung dengan adanya *Web Service* sebagai sistem antarmuka berbasis *web* yang bermanfaat bagi perusahaan serta memberikan informasi untuk mengetahui produk

dan barang apa saja yang sudah di produksi maupun harus di produksi oleh pabrik perusahaan tersebut yang terpisah dengan kantor pusat para *marketing*.

Pada 1991, Tim Berners-Lee menemukan situs *web* pertama. Pada 30 April 1993, ia mengumumkan bahwa WWW dapat digunakan secara gratis. Sejarah *Web Service* terbagi menjadi *web 1.0*, *web 2.0*, dan *web 3.0*. Pada *web 3.0*, mesin dapat membaca halaman *web* dengan menggunakan teknologi *API*. *API* memungkinkan aplikasi perangkat lunak berinteraksi.

Sistem *Web Service* berbasis *REST API* ini dapat digunakan untuk solusi dalam mengatasi resiko kesalahan dalam melakukan pertukaran data. Dalam penelitian ini yang berjudul **“Pengembangan Website Dengan Implementasi Sistem *Web Service* Menggunakan Metode *REST API* Sebagai Media Penghubung *Marketing & Customer* PT. Aura Unggul”**. Sistem ini dapat diimplementasikan agar mudah diakses serta digunakan oleh berbagai pihak, baik pihak perusahaan maupun pihak dari pelanggan itu sendiri. Dengan adanya sistem ini dapat menjadi solusi dari masalah yang telah terjadi selama ini, seperti transaksi pertukaran data, baik data pelanggan, data pemesanan barang, maupun data penagihan.

## 1.2 Identifikasi Masalah

1. Bagaimana cara sistem *Web Service REST API* ini dapat menjadi solusi dan mempermudah hubungan antar pihak seperti perusahaan dengan konsumen, divisi kantor pusat dengan kantor cabang, maupun kantor cabang dengan konsumen dengan efektif serta efisien?
2. Bagaimana cara *Web Service REST API* yang digunakan dapat dimanfaatkan oleh penjualan agar mendapatkan informasi tentang konsumen lebih cepat, efisien, tepat tanpa adanya kekeliruan data dan berlangsung secara *up-to-date*?

3. Bagaimana cara mengetahui serta menyelesaikan permasalahan yang terjadi dalam sistem yang akan digunakan secara cepat, tepat dan efektif?

### 1.3 Ruang Lingkup

Dalam pembahasan ini dapat diberikan batasan diri pada masalah-masalah yang terjadi dan akan dibahas sebagai berikut:

1. Sistem Aura Unggul *portal* dirancang dengan menggunakan metode *Web Service* berbasis *REST API*.
2. Sistem yang dirancang dapat mempermudah *marketing* untuk melakukan pembaruan biodata *customer*, maupun sebaliknya.
3. Sistem yang akan dirancang dapat mempermudah *customer* untuk melakukan pemesanan dan *upload Purchase Order*.
4. Sistem yang dirancang akan dapat melakukan proses verifikasi oleh *administrator* perihal transaksi penjualan seperti kelengkapan biodata, ketepatan pemesanan, maupun penagihan dan piutang *customer*, serta melakukan pembatasan transaksi apabila telah melewati tenggang batas pembayaran.
5. Sistem yang dirancang dapat memberikan laporan *invoice* tagihan serta jumlah piutang yang dimiliki saat ini oleh *customer* secara *real-time*.

### 1.4 Tujuan & Manfaat Penulisan

#### 1.4.1 Tujuan Penulisan

1. *Web Service* yang akan dirancang bertujuan untuk mempermudah kinerja *marketing* dalam mendapatkan informasi tentang *customer*, baik *customer* prospek maupun *customer* yang sudah *existing* secara cepat, mudah, serta efisien.

2. *Web Service* berbasis *REST API* akan dirancang menjadi *website* Aura Unggul *portal* yang akan mengatur berjalannya transaksi.
3. *Web Service* yang akan dirancang bertujuan untuk mengurangi resiko terjadinya kesalahan *input* data, kerusakan data, serta kehilangan data.
4. *Web Service* berbasis *API* ini bertujuan agar *marketing* dan juga *manager* selaku *administrator* dapat melakukan pemantauan terhadap transaksi yang sedang berjalan dan dapat mengatur pembatasan piutang serta syarat pembayaran secara *Real-time*.
5. *Web Service* berbasis *REST API* ini memiliki sistem *login multiuser* yang dimana setiap *user* akan diberikan pembatasan terhadap hak akses pada setiap menu dan *administrator* sebagai *super-user* diharuskan untuk melakukan verifikasi pada setiap transaksi oleh pelanggan.

#### **1.4.2 Manfaat Penulisan**

1. Meningkatkan kemudahan antara *marketing* untuk dapat mengetahui informasi tentang *customer*, baik *customer* prospek maupun *customer* baru.
2. Dapat mempermudah transaksi yang dilakukan karena hanya menggunakan sistem *portal* yang dibuat menjadi 1 pintu.
3. Dapat mengurangi terjadinya kesalahan-kesalahan yang dapat ditimbulkan oleh kelalaian dari sumber daya manusia (*human error*).
4. Dapat mengatasi permasalahan pembayaran yang macet, piutang yang melebihi batas, dan masalah transaksi lainnya.
5. Dapat membatasi hak akses setiap *user*; sehingga akan meminimalisir terjadinya kesalahan-kesalahan transaksi yang tidak di izinkan.

## 1.5 Teknik Pengumpulan Data

Metode pengumpulan data dan informasi yang diperlukan untuk digunakan dalam pembuatan aplikasi ini yaitu :

### 1. Wawancara

Wawancara adalah teknik pengumpulan data di mana penulis langsung bertanya kepada narasumber untuk mendapatkan informasi yang dibutuhkan, mirip dengan proses pengumpulan kebutuhan (*Requirement Elicitation*).

### 2. Observasi

Observasi adalah cara untuk mengumpulkan data dengan cara melihat dan mencatat keadaan atau perilaku objek yang sedang diamati.

### 3. Studi Pustaka

*Review literatur* (studi pustaka) adalah cara mengumpulkan data dengan membaca dan mempelajari buku, jurnal, atau penelitian yang telah dilakukan sebelumnya tentang topik yang akan dibahas dalam penelitian ini.

### 4. Studi Dokumentasi

Pada teknik pengumpulan data ini, penulis akan melakukan pengamatan serta penelitian terhadap dokumen-dokumen dari instansi yang sedang diteliti sebagai acuan dalam mengembangkan sistem yang akan dibangun pada penelitian ini nantinya.

### 5. Kuesioner

Pada tahap pengumpulan data terakhir, akan dilakukan kuisisioner dimana nanti penulis akan menyebarkan kuisisioner berupa *google form* kepada beberapa individu dalam suatu instansi untuk mempelajari pandangan dan pendapat mereka tentang aplikasi yang akan digunakan.

## 1.6 Sistematika Dalam Penulisan

Dalam penyusunan laporan ini, dibutuhkan sistematika penulisan yang bertujuan agar mempermudah dalam melakukan pemahaman penelitian. Sistematika penulisan ini terdiri dari 5 (lima) bab yang secara garis besar dapat dijelaskan dibawah ini:

### **BAB I PENDAHULUAN**

Pada bab ini dapat dijelaskan tentang latar belakang permasalahan, identifikasi, rumusan masalah, tujuan serta manfaat penulisan, ruang lingkup yang ada, serta metode dari pengumpulan data dan juga sistematika penulisan laporan.

### **BAB II LANDASAN TEORI**

Pada bab selanjutnya berisi tentang referensi dari penelitian – penelitian terdahulu yang pernah dilakukan serta rencana yang akan dilakukan. Lalu pada bab ini juga berisi tentang teori yang berkaitan dengan penulisan laporan, tinjauan studi, serta kerangka pemikiran dari penelitian ini.

### **BAB III METODOLOGI PENELITIAN**

Isi bagian ini akan menjelaskan tentang gambaran umum dari proses yang terjadi di perusahaan, diagram kegiatan dari sistem yang sedang berjalan, dokumen masukan dan keluaran, analisis masalah yang ditemui, identifikasi kebutuhan sistem, penjelasan metode yang lebih mendetail, pengumpulan kebutuhan sistem, dan jadwal penelitian.

### **BAB IV PENGUJIAN DAN IMPLEMENTASI**

Bab ini akan membahas pengujian dan implementasi sistem usulan, termasuk prosedur sistem, desain sistem (*use case diagram, activity diagram, sequence diagram*), desain basis data (*class diagram, struktur*

file), desain tampilan program, dan implementasi sistem (*screenshot* program, spesifikasi *hardware & software*, pengujian sistem).

## **BAB V    SIMPULAN DAN SARAN**

Bagian ini akan merangkum kesimpulan dari analisis dan perancangan, serta memberikan saran untuk tindakan di masa mendatang.



## BAB II

### LANDASAN TEORI

#### 2.1 Teori Umum

##### 2.1.1. Data

Data adalah informasi dasar yang digunakan untuk membuat informasi lebih lanjut. Data bisa berupa berbagai hal seperti angka, teks, suara, simbol, gambar, dan banyak lagi. Data tidak selalu berupa teks, tapi bisa juga dalam bentuk lain seperti gambar, suara, atau simbol. (Rahman & Saudin, 2022).

Data merupakan materi mentah yang digunakan untuk menciptakan informasi tertentu. Data terbentuk dari kumpulan simbol yang menunjukkan tindakan dalam suatu konteks. Simbol-simbol ini bisa berupa gambaran grafis seperti bintang (\*), tanda *ampersand* (&), atau simbol *dollar* (\$). Data ini dikelompokkan dan diatur untuk memenuhi berbagai tujuan, seperti pengolahan data, manajemen arsip, dan dasar dari suatu sistem *database*. (Davis, 2023).

Jadi, data dapat disebut sebagai fakta atau informasi mentah yang diperoleh dari berbagai sumber. Data merupakan materi yang akan diolah menjadi informasi yang bermanfaat untuk analisis dan pengambilan keputusan. Data bisa berupa angka, teks, gambar, suara, video, atau bentuk lain yang bisa diukur, direkam, dan dibuktikan kebenarannya. Pengolahan data melibatkan cara data diambil, diproses, dikontrol, dan disimpan.

##### 2.1.2. Informasi

Informasi adalah data yang telah diolah untuk memberikan gambaran lebih jelas tentang kondisi atau peristiwa yang mungkin terjadi di masa depan. Informasi ini dibuat dari data yang telah dikumpulkan dan diolah sehingga dapat

digunakan sebagai dasar dalam membuat kesimpulan, argumen, atau keputusan. (Rahman & Saudin, 2022).

Informasi adalah data yang sudah diolah agar memberikan hasil yang bisa dipahami dan bermanfaat bagi pengguna. (Dian Sano, 2020).

Menurut Muslihudin dalam bukunya, kualitas suatu informasi ditentukan oleh tiga faktor.”(Muslihudin & Oktafianto, 2016), yaitu:

1) Keakuratan Informasi

Informasi harus benar dan tidak menyesatkan, serta mudah dipahami.

2) Ketepatan Waktu Informasi

Informasi harus diterima tepat waktu, karena informasi yang sudah kadaluwarsa tidak lagi bermanfaat.

3) Relevan

Informasi harus bermanfaat dan relevan bagi setiap individu yang menggunakannya.

Jadi, informasi adalah data yang sudah diolah sehingga memiliki arti dan gunanya bagi pengguna, terutama dalam membantu pengambilan keputusan. Kualitas informasi dipengaruhi oleh tiga faktor penting: kebenaran, keakuratan waktu, dan relevansi informasi yang diberikan.

### 2.1.3. Sistem

Sistem adalah kumpulan bagian yang bekerja sama dengan aturan tertentu untuk mencapai tujuan tertentu. Mereka bekerja bersama dalam suatu jaringan untuk mencapai tujuan yang sudah ditetapkan. (Muslihudin & Oktafianto, 2016).

Sistem itu seperti tim yang bekerja bersama untuk mencapai tujuan tertentu. Mereka mengatur berbagai kegiatan sehari-hari dalam organisasi dan

juga membantu dalam membuat laporan untuk pihak luar yang membutuhkan informasi. (Afifah & Setyantoro, 2021).

Sistem dapat juga diartikan sekelompok bagian yang saling terhubung dan bekerja sama untuk mencapai beberapa tujuan dan objektif yang telah ditetapkan. (Davis, 2024).

Menurut sumber lain, sistem adalah sekumpulan bagian yang bekerja sama untuk memproses masukan dan menghasilkan keluaran. (Rusdi, Mulyani, & Herlina, 2020).

Jadi, sistem itu ibarat tim yang terdiri dari bagian-bagian yang bekerja bersama. Mereka menerima informasi, menjalankan tugas, dan menghasilkan hasil untuk mencapai tujuan yang telah ditetapkan.

#### **2.1.4. Aplikasi**

Aplikasi berasal dari kata "*application*" yang berarti digunakan untuk suatu tujuan. Aplikasi bisa digunakan untuk berbagai hal, seperti mengedit dokumen, mengatur tampilan *Windows*, atau bermain *game*. Biasanya, aplikasi terdiri dari berbagai *file* yang saling terhubung, seperti formulir, kelas, atau laporan. (Bagus Tri, 2020).

Aplikasi dapat diartikan sebagai program komputer yang dibuat untuk melakukan tugas-tugas tertentu pada perangkat seperti komputer, laptop, atau ponsel. Aplikasi digunakan untuk menjalankan perintah-perintah khusus. (Pane, Fadillah, & Zamzam, 2020).

Mengutip dari sumber lain, berikut dapat dijabarkan kutipan pengertian aplikasi dari beberapa ahli: (Kuncoro, 2022) :

- 1) Jogiyanti menjelaskan bahwa aplikasi adalah cara mengatur komputer agar bisa mengubah masukan menjadi keluaran.

- 2) Rachmad H.S, aplikasi adalah sebuah program, digunakan dalam berbagai tujuan, seperti mengelola *file*, mengatur tampilan *Windows*, dan bermain *game*.
- 3) Hengky W. Pramana mengatakan bahwa aplikasi adalah program komputer yang dibuat untuk memenuhi kebutuhan dalam berbagai aktivitas, seperti bisnis, *game*, layanan masyarakat, iklan, dan tugas-tugas lainnya.
- 4) Verman menjelaskan bahwa aplikasi adalah kumpulan instruksi khusus di komputer yang membantu kita menyelesaikan tugas-tugas tertentu.

Aplikasi itu sendiri pun dibagi menjadi beberapa jenis kelompok, yaitu :

- 1) Aplikasi Internet

Aplikasi yang membantu kita menggunakan internet, seperti Gmail, Yahoo, dan Meta..

- 2) Aplikasi Perkantoran

Aplikasi untuk pekerjaan kantor, seperti *MS Office dan Software Accurate*, yang membantu mempercepat pekerjaan.

- 3) Aplikasi Grafis

Aplikasi untuk mengedit foto, video, dan membuat desain, seperti Adobe Photoshop, Corel Draw, dan Adobe After Effects

- 4) Aplikasi *Programming*

Aplikasi khusus untuk membuat program atau *software*, seperti Visual Studio Code, Android Studio, dan Netbeans.

- 5) Aplikasi Multimedia

Aplikasi untuk menciptakan informasi dengan teks, video, dan audio, seperti Windows Media Player, VLC Studio, dan OBS Studio.

## 6) Aplikasi *Games*

Aplikasi berisi permainan untuk hiburan semata.

### 2.1.5. *Internet*

*Internet* adalah jaringan komputer global yang menghubungkan seluruh dunia tanpa batasan wilayah, hukum, atau budaya. (Nurbaiti & Alfarisyi, 2023)

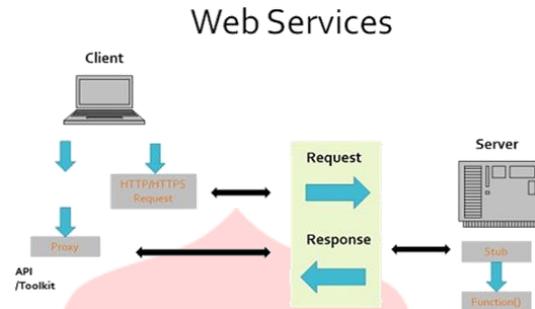
*Internet* adalah teknologi yang memungkinkan kita terhubung ke seluruh dunia melalui komputer. Ini adalah jaringan komunikasi global yang menghubungkan banyak komputer menggunakan berbagai metode seperti telepon dan satelit, memungkinkan komunikasi dan akses informasi dari seluruh dunia. (Nurbaiti & Alfarisyi, 2023)

*Internet* telah merasuk ke berbagai aspek kehidupan masyarakat *modern*, dari pekerjaan, pendidikan, militer, kesehatan, hingga aktivitas sosial sehari-hari. Sektor publik, swasta, dan privat juga tidak bisa terlepas dari teknologi internet ini. (Nurbaiti & Alfarisyi, 2023)

*Internet* adalah alat yang sering digunakan untuk terhubung ke komputer di seluruh dunia dan untuk berkomunikasi. *Internet* memiliki banyak manfaat di berbagai bidang seperti bisnis, pendidikan, pemerintahan, dan lainnya. Keunggulan *internet* sebagai media komunikasi dan informasi adalah kemudahan, kecepatan, dan hemat biaya dalam mendapatkan informasi secara global. *Internet* berbeda dengan media komunikasi lain karena penggunaannya dan karakteristik uniknya sebagai media komunikasi. (Nurbaiti & Alfarisyi, 2023)

## 2.2 Teori Khusus

### 2.2.1. Web Service



**Gambar 2.1** *Web Service Schema* (Sumber: Intern, 2021)

*Web Service* adalah teknologi *internet* yang memungkinkan komunikasi *program-to-program*, memungkinkan akses layanan *web* dari berbagai *platform* seperti *desktop*, *website*, dan aplikasi *mobile*. (Nugroho & Primadewi, 2021)

*Web Service* adalah kumpulan fungsi yang bisa diakses melalui *Internet* untuk aplikasi sistem yang tersebar. Ini menyediakan informasi, logika aplikasi, dan dapat diakses dari berbagai platform melalui HTTP dan XML. (Ikhwanuzaki & Handayani, 2024)

*Web Service* juga dapat diartikan sebagai aplikasi yang berisi sekumpulan *database* dan *software* atau bagian dari program yang diakses secara *remote* oleh piranti dengan perantara tertentu, yang memungkinkan pengguna agar dapat mengatasi permasalahan *interoperability* dan integrasi sistem yang berbeda. (Intern, 2021)

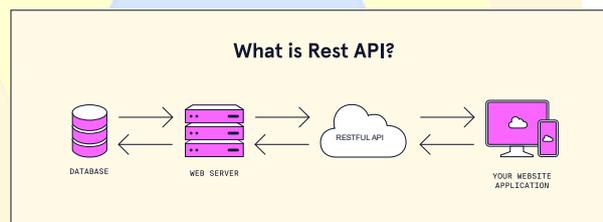
*Web Service* adalah perangkat lunak yang memungkinkan sistem-sistem berbeda berkomunikasi tanpa masalah, meskipun mereka berbeda *platform* dan tidak langsung terhubung ke basis data. (Baharuddin dkk., 2022) *Web Service* mendukung sistem terdistribusi yang berjalan di infrastruktur berbeda. Dengan XML dan HTTP, implementasinya juga mudah. (Ikhwanuzaki & Handayani, 2024)

Dengan teknologi *Web Service*, pengguna bisa menghubungkan berbagai jenis perangkat lunak yang berbeda platform dan sistem operasinya. Ini memungkinkan pengguna untuk tetap mendapatkan informasi dari berbagai situs tanpa harus berkunjung ke situs-situs dengan *platform* dan sistem operasi yang berbeda. (Baharuddin, Wakkang, & Irianto, 2022)

Ada tiga cara standar yang digunakan oleh konsorsium web untuk pertukaran informasi melalui Web Service, yaitu XML-RPC, SOAP, dan REST. (Nugroho & Primadewi, 2021)

Kesimpulannya ialah *web service* merupakan sebuah aplikasi yang memiliki peran fungsi utama untuk melakukan pertukaran data dengan menggunakan *internet* yang dapat memungkinkan adanya interaksi antar aplikasi.

### 2.2.2. REST



**Gambar 2. 2** Contoh cara kerja *REST API* (Sumber: Team, 2024)

*REST* adalah cara bagi sistem untuk berkomunikasi melalui internet dengan menggunakan gaya arsitektur yang fokus pada penggunaan *hyperlink* dan transfer representasi sederhana. Dalam *REST*, *client* mengirim permintaan ke *server*; *server* memproses permintaan tersebut, dan mengirimkan kembali *respons*. Tiap permintaan dianggap sebagai transaksi independen dan tidak tergantung satu sama lain (*stateless*), sehingga aplikasi *REST* menjadi sederhana dan ringan. (Wardhana, Arwani, & Rahayudi, 2020)

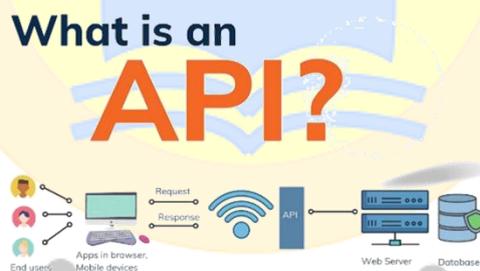
*REST* adalah sekumpulan prinsip-prinsip desain untuk membangun sistem komunikasi jaringan yang fokus pada definisi dan pengelolaan sumber daya. Secara umum, istilah ini sering digunakan untuk menggambarkan antarmuka yang sederhana untuk mengirimkan dan menerima data dalam domain tertentu melalui protokol HTTP. (Baharuddin dkk., 2022)

*Restful Web Service* adalah aplikasi *web* yang mengikuti arsitektur *REST*. (Wardhana dkk., 2020). *REST* memiliki *latency* terendah dalam beberapa kondisi pengujian. (Nugroho & Primadewi, 2021b)

*RESTful Web Service* menggunakan metode *HTTP* seperti *GET*, *POST*, *PUT*, dan *DELETE* untuk melakukan berbagai operasi terkait sumber daya. (Wardhana dkk., 2020)

*RESTful API* adalah salah satu jenis arsitektur *API* yang menggunakan *HTTP* sebagai media pertukaran data. (Ikhwanuzaki & Handayani, 2024)

### 2.2.3. *API*



**Gambar 2. 3** Ilustrasi cara kerja *API* dalam jaringan (Sumber: Rawat, 2021)

*API* memungkinkan sistem-sistem berbeda untuk berkomunikasi. Terdiri dari *server* yang menyediakan *API* dan *client* yang mengelola data sesuai permintaan pengguna. (Wardhana dkk., 2020)

Dalam teknologi *REST*, *API* digunakan untuk komunikasi antar sistem. Terdiri dari *server* yang menyediakan data dan *client* yang meminta data. *REST*

API adalah jenis API berbasis *website* yang menggunakan REST dan JSON. (Wardhana dkk., 2020)

*API* adalah seperangkat teknik dasar yang digunakan oleh programmer untuk membuat perangkat lunak di sistem operasi tertentu. Berisi aturan, fungsi, dan protokol untuk menjelaskan cara melakukan tugas. Biasanya berupa penjelasan tentang fungsi atau rutin yang bisa digunakan dalam bahasa pemrograman seperti C. (Baharuddin dkk., 2022)

Dalam konteks *web*, *API* digunakan untuk memanggil fungsi melalui HTTP dan mendapatkan respons dalam bentuk XML atau JSON. *Respons* yang diterima dari pemanggilan fungsi dapat bervariasi tergantung dari penyedia *API* yang mengatur respons tersebut. (Baharuddin dkk., 2022)

Penggunaan konsep *API web service* memberikan manfaat bagi pengguna karena memudahkan akses data. Contohnya, jika ada perubahan data pada sistem *website* yang memiliki *web service*, maka data di aplikasi *Android* juga akan berubah otomatis. Dengan adanya *API*, orang lain tidak dapat langsung mengakses *database* kita, melainkan mereka harus melakukan permintaan melalui internet, mengakses *API* kita, dan kemudian menerima *respons* dari *API* kita. (Baharuddin dkk., 2022).

#### **2.2.4. Website**

*Website* adalah kumpulan halaman *web* yang terhubung dan diakses melalui *internet*. Berisi berbagai konten seperti teks, gambar, video, audio, formulir, dan dapat diakses melalui URL menggunakan *web browser*. (Kurniawan dkk., 2023)

*Website* adalah kumpulan halaman *web* yang telah dipublikasikan di internet dengan URL/*domain* yang dapat diakses oleh pengguna internet dengan mengetik halamannya. (Sofyan, Asia, & Quraisy, 2022)

*Website* adalah kumpulan halaman yang menampilkan informasi teks, gambar, animasi, suara, statis, atau dinamis. Halaman-halaman ini saling terkait dan dihubungkan dalam satu rangkaian di jaringan. (A. B. Santoso, 2022)

*Website* dapat dikategorikan dalam beberapa jenis, yaitu (Sari, Abdilah, & Sunarti, 2019):

1. *Web Statis*

Merupakan *website* yang memiliki halaman yang tidak berubah, dan perubahan yang dilakukan biasanya secara *manual* dengan mengedit *code* struktur dari *website* tersebut.

2. *Web Dinamis*

*Website* jenis ini memiliki struktural yang diperuntukan untuk dapat diupdate sesering mungkin. Biasanya *website* jenis ini memiliki halaman *backend* untuk melakukan perubahan konten dari *website* tersebut seperti contoh; *web portal*, *web* berita, dll.

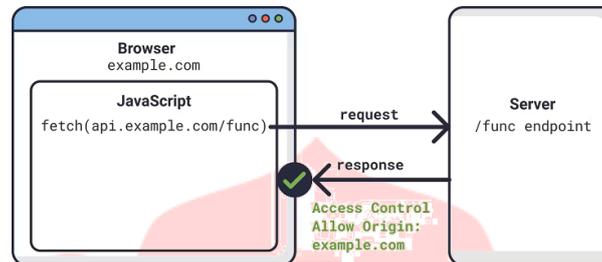
3. *Web Interaktif*

Pada jenis ini, *website* yang akan berinteraksi antara penggunanya, biasanya berupa forum diskusi maupun blog yang dimana adanya moderator sebagai pengatur alur diskusi.

Maka dari itu *website* dapat diartikan sebagai sekumpulan halaman *web* yang saling terkait satu sama lain, berisi berbagai macam konten seperti informasi teks, gambar bergerak ataupun tidak, video, suara, animasi, maupun

gabungan kesemua konten, yang dipublikasikan dan dapat diakses melalui internet dengan menggunakan *domain/URL (Uniform Resource Locator)*.

### 2.2.5. CORS (*Cross-Origin Resource Sharing*)



**Gambar 2. 4** Ilustrasi CORS (Sumber: Priwanto, 2023)

CORS merupakan sebuah mekanisme yang berfungsi untuk mengelola perizinan serta membatasi permintaan sumber daya dari *domain* yang berbeda.(Johns, Müller, Wirth, & Prasser, 2021). CORS juga dapat diartikan dengan suatu kebijakan keamanan yang diterapkan oleh *web browser* untuk melindungi penggunaannya dari serangan lintas *domain*. Contohnya, CORS *origin* memiliki fungsi dalam mengatur sumber daya dari *web server (javascript, font. Gambar)*, agar dapat diakses atau melakukan akses pada halaman *web* di *domain* lain (Priwanto, 2023).

*Cross-Origin Resource Sharing (CORS)* memiliki beberapa tujuan dan manfaat yaitu: (Johns dkk., 2021)

1. **Keamanan**, CORS dirancang untuk meningkatkan keamanan aplikasi *web* dengan membatasi akses ke sumber daya dari *domain* yang berbeda. Dengan mengizinkan hanya permintaan dari domain tertentu (dari *Gateway*), CORS membantu mencegah potensi serangan seperti *Cross-Site Request Forgery (CSRF)* dan memastikan bahwa hanya aplikasi yang terpercaya yang dapat mengakses data.

2. **Interoperabilitas**, CORS memungkinkan aplikasi *web* yang berjalan di satu *domain* untuk berinteraksi dengan *API* yang berada di domain lain. Ini sangat penting dalam konteks portal yang mengintegrasikan berbagai *data warehouse*, karena pengguna dapat mengakses data dari berbagai sumber tanpa terhalang oleh batasan *domain*.
3. **Pengalaman yang lebih baik**, Dengan mengizinkan akses lintas *domain*, CORS memungkinkan portal untuk memberikan pengalaman pengguna yang lebih mulus dan terintegrasi. Pengguna dapat mengakses berbagai layanan dan data dari satu antarmuka tanpa harus menghadapi masalah terkait pembatasan *domain*.
4. **Fleksibilitas dalam pengembangan**, CORS memberikan fleksibilitas bagi pengembang untuk merancang aplikasi yang dapat berkomunikasi dengan berbagai layanan *backend*. Ini memungkinkan pengembang untuk membangun aplikasi yang lebih kompleks dan terdistribusi, yang dapat memanfaatkan berbagai sumber daya dan layanan yang tersedia di internet.

#### 2.2.6. SRM & CRM

*Supplier Relationship Management (SRM)* merupakan sebuah sistem yang digunakan untuk mengelola hubungan perusahaan dengan pemasok atau *supplier*, yang bertujuan untuk menciptakan dan mempertahankan hubungan yang saling menguntungkan antara perusahaan dengan pemasok. (Putra, 2024)

Dengan mengimplementasikan *Supplier Relationship Management (SRM)*, perusahaan akan mendapatkan berbagai macam manfaat yang ada sebagai contohnya dapat meningkatkan efisiensi dalam manajemen hubungan *supplier* (pemasok), mengurangi biaya operasional, serta mencapai keunggulan

dalam kompetitif di dalam pasar. SRM juga dapat membantu dalam meningkatkan kualitas barang ataupun jasa yang digunakan di dalam hubungan transaksi antara pelanggan dengan pemasok. (Putra, 2024)

*Customer Relationship Management* (CRM), merupakan sistem yang digunakan untuk mengelola dan menganalisis hubungan dengan pelanggan (*Customer*). Aplikasi ini dapat digunakan oleh perusahaan untuk membantu mengatur data pelanggan, melacak interaksi yang terjadi, meningkatkan pemasaran, serta menganalisis perilaku pelanggan. (Viktor, 2024)

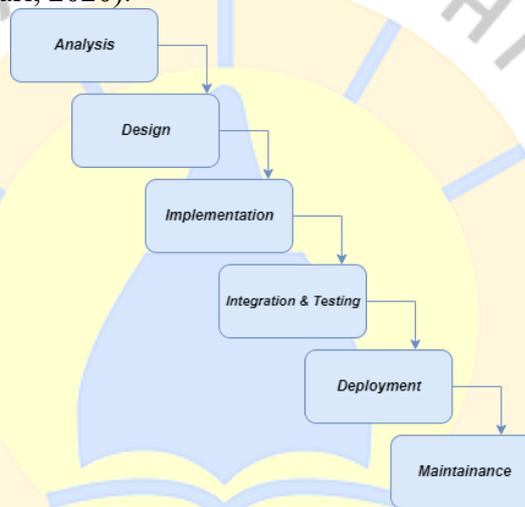
Perbedaan utama antara CRM dengan SRM dapat dijabarkan sebagai berikut: (Viktor, 2024)

- a. **Fokus** : Sasaran utama SRM untuk optimalisasi kinerja pemasok dan pengelolaan hubungan sedangkan CRM untuk pengelolaan hubungan pelanggan serta peningkatan keterlibatan.
- b. **Data** : pada sistem CRM memiliki data berupa informasi kontak, pembelian dan preferensi, sedangkan SRM menyimpan ketentuan, harga, dan jadwal pengiriman untuk pemasok.
- c. **Tujuan** : CRM memiliki tujuan utama untuk meningkatkan nilai umur pelanggan, loyalitas, dan retensi; Sedangkan SRM untuk meningkatkan ketergantungan pemasok, kualitas serta produktivitas.
- d. **Fungsionalitas** : SRM biasanya menyediakan evaluasi pemasok, administrasi, kontrak, dan pengadaan alat; sedangkan CRM sering kali menggabungkan kemampuan seperti manajemen prospek, otomatisasi penjualan, serta portal dukungan pelanggan.

## 2.3 Teori Analisa & Perancangan

### 2.3.1. Teori Metodologi Penelitian *Waterfall*

*Waterfall* atau dapat disebut juga dengan model air terjun, merupakan pendekatan model *classic* dalam mengembangkan perangkat lunak dengan menggambarkan metode pengembangan secara linear dan berurutan, biasanya terdiri antara lima hingga tujuh fase yang dimana setiap fasenya didefinisikan dengan tugas dan tujuan yang berbeda-beda, di mana setiap fase menggambarkan siklus hidup perangkat lunak hingga ke pengirimannya (Hasanah & Untari, 2020).



**Gambar 2. 5** Metode SDLC (Sumber: Hartono, 2021)

Mengutip dari buku “Cara mudah dan Cepat Belajar Pengembangan Sistem Informasi”, *waterfall* adalah pengerjaan suatu sistem yang dilakukan dengan beruntun dan secara linear. Dikerjakan dalam beberapa langkah yang dilaksanakan secara bertahap dan apabila ada gangguan dari suatu langkah, maka langkah selanjutnya tidak dapat dikerjakan (Hartono, 2021).

Menurut informasi yang tertuang didalam buku “Rekayasa Perangkat Lunak”, bahwa secara garis besar metode *waterfall* memiliki langkah-langkah seperti ; *Requirement Gathering and analysis*, Desain, Implementasi, *Integration*

& *Testing*, Verifikasi, *Operation & Maintenance*. Langkah-langkah tersebut juga akan dijabarkan sebagai berikut (Hasanah & Untari, 2020) :

1. **Requirement Gathering and analysis**, yaitu tahap untuk mengumpulkan kebutuhan secara lengkap yang kemudian akan dianalisis dan didefinisikan kebutuhannya yang harus terpenuhi oleh aplikasi atau program yang akan dibangun.
2. **Desain**, yaitu tahap dimana pengembang akan menghasilkan sebuah sistem secara keseluruhan serta menentukan alur perangkat lunak hingga algoritma yang detail.
3. **Implementasi**, merupakan tahapan dimana seluruh desain yang telah terbentuk, diubah menjadi bahasa pemrograman. Bahasa pemrograman atau kode-kode program yang dihasilkan masih berupa modul-modul yang nantinya akan diintegrasikan menjadi sebuah sistem yang lengkap.
4. **Integration & Testing**, pada tahap ini pengembang akan melakukan penggabungan modul-modul yang sudah dibuat, lalu melakukan pengujian untuk mengetahui apakah *software* yang dibuat telah sesuai dengan desainnya dan apakah ada kesalahan pada fungsi perangkat lunak (*software*) yang telah di gabungkan.
5. **Verifikasi**, merupakan tahap dimana klien atau pengguna melakukan pengujian terhadap sistem tersebut apakah sesuai dengan yang setuju atau belum sesuai.
6. **Operation & Maintenance**, apabila semua langkah sebelumnya tidak ada kendala, maka langkah selanjutnya yaitu instalasi dan proses perbaikan sistem sesuai yang disetujui.

### 2.3.2. Teori Pemodelan Objek

Teori Pemodelan Objek atau dapat disebut juga dengan *Unified Modeling Language* (UML) merupakan bahasa yang digunakan untuk membangun, menentukan, mendokumentasikan serta menentukan sistem informasi yang dibutuhkan, serta dapat juga digunakan untuk memahami dan mendokumentasikan setiap sistem informasi yang ada (Aditya, Pranatawijaya, & Putra, 2021)

Berdasarkan sumber lainnya, UML merupakan istilah Bahasa yang menjadi standar dalam industry untuk dapat visualisasi, perancangan serta dokumentasi pada perangkat lunak dengan menggunakan gabungan beberapa elemen tidak praktis ditambah dengan elemen-elemen dari metode lain yang lebih efektif dan juga elemen-elemen baru yang belum ada dalam metode yang terdahulu, sehingga UML (*Unified Modelling Language*) lebih ekspresif dan seragam dibandingkan metode lainnya (Arafat, Agustian, Frindo, & Oktavia, 2022)

Adapun UML dapat digunakan untuk tujuan tertentu, seperti (Arafat dkk., 2022):

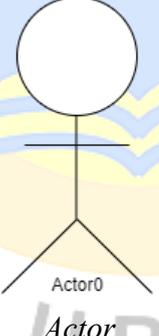
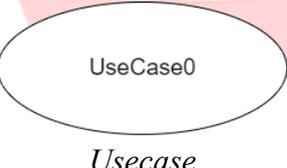
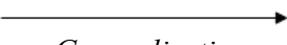
- a. Membuat *software*
- b. Cara *software* berkomunikasi dengan proses bisnis.
- c. Membahas secara detail untuk menganalisis dan mencari kebutuhan sistem.
- d. Membuat Dokumentasi untuk sistem dan proses dalam organisasi.

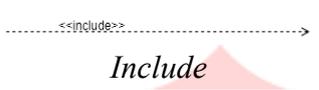
### 2.3.2.1. Diagram Use Case

Diagram use case adalah gambaran dalam bentuk diagram yang menunjukkan hubungan dari sistem yang sedang dikembangkan. Diagram ini digunakan untuk mengembangkan perangkat lunak ataupun sistem untuk kebutuhan fungsional dari sistem tersebut. Diagram use case berfokus pada apa yang sistem lakukan dan bukan bagaimana cara sistem melakukannya. (Arafat dkk., 2022).

*Usecase Diagram* dapat juga diartikan sebuah model untuk mempelajari kelakuan pada sistem yang akan dibuat. *Usecase* dapat digunakan untuk menjelaskan seluruh interaksi yang terjadi, dapat satu atau lebih *actor* dengan sistem yang dibuat. (Aditya dkk., 2021).

**Tabel 2. 1** Komponen *Usecase Diagram*

No	Nama Komponen & Gambar	Fungsi
1		<i>Actor</i> adalah orang, benda, atau sesuatu yang berinteraksi langsung dengan sistem komputer. Tugasnya adalah memberikan informasi kepada sistem dan memungkinkan sistem untuk melakukan tugasnya.
2		<i>Usecase</i> adalah gambaran dari suatu fungsi tertentu dalam sistem komponen. Setiap <i>usecase</i> adalah urutan dari hubungan transaksi oleh actor dan sistem.
3		<i>Association</i> ialah simbol penghubung antara <i>actor</i> dengan <i>usecase</i> .
4		<i>Generalization</i> merupakan penunjukan spesialisasi <i>actor</i> untuk dapat berpartisipasi dengan <i>usecase</i> .

5		Penggunaan " <i>Extend</i> " dalam <i>s</i> digunakan untuk menunjukkan bahwa suatu fitur tambahan dari suatu use case akan terjadi jika kondisi tertentu terpenuhi.
6		<i>Include</i> berfungsi untuk menunjukkan bahwa suatu <i>usecase</i> seluruhnya merupakan fungsionalisasi dari <i>usecase</i> lainnya.

(Sumber: Arafat dkk., 2022)

### 2.3.2.2. *Activity Diagram*

*Activity Diagram* merupakan gambaran aliran rangkaian aktifitas baik proses bisnis atau *usecase*, menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem dan bukan apa yang dilakukan oleh aktor. (Tarigan & Yunita, 2021)

Sedangkan menurut sumber lain, *Activity Diagram* merupakan diagram yang memodelkan proses yang terjadi dalam sebuah sistem, digambarkan secara vertical, dan pengembangan dari *usecase* yang memiliki alur aktifitas. (Arafat dkk., 2022)

*Activity Diagram* digunakan sejajar dengan diagram use case dan state untuk pemodelan. Bisa menganalisis diagram use case dengan deskripsi actor, tindakan, dan waktu terjadinya. (Arafat dkk., 2022)

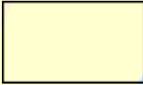
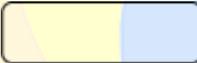
*Activity Diagram* memiliki beberapa fungsi dan tujuan, yaitu: (Arafat dkk., 2022)

1. Dalam dunia bisnis biasanya digunakan untuk memperlihatkan urutan proses pada bisnis
2. Merupakan metode perancangan yang terstruktur dan mirip dengan *flowchart* maupun *dataflow diagram*

3. Dapat menjelaskan urutan aktifitas dalam suatu proses
4. Dapat mengetahui aktifitas actor atau pengguna berdasarkan *usecase diagram* yang telah dibuat sebelumnya
5. Mempermudah memahami proses yang ada dalam sistem secara keseluruhan.

Menurut sumber lain, *Activity Diagram* memiliki beberapa simbol yang sudah di rangkum dalam sebuah tabel berikut : (Tarigan & Yunita, 2021)

**Tabel 2. 2** Simbol *Activity Diagram*

SIMBOL	FUNGSI
 <i>Activity</i>	<i>Activity</i> memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
 <i>Action</i>	<i>Action</i> merupakan simbol aktivitas <i>state</i> dari sistem yang mencerminkan eksekusi dari suatu aksi.
 <i>Activity Final Node</i>	<i>Activity Final Node</i> merupakan simbol aktivitas yang menggambarkan bagaimana objek dibentuk dan dihancurkan.
 <i>Initial Node</i>	<i>Initial Node</i> merupakan simbol dari bagaimana objek dibentuk atau diawali.
 <i>Fork Node</i>	<i>Fork Node</i> merupakan simbol dari suatu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.

(Sumber : Tarigan & Yunita, 2021)

### 2.3.2.3. *Class Diagram*

*Class Diagram* adalah gambar grafis yang menunjukkan struktur objek statis sistem, termasuk kelas objek dan hubungan antara kelas-kelas tersebut. (Tarigan & Yunita, 2021)

Diagram kelas adalah model statis yang menunjukkan kelas-kelas dan hubungan antar kelas yang konstan dalam sistem dari waktu ke waktu. Ini mencakup perilaku, status, dan hubungan antar kelas dengan elemen pendekatan yang berbeda untuk menyederhanakan diagram kelas dan alternatif struktur sistem. (J. T. Santoso & Migunani, 2021)

Menurut salah satu jurnal, Diagram Kelas (*Class Diagram*) dalam UML digunakan untuk menampilkan kelas atau paket-paket dalam sistem. Ini menggambarkan struktur statis sistem dengan menunjukkan *class-class* dan hubungannya secara logika. (Aditya dkk., 2021)

Kelas dalam diagram kelas memiliki atribut (*variabel*) dan metode (fungsi). Atribut adalah *variabel* kelas, sedangkan metode adalah fungsi kelas. Berikut merupakan tabel beberapa simbol diagram kelas berdasarkan beberapa referensi yang ada :

**Tabel 2. 3** Simbol *Class Diagram*

SIMBOL	FUNGSI
 <i>Generalization</i>	<i>Generalization</i> adalah hubungan di mana objek anak membagi perilaku dan struktur data dari objek orang tua.
 <i>Nary Association</i>	<i>Nary Association</i> membantu menghindari asosiasi dengan lebih dari 2 objek.

 <i>Realization</i>	<i>Realization</i> digunakan untuk menunjukkan operasi yang dilakukan oleh suatu objek.
 <i>Association</i>	<i>Association</i> adalah simbol yang menghubungkan objek satu dengan objek lainnya.
 <i>Class</i>	<i>Class</i> adalah himpunan objek yang memiliki atribut dan operasi yang sama.
 <i>Collaboration</i>	<i>Collaboration</i> digunakan untuk mendeskripsikan urutan aksi sistem dan menghasilkan output yang terukur untuk seorang pengguna.
 <i>Dependency</i>	<i>Dependency</i> menunjukkan hubungan di mana perubahan pada satu elemen akan mempengaruhi elemen lain yang bergantung padanya.

(Sumber: Tarigan & Yunita, 2021)

#### 2.3.2.4. *Sequence Diagram*

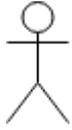
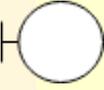
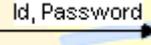
*Sequence Diagram* adalah diagram yang menunjukkan interaksi antara objek dan komunikasi mereka, serta menjelaskan perilaku pada skenario dengan menggambarkan bagaimana entitas dan sistem berinteraksi melalui pesan. (Aditya dkk., 2021)

*Sequence Diagram* adalah diagram yang digunakan untuk menjelaskan interaksi antar objek dalam sistem secara terperinci, menampilkan pesan yang dikirim beserta waktu pelaksanaannya.

(Sumirat, Cahyono, Kristyawan, & Kacung, 2023)

Berikut dapat dijabarkan beberapa komponen utama dalam diagram sequence

**Tabel 2. 4** Simbol-simbol *Sequence Diagram*

Simbol	Keterangan
 <p>Actor <b>Aktor</b></p>	<p>Aktor adalah representasi seorang pengguna di luar sistem yang sedang berinteraksi dengan sistem, digambarkan dengan simbol <i>stick figure</i>.</p>
 <p><b>Activation Box</b></p>	<p><i>Activation Box</i> merepresentasikan waktu yang diperlukan oleh suatu objek untuk menyelesaikan tugasnya dalam diagram urutan. Semakin lama waktu, <i>Activation Box</i> akan semakin panjang.</p>
 <p><b>Lifeline</b></p>	<p><i>Lifeline</i> digambarkan dengan garis putus-putus dan kotak berisi objek untuk menggambarkan aktivitas dari objek dalam diagram urutan.</p>
 <p><b>Objek</b></p>	<p>Objek didokumentasikan dengan simbol dan nama serta digunakan untuk merepresentasikan perilaku objek dalam sistem, seperti <i>Boundary</i>, <i>Entity</i>, dan <i>Actor</i>.</p>
 <p><b>Messages</b></p>	<p><i>Messages</i> digunakan untuk menggambarkan komunikasi antar objek dalam diagram urutan melalui anak panah pada <i>lifeline</i>. Mereka merupakan inti dari sebuah diagram urutan.</p>

(Sumirat dkk., 2023)

Jadi, dapat disimpulkan bahwa *Sequence Diagram* adalah diagram yang merepresentasikan interaksi antara objek dalam sistem, menunjukkan pesan atau perintah yang dikirim serta urutan waktu pelaksanaannya.

### 2.3.3. Teori Aplikasi

#### 2.3.3.1. Postman

Postman adalah platform kolaborasi untuk pengembangan *API* yang dapat digunakan sebagai *client* untuk mengakses *REST* langsung, melakukan pengujian otomatis, simulasi *endpoint*, dokumentasi *API*, pemantauan performa *API*, dan berbagi konteks dalam workspace untuk membangun dan menggunakan *API* secara *real-time*. (Wardhana dkk., 2020)

#### 2.3.3.2. MySQL

MySQL adalah tempat di mana kita bisa menyimpan data, dan dapat digunakan bersama dengan *script* PHP (Tumini & Fitria, 2021). MySQL singkatan dari *Structured Query Language* (SQL), yaitu bahasa yang digunakan untuk mengolah *database*. (Nirsal, Rusmala, & Syafriadi, 2020)

MySQL adalah perangkat lunak untuk mengelola basis data dengan bahasa SQL. Dikembangkan oleh Michael "Monty" Widenius pada 1979, MySQL memungkinkan pengaksesan dan penggunaan data oleh banyak pengguna secara bersamaan. *Software* ini membuat pengelolaan data menjadi lebih efisien dengan operasi data yang mudah dan otomatis. (Suhartini, Sadali, & Putra, 2020)

MySQL adalah jenis sistem basis data yang memisahkan data ke dalam beberapa tabel terpisah. Dengan cara ini, pengolahan data menjadi lebih cepat dan efisien. (Nirsal dkk., 2020) MySQL memiliki fitur query atau bahasa SQL yang mudah dipahami dan menggunakan karakter escape yang sama dengan PHP. Selain itu, MySQL diakui

sebagai salah satu *database* dengan kecepatan tertinggi saat ini. (Tumini & Fitria, 2021)

#### 2.3.3.3. XAMPP

XAMPP adalah *software* gratis yang dapat digunakan di berbagai sistem operasi. Ini termasuk program seperti Apache HTTP *Server*, MySQL *database*, PHP, dan Perl. XAMPP berfungsi sebagai *server* yang dapat berjalan di komputer Anda (*localhost*). (Siallagan & Wisnu, 2020)

XAMPP merupakan kombinasi X (empat sistem operasi), Apache, MySQL, PHP, dan Perl dalam satu *tool* yang mudah digunakan (Nirsal dkk., 2020). Program ini gratis dengan lisensi GNU, merupakan *web server* mudah digunakan yang dapat menampilkan halaman web dinamis. (Siallagan & Wisnu, 2020)

XAMPP adalah paket lengkap yang mencakup Apache (*web server*), MySQL (*database*), PHP (*serverside scripting*), Perl, FTP *server*, PhpMyAdmin, dan pustaka bantu lainnya. Populer digunakan oleh pemula untuk belajar PHP secara mandiri. (Tumini & Fitria, 2021)

#### 2.3.3.4. Visual Studio

*Visual Studio* adalah *software* pengembangan lengkap yang dapat digunakan untuk menyelesaikan seluruh tahap pembuatan program dalam satu tempat. Ini merupakan lingkungan pengembangan terpadu (IDE) yang menyediakan fitur untuk menulis, mengedit, men-debug, dan membangun kode program, serta menyebarkan aplikasi. Selain itu, Visual Studio juga menyertakan

berbagai fitur seperti compiler, alat bantu penulisan kode, kontrol versi, ekstensi, dan banyak lagi untuk mempermudah pengembangan perangkat lunak. (Microsoft, 2023)

Dengan bantuan fitur dan dukungan bahasa yang beragam di Visual Studio, pengguna dapat mulai dari menulis program sederhana seperti "*Hello World*" hingga mengembangkan serta menyebarkan aplikasi yang kompleks. Contohnya, pengguna bisa membangun, melakukan *debugging*, dan menguji aplikasi .NET dan C++, mengedit halaman web ASP.NET menggunakan tampilan desain web, mengembangkan aplikasi untuk perangkat mobile dan desktop dengan .NET yang bisa berjalan di berbagai platform, atau membuat antarmuka web responsif menggunakan C#.(Microsoft, 2023)

Visual Studio adalah sebuah perangkat lunak yang memberikan kepada para pengembang sebuah lingkungan pengembangan dengan berbagai fitur yang membantu mereka dalam menulis kode berkualitas tinggi dengan efisien dan kolaboratif.

1. Peningkatan berbasis beban kerja - menginstal hanya komponen yang diperlukan.
2. Alat dan fitur pengkodean yang kuat - menyediakan segala yang dibutuhkan untuk membangun aplikasi dalam satu tempat.
3. Mendukung berbagai bahasa - pengguna dapat menulis kode dalam C++, C#, *JavaScript*, *TypeScript*, *Python*, dan lainnya.
4. Pengembangan lintas *platform* - memungkinkan pengembangan aplikasi untuk berbagai *platform*.

5. Integrasi dengan kontrol versi - memungkinkan kolaborasi dalam pengembangan kode dengan tim.
6. Pembantu pengembangan AI - membantu pengguna menulis kode dengan lebih efisien menggunakan teknologi kecerdasan buatan.

#### 2.3.3.5. *FileZilla*

*FileZilla* adalah program yang digunakan untuk mengirim file melalui internet atau jaringan komputer menggunakan protokol FTP. (Maulana, Sungkar, & Harimadi, 2020)

*FileZilla* merupakan salah satu *software client* FTP (*FTP client*) yang terbilang populer dan banyak digunakan oleh para developer, baik yang profesional hingga pemula sekalipun. *FileZilla* dapat digunakan secara gratis karena ini merupakan *software* yang bersifat *open-source* dan *cross- platform*. Walau demikian, *FileZilla* juga menyediakan versi Pro atau berbayar dengan fitur yang lebih komprehensif. (Gio, 2023)

*Software* ini hadir untuk memudahkan pengguna dalam mengunggah, mengunduh, dan mengelola beragam *file* dengan segala ukuran dari *web hosting* yang digunakan, tanpa perlu melalui *control panel* terlebih dahulu. Tak hanya itu, *developer* juga dapat mengedit dan menyimpan perubahan pada *file* tanpa perlu mengunduh dan mengunggahnya secara manual. (Gio, 2023)

Tak hanya terkenal karena tampilannya yang *user friendly*, *FileZilla* juga mendukung berbagai sistem operasi. Hal inilah yang membuat *software* ini populer di kalangan *FTP client*. Selain itu,

FileZilla juga memiliki beragam fungsi dan fitur, antara lain (Gio, 2023) :

### **1. Site Manager**

*Site manager* merupakan fitur yang berfungsi untuk menghubungkan situs ke FileZilla. Secara lengkapnya, fitur ini hadir untuk membantu developer dalam mengelola dan membuat daftar situs FTP dan data koneksinya, seperti nomor *port* yang akan digunakan, protokol, dan lain sebagainya. Untuk menghubungkan situs ke FileZilla bisa dilakukan dengan mengakses site manager kemudian mengeklik Ctrl+S pada Windows dan CMD+S pada Mac.

### **2. Message Log**

Fitur yang ditampilkan pada bagian atas jendela ini menampilkan *output console-type* yang menunjukkan perintah yang dikirim oleh FileZilla dan respon yang diterima dari *server*.

### **3. File and Folder View**

*File and Folder View* merupakan fitur yang menyajikan tampilan grafis antarmuka untuk FTP. Fitur ini terletak pada bagian bawah pesan *log* atau *message log*. Melalui fitur ini, pengguna dapat menavigasi *folder*, melihat, dan mengubah isinya pada komputer lokal dan *server*. Selain itu, pengguna juga dapat melakukan *drag* dan *drop file* yang dilakukan antara komputer lokal dan *server*.

#### 4. *Transfer Queue*

Fitur transfer *queue* atau antrian transfer merupakan fitur yang berada pada bagian bawah jendela. Fitur ini menampilkan status *real-time* dari setiap antrian atau transfer *file* yang aktif.

#### 5. Menjalankan Beberapa *Transfer Server* Secara Bersamaan

Fitur unggulan FileZilla yang menjadi favorit para *developer* adalah kemampuannya untuk menjalankan beberapa transfer *server* secara bersamaan. Walau demikian, hal ini tetap kembali lagi kepada kecepatan internet dan jumlah *file* pada proses transfer. Untuk mengoptimalkan proses ini, pengguna dapat membatasi jumlah koneksi *server* simultan dari menu *server*.

#### 6. *Support SFTP – SSH FTP*

Sebagian orang mungkin menganggap bila beberapa aplikasi FTP kurang memperhatikan keamanan karena memiliki pengaturan kredensial yang *default*. Namun, dengan SSH FTP (SFTP) artinya *FileZilla* dapat digunakan untuk transfer *file* dengan koneksi aman yang terenkripsi.

Sementara pada , pengguna dapat mentransfer *file* ke aplikasi pihak ketiga, seperti Google Drive, Dropbox, One Drive, Amazon S3, dan lain sebagainya. Kemudian, filter koneksi berdasarkan alamat IP.

#### 2.3.3.6. PHP

*Hypertext PreProcessor* atau yang dapat disebut juga dengan PHP merupakan sebuah bahasa pemrograman yang didesain untuk pengembangan *web*. PHP juga dapat digunakan untuk bahasa pemrograman umum selain *web*. Awal mula dari kemunculan dan

perkembangan PHP yaitu pada tahun 1995 oleh Rasmus Lerdorf, Awal mulanya PHP disebut sebagai bahasa pemrograman *server side* karena PHP diproses pada komputer *server* yang dalam konteks nya hal ini berbeda jika dibandingkan dengan bahasa pemrograman *client-side* seperti *JavaScript* yang prosesnya dilakukan pada *web browser client*. (Suhartini dkk., 2020)

Bahasa pemrograman pada PHP bekerja dalam sebuah *web server*, dimana *script* dari PHP haruslah tersimpan kedalam sebuah *server* dan melakukan eksekusi atau proses pun dalam *server* tersebut. Dengan menggunakan pemrograman PHP, sebuah *website* akan menjadi lebih interaktif serta dinamis. (Nirsal dkk., 2020).

PHP memiliki fungsi utama dalam membangun sebuah web untuk melakukan pengelolaan data dalam *database*. PHP juga merupakan bahasa pemrograman yang dapat digunakan untuk menterjemahkan baris-baris kode pemrograman agar menjadi kode mesin yang dapat di mengerti oleh komputer yang sifatnya *server-side* dan dapat di tambahkan ke dalam HTML. (Tumini & Fitria, 2021)

#### 2.3.3.7. JSON

*Javascript Object Notation* atau dapat disebut juga dengan JSON adalah format dalam melakukan sebuah pertukaran data yang digunakan baik pada *front-end* ataupun pada *back-end* dari *website* ataupun sebuah *service* (Wardhana dkk., 2020). Dalam pertukaran data pada *web service* , biasanya menggunakan XML dan juga JSON.(Nugroho & Primadewi, 2021)

Alasan penggunaan JSON yaitu dikarenakan pertukaran data dengan menggunakan format ini merupakan pertukaran data yang ringan serta mudah dibaca dan ditulis oleh manusia, serta mudah untuk di terjemahkan dan dibuat oleh komputer. Format penulisan JSON ini tidak bergantung pada salah satu bahasa pemrograman saja, sehingga JSON dapat digunakan untuk pertukaran data antar bahasa pemrograman.(Sahrial, Fauzi, & Susilawati, 2022)

JSON menggunakan struktur data universal yang meliputi kumpulan nilai atau objek serta daftar nilai yang terurut atau larik (*array*). Penulisan objek yang tidak terurutkan dimulai dengan simbol { dan diakhiri dengan simbol }, serta setiap nama dari objek data diikuti dengan menggunakan titik dua (:) serta koma (,) untuk memisahkan setiap pasangan nilai.(Sahrial, Fauzi, & Susilawati, 2022)

JSON memiliki keunggulan yang khas dibandingkan dengan XML, yaitu salah satu diantaranya ukuran karakter yang dibutuhkan JSON lebih kecil, proses penggunaan *browser* parsing JSON yang lebih sederhana serta pertukaran data yang lebih unggul apabila dibandingkan dengan XML. (Nugroho & Primadewi, 2021b)

#### 2.4 Teori Pengujian *Blackbox*

Definisi *Software testing* merupakan sebuah proses analisa suatu entitas perangkat lunak dalam melakukan deteksi perbedaan antara kondisi yang sudah ada dengan kondisi yang diminta (*Defectsm/error/bos*) dan mengevaluasi fitur dari entitas perangkat lunak (*Software*). (Wicaksono, 2021)

Pengujian Box Hitam (*Black Box Testing*) merupakan metode pengujian alternatif yang dimana metode pengujian ini berfokus dalam spesifikasi yang dibutuhkan seperti fungsionalitas saja, dalam tujuan untuk mengetahui dan juga memastikan semua fungsi seperti *input* dan *output* dari aplikasi sudah sesuai dan berjalan berdasarkan pada spesifikasi awal yang diminta. Salah satu alasan penggunaan *Blackbox Testing* dikarenakan mudah untuk diimplementasikan karena memfokuskan pada spesifikasi dari sisi fungsional sebuah aplikasi dan tidak mementingkan bagaimana aplikasi tersebut dibuat. (Kusuma Dewi, Pratama, Putera, & Carudin, 2022)

Sedangkan menurut buku "*Blackbox Testing Teori dan Studi Kasus*", *blackbox* merupakan metode pengujian yang mendasar dan mengasumsikan bahwa adanya ketidaktahuan tentang struktur maupun detail dari pelaksanaan secara internal berdasarkan pada objek yang akan dinilai, seperti struktur internal kode atau program. (Wicaksono, 2021)

## 2.5 Tinjauan Studi

### A. Tinjauan Studi 1

**Tabel 2. 5** Tinjauan Studi 1

No	Data Jurnal / Makalah	Keterangan
1	Judul	Implementasi Teknologi RESTFUL <i>Web Service</i> Dalam Pengembangan Sistem Informasi Perekaman Prestasi Mahasiswa Berbasis <i>Website</i> (Studi Kasus: Fakultas Teknologi Pertanian Universitas Brawijaya)
2	Jurnal	Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN: 2548-964X
3	Volume, Nomor dan halaman	Volume 4 Nomor 2, 680-689
4	Bulan & Tahun	Februari 2020
5	Penulis	Wiku Galindra Wardhana, Issa Arwani, Bayu Rahayudi
6	Penerbit	Universitas Brawijaya

7	Tujuan Penelitian	Penelitian ini bertujuan untuk membuat sebuah sistem informasi yang akan mencatat pencapaian mahasiswa di Fakultas Teknologi Pertanian Universitas Brawijaya dengan menggunakan teknologi layanan web RESTFUL. Sistem ini akan membantu dalam mengumpulkan data prestasi mahasiswa, memberikan penghargaan atas prestasi yang mereka raih, serta memberikan manfaat bagi semua pihak yang terlibat.
8	Lokasi & Subjek Penelitian	Lokasi penelitian ini adalah Fakultas Teknologi Pertanian Universitas Brawijaya. Subjek penelitian meliputi mahasiswa dan pihak terkait di Fakultas Teknologi Pertanian, dengan ketua Program Studi Teknologi Informasi sebagai salah satu subjek yang terlibat dalam pengumpulan data melalui wawancara.
9	Metode yang digunakan	<i>Web Service restful api</i>
10	Perancangan Sistem	Dalam penelitian ini, sistem dirancang berdasarkan kebutuhan yang sudah direncanakan. Proses perancangan melibatkan pembuatan diagram urutan, diagram kelas, dan model data fisik. Selain itu, juga dirancang REST API untuk mendukung implementasi sistem perekaman prestasi mahasiswa.
11	Hasil Penelitian	Hasil dari penelitian ini adalah pengembangan sistem informasi untuk mencatat prestasi mahasiswa di Fakultas Teknologi Pertanian Universitas Brawijaya menggunakan teknologi layanan web RESTful. Sistem ini telah diimplementasikan dan pengujian menunjukkan tingkat keberhasilan yang tinggi, serta usability yang baik.
12	Kekuatan Penelitian	<ol style="list-style-type: none"> <li>1. Penggunaan teknologi RESTFUL <i>Web Service</i> yang memungkinkan pengembangan sistem informasi yang efisien.</li> <li>2. Penerapan <i>website</i> menggunakan <i>framework</i> Laravel dan REST API dengan <i>framework</i> Lumen menunjukkan penggunaan teknologi yang modern dan relevan.</li> <li>3. Pengujian sistem dilakukan dalam 3 tahap, termasuk pengujian terhadap REST API dan <i>usability</i>, menunjukkan pendekatan komprehensif dalam memastikan kualitas sistem.</li> </ol>
13	Kelemahan Penelitian	<ol style="list-style-type: none"> <li>1. Terbatasnya jumlah responden pada pengujian <i>usability</i> (8 responden) mungkin</li> </ol>

		<p>mempengaruhi representativitas hasil pengujian.</p> <p>2. Tidak disebutkan secara eksplisit mengenai evaluasi terhadap kebutuhan fungsional dan non-fungsional sistem setelah implementasi, yang dapat menjadi pertimbangan penting dalam pengembangan sistem informasi.</p> <p>3. Tidak disebutkan secara detail mengenai metode evaluasi yang digunakan dalam menilai keberhasilan implementasi sistem, sehingga informasi mengenai validitas dan reliabilitas hasil penelitian menjadi kurang jelas.</p>
14	Kesimpulan	<p>Kesimpulan dari penelitian ini adalah pengembangan sistem perekaman prestasi mahasiswa di Fakultas Teknologi Pertanian Universitas Brawijaya menggunakan teknologi RESTFUL <i>Web Service</i> berjalan dengan baik. Implementasi dilakukan dengan framework Laravel untuk <i>website</i> dan Lumen untuk REST API. Pengujian menunjukkan tingkat keberhasilan yang tinggi dan sistem mendapat penilaian baik dalam pengujian <i>usability</i>. Penggunaan metode <i>prototyping</i> dan teknologi terkini juga memberikan kontribusi penting dalam penelitian ini.</p>

#### B. Tinjauan Studi 2

**Tabel 2. 6** Tinjauan Studi 2

No	Data Jurnal / Makalah	Keterangan
1	Judul	Implementasi <i>Web Service</i> Sebagai Penyedia Informasi untuk Aplikasi Pengelolaan Jadwal Pemberian Pakan Ikan (Studi Kasus : Pokdakan Karya Bersama)
2	Jurnal	Jurnal Informatika dan Rekayasa Perangkat Lunak (JATIKA) p-ISSN: 2797-3492 e-ISSN: 2797-2011
3	Volume, Nomor & Halaman	Volume 4 Nomor 3, 335-341
4	Bulan & Tahun	September 2023
5	Penulis	Fredy Hermawan, A Ferico Octaviansyah Pasaribu
6	Penerbit	Universitas Teknokrat Indonesia
7	Tujuan Penelitian	Penelitian ini bertujuan membuat <i>Web Service</i> yang memberikan informasi untuk mengatur jadwal pemberian makan ikan, dan dapat terhubung dengan sistem lain.

8	Lokasi & Subjek Penelitian	Lokasi penelitian ini adalah Desa Jatimulyo, Kecamatan Jatiagung, Kabupaten Lampung Selatan. Subjek penelitian adalah kelompok pembudidaya ikan yang tergabung dalam Pokdakan Karya Bersama.
9	Metode yang digunakan	<i>Web Service &amp; UML</i>
10	Perancangan Sistem	Dalam penelitian ini, sistem direncanakan dengan pembuatan diagram <i>use case</i> , diagram aktivitas, diagram kelas, dan kartu CRC. Desain <i>wireframe</i> juga digunakan sebagai representasi <i>visual</i> sistem untuk pengujian integrasi.
11	Hasil Penelitian	Hasil dari penelitian ini adalah pembuatan sistem layanan <i>web</i> yang memberikan informasi untuk aplikasi <i>mobile</i> pembudidaya ikan. Sistem ini fokus pada pengelolaan jadwal pemberian pakan ikan dengan enam kategori layanan: otentikasi, otorisasi, komunitas, anggota, kelompok, dan jadwal.
12	Kekuatan Penelitian	Kekuatan penelitian ini adalah penggunaan metode <i>Extreme Programming (XP)</i> yang meliputi tahap <i>planning, design, coding, dan testing</i> , serta pemanfaatan <i>framework Spring Boot</i> dengan bahasa pemrograman Java untuk menjaga kualitas sistem.
13	Kelemahan Penelitian	Kelemahan penelitian ini mungkin terletak pada keterbatasan ruang lingkup studi kasus hanya pada kelompok pembudidaya ikan tertentu, sehingga generalisasi hasil mungkin terbatas pada konteks tersebut.
14	Kesimpulan	Dari penelitian, aplikasi dibangun dengan metode UML dan <i>framework Spring Boot</i> untuk membuat <i>Web Service</i> yang mengatur jadwal pemberian makan ikan. <i>Web Service</i> ini bisa terhubung dengan sistem lain di platform yang berbeda. Sukses 100% dalam pengujian <i>blackbox</i> menunjukkan sistem dibuat sesuai spesifikasi dengan baik.

### C. Tinjauan Studi 3

**Tabel 2. 7** Tinjauan Studi 3

No	Data Jurnal / Makalah	Keterangan
1	Judul	Penerapan <i>Web Service</i> untuk Integrasi Data Simperpus dan SIAK
2	Jurnal	Jurnal Komtika (Komputasi dan Informatika) e-ISSN: 2580-2852
3	Volume, Nomor & Halaman	Volume 4 Nomor 2, 71-81

4	Bulan & Tahun	November 2020
5	Penulis	Setiya Nugroho, Ardhin Primadewi
6	Penerbit	Universitas Muhammadiyah Magelang
7	Tujuan Penelitian	Penelitian ini bertujuan mengintegrasikan basis data antara Sistem Informasi Perpustakaan (Simperpus) dan Sistem Informasi Akademik (SIAK) menggunakan <i>Web Service</i> . Integrasi ini dilakukan untuk mengatasi masalah duplikasi dan ketidakkonsistenan data yang disebabkan oleh pemrosesan data yang tidak terintegrasi di kedua sistem tersebut. Dengan menggunakan <i>Rest Client</i> , <i>Rest Server</i> , dan pertukaran data dalam format JSON, penelitian ini bertujuan menciptakan sistem informasi yang terintegrasi dan efisien antara Simperpus dan SIAK.
8	Lokasi & Subjek Penelitian	Penelitian ini dilakukan di Universitas Muhammadiyah Magelang dengan fokus pada integrasi Sistem Informasi Perpustakaan (Simperpus) dan Sistem Informasi Akademik (SIAK) universitas. Penelitian ini bertujuan mengembangkan sistem informasi terintegrasi antara Simperpus dan SIAK menggunakan <i>Web Service</i> untuk menyelesaikan masalah duplikasi dan ketidakkonsistensian data di kedua sistem tersebut.
9	Metode yang digunakan	<i>Web Service</i> , XML-RPC, SOAP, REST
10	Perancangan Sistem	Dalam penelitian ini, rancangan sistem menggunakan <i>Unified Modeling Language</i> (UML). Beberapa diagram yang digunakan termasuk <i>use case</i> , <i>class</i> , <i>sequence</i> , <i>activity</i> , <i>component</i> , dan <i>deployment diagram</i> untuk menggambarkan interaksi, struktur, dan implementasi sistem informasi terintegrasi antara Simperpus dan SIAK.
11	Hasil Penelitian	Hasil penelitian ini berhasil mengimplementasikan integrasi data antara Sistem Informasi Perpustakaan (Simperpus) dan Sistem Informasi Akademik (SIAK) menggunakan REST <i>Web Service</i> . Waktu respon pertukaran data antara kedua sistem hanya 137ms saat diuji. Format pertukaran data yang digunakan adalah JSON. Dengan <i>Rest Server</i> di SIAK dan <i>Rest Client</i> di Simperpus, sistem ini memungkinkan akses data mahasiswa dari SIAK melalui API yang disediakan oleh <i>Rest Server</i> . Integrasi data antara Simperpus dan SIAK dapat dilakukan dengan efisien.

		Sistem ini kini digunakan di perpustakaan dan biro akademik Universitas Muhammadiyah Magelang. Untuk menjaga kinerja sistem, pemeliharaan akan dilakukan secara berkala dengan pemeriksaan data aplikasi untuk memastikan sistem tetap berjalan dengan baik.
12	Kekuatan Penelitian	<ol style="list-style-type: none"> <li>1. Penelitian berhasil mengintegrasikan data antara Sistem Informasi Perpustakaan (Simperpus) dan Sistem Informasi Akademik (SIAK) menggunakan REST <i>Web Service</i>, mengatasi masalah duplikasi dan ketidakkonsistensian data.</li> <li>2. Pertukaran data efisien dengan format JSON, mengimplementasikan <i>Rest Server</i> di SIAK dan <i>Rest Client</i> di Simperpus, dengan <i>respon time</i> cepat 137ms.</li> <li>3. Metode pengembangan sistem <i>Waterfall</i> digunakan, memungkinkan perancangan dan implementasi sistem yang terstruktur.</li> </ol>
13	Kelemahan Penelitian	<ol style="list-style-type: none"> <li>1. Terbatas pada Lingkungan Tertentu Penelitian ini dilakukan di lingkungan Universitas Muhammadiyah Magelang, sehingga hasil dan implementasinya mungkin tidak langsung dapat diterapkan pada lingkungan atau institusi lain yang memiliki karakteristik yang berbeda.</li> <li>2. Ketergantungan pada Teknologi Tertentu Penggunaan teknologi dan <i>framework</i> tertentu seperti Codeigniter, MySQL, dan Postman dalam penelitian ini dapat menjadi ketergantungan yang membatasi fleksibilitas sistem dalam pengembangan lebih lanjut.</li> <li>3. Keterbatasan Pembahasan Beberapa aspek seperti keamanan data, skalabilitas sistem, dan integrasi dengan sistem lain mungkin tidak dibahas secara mendalam dalam penelitian ini, sehingga dapat menjadi area pengembangan lebih lanjut untuk penelitian selanjutnya.</li> </ol>
14	Kesimpulan	Kesimpulan dari penelitian ini adalah integrasi data menggunakan Rest API berhasil dengan waktu respon pertukaran data 137ms. Format data yang digunakan adalah JSON. <i>Rest Server</i> digunakan di SIAK dan <i>Rest Client</i> di Simperpus. Sistem lain bisa mengakses data mahasiswa dari SIAK melalui <i>Rest Server</i> .

D. Tinjauan Studi 4

**Tabel 2. 8** Tinjauan Studi 4

No	Data Jurnal / Makalah	Keterangan
1	Judul	Analisis perbandingan metode SOAP dan REST yang digunakan pada <i>framework</i> FLASK untuk membangun <i>Web Service</i>
2	Jurnal	Jurnal Teknologi Informasi dan Komunikasi ISSN: 1978-0087
3	Volume, Nomor & Halaman	Volume XIV Nomor 2, 1-7
4	Bulan & Tahun	Juni 2019
5	Penulis	M Gilvy Langgawan Putra, M Ihsan Alfani Putera
6	Penerbit	Institut Teknologi Kalimantan
7	Tujuan Penelitian	Tujuan dari penelitian ini adalah untuk melakukan analisis perbandingan antara metode SOAP dan REST yang digunakan pada <i>framework</i> Flask dalam membangun <i>web service</i> . Penelitian ini bertujuan untuk mengevaluasi performa kedua metode tersebut dalam konteks pengiriman request dan respon pada <i>web service</i> berbasis Flask.
8	Lokasi & Subjek Penelitian	Penelitian ini dilakukan di Institut Teknologi Kalimantan. Subjek penelitian ini mencakup analisis perbandingan metode SOAP dan REST yang digunakan pada <i>framework</i> Flask untuk membangun <i>web service</i> .
9	Metode yang digunakan	Dalam penelitian tersebut, metode yang digunakan adalah metode SOAP dan REST yang diimplementasikan pada <i>framework</i> Flask. Penelitian ini melibatkan pengujian dan evaluasi performa kedua metode tersebut dalam konteks pengiriman request dan respon pada <i>web service</i> berbasis Flask.
10	Perancangan Sistem	<ol style="list-style-type: none"> <li>1. Pengujian analisis perbandingan request dan respon antara SOAP dan REST pada <i>framework</i> Flask.</li> <li>2. Pengujian <i>web socket</i> untuk membandingkan interaksi antara klien dan <i>web server</i> antara REST dan SOAP.</li> <li>3. Melakukan pengujian request dan respon yang terdiri dari GET, PUT, dan POST untuk REST, serta POST dan GET untuk SOAP.</li> <li>4. Menganalisis performa kedua metode berdasarkan data yang diperoleh dari pengujian.</li> </ol>
11	Hasil Penelitian	Hasil dari penelitian ini adalah:

		<ol style="list-style-type: none"> <li>1. REST menunjukkan performa yang lebih baik daripada SOAP dalam pengiriman request dan respon pada web service berbasis Flask.</li> <li>2. Penggunaan socket XML disarankan daripada JSON untuk mendapatkan metode yang tidak diketahui oleh klien pada web browser tanpa perlu lemparan halaman lain.</li> <li>3. Penelitian ini memberikan pemahaman yang lebih mendalam tentang perbandingan antara SOAP dan REST dalam konteks pengembangan web service.</li> <li>4. Hasil penelitian ini dapat menjadi panduan bagi pengembang web dalam memilih metode yang sesuai dengan kebutuhan dan tujuan pengembangan web service.</li> </ol>
12	Kekuatan Penelitian	<p>Kekuatan penelitian ini adalah :</p> <ol style="list-style-type: none"> <li>1. Melakukan perbandingan antara metode SOAP dan REST dalam konteks pengembangan web service, memberikan wawasan yang berharga bagi pengembang web.</li> <li>2. Menggunakan framework Flask sebagai basis penelitian, memberikan pemahaman yang lebih mendalam tentang implementasi kedua metode pada lingkungan pengembangan yang spesifik.</li> <li>3. Melakukan pengujian request dan respons serta web socket untuk mengevaluasi performa kedua metode secara komprehensif.</li> </ol>
13	Kelemahan Penelitian	<p>Kelemahan penelitian ini adalah:</p> <ol style="list-style-type: none"> <li>1. Melakukan perbandingan antara metode SOAP dan REST dalam konteks pengembangan web service, memberikan wawasan yang berharga bagi pengembang web.</li> <li>2. Menggunakan framework Flask sebagai basis penelitian, memberikan pemahaman yang lebih mendalam tentang implementasi kedua metode pada lingkungan pengembangan yang spesifik.</li> <li>3. Melakukan pengujian request dan respons serta web socket untuk mengevaluasi performa kedua metode secara komprehensif.</li> </ol>
14	Kesimpulan	<p>Dari penelitian yang dilakukan, dapat disimpulkan bahwa REST memiliki performa yang lebih baik daripada SOAP dalam pengiriman request dan respon. Penggunaan</p>

		socket XML direkomendasikan daripada JSON untuk mendapatkan beberapa metode yang tidak diketahui oleh klien pada web browser tanpa perlu lemparan halaman lain. Penelitian ini memberikan pemahaman yang lebih mendalam tentang perbandingan antara SOAP dan REST dalam konteks pengembangan web service, serta memberikan wawasan berharga bagi pengembang web dalam memilih metode yang sesuai dengan kebutuhan dan tujuan pengembangan web service.
--	--	--

E. Tinjauan Studi 5

**Tabel 2. 9** Tinjauan Studi 5

No	Data Jurnal / Makalah	Keterangan
1	Judul	Implementasi <i>Web Service</i> menggunakan <i>RESTFUL API</i> pada aplikasi pemesanan sarung goyor suhutex
2	Jurnal	Jurnal Riset dan Aplikasi Mahasiswa Informatika (JRAMI) e-ISSN: 2715-8756
3	Volume, Nomor & Halaman	Volume 05 Nomor 01, 191-199
4	Bulan & Tahun	2024
5	Penulis	Muhammad Fakhri Ikhwanuzaki, Irma Handayani
6	Penerbit	Universitas Teknologi Yogyakarta
7	Tujuan Penelitian	Tujuan dari penelitian ini adalah untuk mengimplementasikan web service menggunakan RESTful API pada aplikasi pemesanan sarung goyor Suhutex guna memudahkan proses pemasaran produk dan pertukaran data antara pelanggan dan produsen .
8	Lokasi & Subjek Penelitian	Penelitian ini dilakukan di Universitas Teknologi Yogyakarta. Subjek penelitian ini adalah implementasi web service menggunakan RESTful API pada aplikasi pemesanan sarung goyor Suhutex .
9	Metode yang digunakan	Penelitian ini menggunakan metode Research and Development (R&D) yang merupakan pendekatan untuk pengembangan lebih lanjut hasil penelitian atau produk penelitian.
10	Perancangan Sistem	<ol style="list-style-type: none"> <li>1. Observasi langsung pada Perusahaan Suhutex untuk memahami proses pembuatan dan pemasaran sarung goyor.</li> <li>2. Wawancara dengan pemilik usaha dan pelanggan untuk mendapatkan data</li> </ol>

		<p>mengenai kekurangan dan kelebihan mekanisme pemasaran.</p> <ol style="list-style-type: none"> <li>3. Studi pustaka untuk mencari informasi relevan.</li> <li>4. Pembuatan Data Flow Diagram (DFD) untuk memahami aliran informasi dalam sistem.</li> <li>5. Pembuatan Entity Relationship Diagram (ERD) sebagai dasar dalam pembuatan database.</li> <li>6. Implementasi teknologi RESTful API untuk integrasi web dan mobile dalam aplikasi pemesanan sarung goyor</li> </ol>
11	Hasil Penelitian	<p>Hasil dari penelitian ini adalah:</p> <ol style="list-style-type: none"> <li>1. Pengembangan aplikasi pemesanan sarung goyor berbasis web mobile yang terintegrasi dengan teknologi web service menggunakan konsep REST API.</li> <li>2. Aplikasi ini membantu memudahkan proses pemasaran Perusahaan Suhutex dan memungkinkan pelanggan untuk dengan mudah melakukan pemesanan produk.</li> <li>3. Pemanfaatan teknologi REST API dalam aplikasi E-Katalog Sarung Goyor memfasilitasi pertukaran data antara pelanggan dan produsen, memastikan akses informasi produk yang akurat dan mudah</li> </ol>
12	Kekuatan Penelitian	<p>Kekuatan penelitian ini adalah :</p> <ol style="list-style-type: none"> <li>1. Penggunaan metode Research and Development (R&amp;D) yang memungkinkan pengembangan aplikasi pemesanan sarung goyor dengan baik .</li> <li>2. Integrasi teknologi RESTful API dalam aplikasi untuk memudahkan pertukaran data antara pelanggan dan produsen .</li> <li>3. Fokus pada pengembangan sistem yang dapat meningkatkan efisiensi dalam mekanisme pemasaran Perusahaan Suhutex .</li> <li>4. Kolaborasi antara peneliti dan pemangku kepentingan (pemilik usaha dan pelanggan) untuk memastikan kebutuhan yang terpenuhi dalam perancangan aplikasi.</li> </ol>
13	Kelemahan Penelitian	<p>Kelemahan penelitian ini adalah:</p> <ol style="list-style-type: none"> <li>1. Terbatasnya ruang lingkup penelitian hanya pada implementasi web service menggunakan RESTful API pada aplikasi pemesanan sarung goyor Suhutex tanpa memperluas pada aspek lain dari bisnis.</li> </ol>

		<ol style="list-style-type: none"> <li>2. Kurangnya informasi mengenai evaluasi kepuasan pengguna terhadap aplikasi yang dikembangkan.</li> <li>3. Tidak adanya analisis mendalam mengenai dampak sosial, ekonomi, atau lingkungan dari implementasi teknologi ini pada bisnis sarung goyor.</li> <li>4. Keterbatasan sumber daya dan waktu dalam melakukan penelitian yang mungkin mempengaruhi kedalaman analisis dan generalisasi hasil penelitian.</li> </ol>
14	Kesimpulan	<p>Dari penelitian ini, dapat disimpulkan bahwa: Implementasi web service menggunakan RESTful API pada aplikasi pemesanan sarung goyor Suhutex merupakan langkah inovatif dalam memperbaiki mekanisme pemasaran tradisional menjadi lebih efisien dan mudah diakses oleh pelanggan. Aplikasi ini memungkinkan pertukaran data yang akurat antara produsen dan konsumen melalui platform mobile, memperbaiki proses pemesanan produk. Dengan memanfaatkan teknologi RESTful API, aplikasi ini memberikan solusi bagi permasalahan konvensional dalam pemasaran produk sarung goyor, meningkatkan keterjangkauan informasi produk bagi pelanggan, serta mempercepat proses transaksi secara online.</p>

F. Tinjauan Studi 6

**Tabel 2. 10** Tinjauan Studi 6

No	Data Jurnal / Makalah	Keterangan
1	Judul	Implementasi <i>Web Service</i> dengan metode <i>REST API</i> untuk integrasi data covid 19 di Sulawesi Selatan
2	Jurnal	Jurnal Sintaks Logika (JSILOG) e-ISSN: 2775-412X
3	Volume, Nomor & Halaman	Volume 02 Nomor 01, 236-241
4	Bulan & Tahun	Januari 2022
5	Penulis	Baharudin, Hamka Wakkang, Bambang Irianto
6	Penerbit	Universitas Muhammadiyah Parepare
7	Tujuan Penelitian	Tujuan penulisan jurnal ini adalah untuk menjelaskan implementasi web service dengan metode REST API untuk integrasi data perkembangan COVID-19 di Sulawesi Selatan. Penelitian ini bertujuan untuk memperlihatkan bagaimana web service dapat digunakan untuk menghubungkan data

		COVID-19 dari berbagai sumber secara efisien melalui teknologi REST API.
8	Lokasi & Subjek Penelitian	Penelitian dalam jurnal ini dilakukan di Dinas Kesehatan yang berada di kota Parepare, Sulawesi Selatan. Subjek penelitian meliputi implementasi web service dengan metode REST API untuk integrasi data perkembangan COVID-19 di wilayah Sulawesi Selatan.
9	Metode yang digunakan	Metode yang digunakan dalam jurnal ini meliputi pembuatan API web service dengan metode RESTful untuk menarik data dari database. Data yang telah ditarik kemudian dapat diakses melalui website dan aplikasi mobile secara terpusat. Selain itu, penelitian ini juga melibatkan penggunaan teknologi REST API untuk pertukaran data dalam bentuk JSON melalui URI..
10	Perancangan Sistem	<ol style="list-style-type: none"> <li>1. Pembuatan API web service dengan metode RESTful.</li> <li>2. Penarikan data dari database menggunakan API.</li> <li>3. Akses data secara terpusat melalui website dan aplikasi mobile.</li> <li>4. Pertukaran data dalam bentuk JSON melalui URI.</li> <li>5. Penggabungan data dari berbagai database menggunakan REST API. Pembuatan Entity Relationship Diagram (ERD) sebagai dasar dalam pembuatan database.</li> </ol>
11	Hasil Penelitian	<p>Hasil dari penelitian ini adalah:</p> <ol style="list-style-type: none"> <li>1. Grafik, tabel, dan gambar digunakan untuk memvisualisasikan data penelitian.</li> <li>2. Pembahasan logis mengaitkan hasil penelitian dengan sumber rujukan yang relevan.</li> <li>3. Perancangan sistem menggunakan model <i>UML (Unified Modelling Language)</i> seperti Activity diagram, class diagram, dan Sequence diagram.</li> <li>4. Implementasi web service dengan metode REST API berhasil mengintegrasikan data COVID-19 dari berbagai sumber di Sulawesi Selatan.</li> <li>5. Penggunaan teknologi RESTful memungkinkan pertukaran data secara efisien melalui web service.</li> </ol>
12	Kekuatan Penelitian	<p>Kekuatan penelitian ini adalah :</p> <ol style="list-style-type: none"> <li>1. Penggunaan metode REST API memungkinkan integrasi data dari berbagai</li> </ol>

		<p>sumber tanpa perlu melakukan perubahan pada sistem yang sudah ada.</p> <ol style="list-style-type: none"> <li>2. Implementasi web service dengan metode RESTful memungkinkan akses data secara terpusat melalui website dan aplikasi mobile.</li> <li>3. Penggunaan teknologi REST API dalam pertukaran data JSON melalui URI mempermudah proses pengambilan dan penggabungan data dari database yang berbeda.</li> <li>4. Hasil pengujian menunjukkan keberhasilan dalam mengimplementasikan web service dengan metode REST untuk integrasi data COVID-19 di Sulawesi Selatan.</li> <li>5. Penelitian ini memberikan kontribusi dalam pengembangan sistem informasi perkembangan COVID-19 dengan pendekatan yang efisien dan terpusat.</li> </ol>
13	Kelemahan Penelitian	<p>Kelemahan penelitian ini adalah:</p> <ol style="list-style-type: none"> <li>1. Tidak disebutkan secara rinci mengenai metodologi pengujian yang digunakan dalam menguji keberhasilan implementasi web service dengan metode REST.</li> <li>2. Tidak ada informasi mengenai skala atau ukuran sampel yang digunakan dalam penelitian, sehingga sulit untuk mengevaluasi generalisasi hasil penelitian.</li> <li>3. Tidak disebutkan secara detail mengenai kendala atau hambatan yang dihadapi selama proses implementasi web service dengan metode REST.</li> <li>4. Tidak ada analisis mendalam mengenai dampak atau manfaat dari integrasi data COVID-19 di Sulawesi Selatan menggunakan web service.</li> <li>5. Tidak disertakan informasi mengenai rencana pengembangan atau perbaikan sistem web service di masa depan berdasarkan temuan dari penelitian ini.</li> </ol>
14	Kesimpulan	<p>Dari jurnal ini, dapat disimpulkan bahwa implementasi web service dengan metode REST API untuk integrasi data COVID-19 di Sulawesi Selatan berhasil dilakukan. Teknologi web service memungkinkan pertukaran data antara sistem lama dan baru tanpa perlu melakukan perubahan pada keduanya. Penggunaan RESTful API dalam pertukaran data JSON melalui URI mempermudah pengambilan data dari berbagai database dan</p>

		menyatukan informasi tersebut secara terpusat. Meskipun terdapat kelemahan seperti kurangnya detail mengenai metodologi pengujian dan ukuran sampel, penelitian ini memberikan kontribusi dalam pengembangan sistem informasi COVID-19 yang efisien dan terintegrasi.
--	--	---

G. Tinjauan Studi 7

**Tabel 2. 11** Tinjauan Studi 7

No	Data Jurnal / Makalah	Keterangan
1	Judul	Implementasi <i>REST API Web Service</i> dalam membangun aplikasi <i>multiplatform</i> untuk usaha jasa
2	Jurnal	Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer (Matrik) e-ISSN: 2476-9843 p-ISSN: 1858-4144
3	Volume, Nomor & Halaman	Volume 18 Nomor 02, 284-293
4	Bulan & Tahun	Mei 2019
5	Penulis	Romi Choirudin, Ahmat Adil
6	Penerbit	STMIK Bumigora Mataram
7	Tujuan Penelitian	Tujuan penulisan jurnal ini adalah untuk menjelaskan implementasi web service dengan metode REST API untuk integrasi data perkembangan COVID-19 di Sulawesi Selatan. Penelitian ini bertujuan untuk memperlihatkan bagaimana web service dapat digunakan untuk menghubungkan data COVID-19 dari berbagai sumber secara efisien melalui teknologi REST API.
8	Lokasi & Subjek Penelitian	Penelitian dalam jurnal ini dilakukan di Dinas Kesehatan yang berada di kota Parepare, Sulawesi Selatan. Subjek penelitian meliputi implementasi web service dengan metode REST API untuk integrasi data perkembangan COVID-19 di wilayah Sulawesi Selatan.
9	Metode yang digunakan	Metode yang digunakan dalam jurnal ini meliputi pembuatan API web service dengan metode RESTful untuk menarik data dari database. Data yang telah ditarik kemudian dapat diakses melalui website dan aplikasi mobile secara terpusat. Selain itu, penelitian ini juga melibatkan penggunaan teknologi REST API untuk pertukaran data dalam bentuk JSON melalui URI..

10	Perancangan Sistem	<ol style="list-style-type: none"> <li>6. Pembuatan API web service dengan metode RESTful.</li> <li>7. Penarikan data dari database menggunakan API.</li> <li>8. Akses data secara terpusat melalui website dan aplikasi mobile.</li> <li>9. Pertukaran data dalam bentuk JSON melalui URI.</li> <li>10. Penggabungan data dari berbagai database menggunakan REST API. Pembuatan Entity Relationship Diagram (ERD) sebagai dasar dalam pembuatan database.</li> </ol>
11	Hasil Penelitian	<p>Hasil dari penelitian ini adalah:</p> <ol style="list-style-type: none"> <li>6. Grafik, tabel, dan gambar digunakan untuk memvisualisasikan data penelitian.</li> <li>7. Pembahasan logis mengaitkan hasil penelitian dengan sumber rujukan yang relevan.</li> <li>8. Perancangan sistem menggunakan model <i>UML (Unified Modelling Language)</i> seperti Activity diagram, class diagram, dan Sequence diagram.</li> <li>9. Implementasi web service dengan metode REST API berhasil mengintegrasikan data COVID-19 dari berbagai sumber di Sulawesi Selatan.</li> <li>10. Penggunaan teknologi RESTful memungkinkan pertukaran data secara efisien melalui web service.</li> </ol>
12	Kekuatan Penelitian	<p>Kekuatan penelitian ini adalah :</p> <ol style="list-style-type: none"> <li>6. Penggunaan metode REST API memungkinkan integrasi data dari berbagai sumber tanpa perlu melakukan perubahan pada sistem yang sudah ada.</li> <li>7. Implementasi web service dengan metode RESTful memungkinkan akses data secara terpusat melalui website dan aplikasi mobile.</li> <li>8. Penggunaan teknologi REST API dalam pertukaran data JSON melalui URI mempermudah proses pengambilan dan penggabungan data dari database yang berbeda.</li> <li>9. Hasil pengujian menunjukkan keberhasilan dalam mengimplementasikan web service dengan metode REST untuk integrasi data COVID-19 di Sulawesi Selatan.</li> <li>10. Penelitian ini memberikan kontribusi dalam pengembangan sistem informasi</li> </ol>

		perkembangan COVID-19 dengan pendekatan yang efisien dan terpusat.
13	Kelemahan Penelitian	Kelemahan penelitian ini adalah: 6. Tidak disebutkan secara rinci mengenai metodologi pengujian yang digunakan dalam menguji keberhasilan implementasi web service dengan metode REST. 7. Tidak ada informasi mengenai skala atau ukuran sampel yang digunakan dalam penelitian, sehingga sulit untuk mengevaluasi generalisasi hasil penelitian. 8. Tidak disebutkan secara detail mengenai kendala atau hambatan yang dihadapi selama proses implementasi web service dengan metode REST. 9. Tidak ada analisis mendalam mengenai dampak atau manfaat dari integrasi data COVID-19 di Sulawesi Selatan menggunakan web service. 10. Tidak disertakan informasi mengenai rencana pengembangan atau perbaikan sistem web service di masa depan berdasarkan temuan dari penelitian ini.
14	Kesimpulan	Dari jurnal ini, dapat disimpulkan bahwa implementasi web service dengan metode REST API untuk integrasi data COVID-19 di Sulawesi Selatan berhasil dilakukan. Teknologi web service memungkinkan pertukaran data antara sistem lama dan baru tanpa perlu melakukan perubahan pada keduanya. Penggunaan RESTful API dalam pertukaran data JSON melalui URI mempermudah pengambilan data dari berbagai database dan menyatukan informasi tersebut secara terpusat. Meskipun terdapat kelemahan seperti kurangnya detail mengenai metodologi pengujian dan ukuran sampel, penelitian ini memberikan kontribusi dalam pengembangan sistem informasi COVID-19 yang efisien dan terintegrasi.

H. Tinjauan Studi 8

**Tabel 2. 12** Tinjauan Studi 8

No	Data Jurnal / Makalah	Keterangan
1	Judul	Rancangan Sistem Pemilihan dan Penetapan Harga dalam Proses Pengadaan Barang dan Jasa Logistik Berbasis Web
2	Jurnal	Jurnal IKRA-ITH Informatika e-ISSN: 2654-8054

		p-ISSN: 2580-4316
	Volume, Nomor & Halaman	Volume 5 Nomor 02, 108-117
4	Bulan & Tahun	Juli 2021
5	Penulis	Vivi Afifah, Dwipo Setyantoro
6	Penerbit	Institut Teknologi dan Bisnis Bank Rakyat Indonesia
7	Tujuan Penelitian	Tujuan dari penelitian dalam jurnal ini adalah membuat sistem online untuk memilih vendor dan menetapkan harga dalam pengadaan barang dan jasa logistik. Sistem ini diharapkan dapat mengelola informasi vendor dengan baik dan memberikan data yang tepat untuk membantu pengambilan keputusan dalam pengadaan barang dan jasa logistik.
8	Lokasi & Subjek Penelitian	Penelitian dalam jurnal ini dilakukan di BRI Institute dengan alamat di RM. Harsono No. 2 Ragunan, Jakarta Selatan 12550. Subjek penelitian dalam jurnal ini adalah pengembangan sistem informasi online untuk memilih vendor dan menetapkan harga dalam proses pengadaan barang dan jasa logistik.
9	Metode yang digunakan	Dalam penelitian jurnal ini, digunakan metode <i>Software Development Life Cycle</i> (SDLC) untuk membuat sistem online dalam memilih vendor dan menetapkan harga dalam pengadaan barang dan jasa logistik. Metode ini melibatkan analisis kebutuhan perangkat lunak, perancangan sistem, pembuatan, pengujian, dan perawatan sistem.
10	Perancangan Sistem	<ol style="list-style-type: none"> <li>1. Sistem online untuk memilih vendor dan menetapkan harga dalam pengadaan barang dan jasa logistik.</li> <li>2. Meliputi proses registrasi vendor, pengolahan data administrasi vendor, seleksi administrasi vendor, penetapan harga, dan penentuan vendor .</li> <li>3. Pengembangan sistem informasi menggunakan metode <i>Software Development Life Cycle</i> (SDLC) yang mencakup analisis kebutuhan perangkat lunak, perancangan sistem, implementasi, pengujian, dan pemeliharaan sistem</li> </ol>
11	Hasil Penelitian	<p>Hasil dari penelitian ini adalah:</p> <ol style="list-style-type: none"> <li>1. Sistem online untuk memilih vendor dan menetapkan harga dalam pengadaan barang dan jasa logistik telah berhasil dibangun.</li> <li>2. Sistem ini mencakup proses pendaftaran vendor, pengelolaan data vendor, seleksi</li> </ol>

		<p>vendor, penentuan harga, dan pemilihan vendor.</p> <p>3. Penulis menyarankan pengembangan sistem ini agar dapat memberikan hasil penelitian yang lebih baik dan berguna di masa mendatang.</p>
12	Kekuatan Penelitian	<p>Kekuatan penelitian ini adalah :</p> <ol style="list-style-type: none"> <li>1. Penggunaan metode <i>Software Development Life Cycle</i> (SDLC) dalam pengembangan sistem seleksi pemilihan dan penetapan harga memberikan pendekatan yang terstruktur dan komprehensif .</li> <li>2. Sistem yang dibangun mampu mengatasi masalah penyimpanan data vendor secara fisik, seleksi administrasi vendor yang manual, dan transaksi penetapan harga yang manual yang dapat menyebabkan tindak kecurangan .</li> <li>3. Penelitian ini memberikan kontribusi dalam pengembangan sistem informasi yang dapat meningkatkan efisiensi dan akurasi dalam proses pengadaan barang dan jasa logistik</li> </ol>
13	Kelemahan Penelitian	<p>Kelemahan penelitian ini adalah:</p> <ol style="list-style-type: none"> <li>1. Data vendor masih disimpan secara fisik yang rentan rusak dan hilang. Penting untuk menjaga keamanan dan membuat cadangan data.</li> <li>2. Proses seleksi vendor secara manual bisa menyebabkan kesalahan dan tidak sesuai dengan prosedur perusahaan.</li> <li>3. Penetapan harga secara manual bisa meningkatkan risiko penipuan. Oleh karena itu, diperlukan sistem otomatisasi yang lebih baik dan dapat termonitor.</li> </ol>
14	Kesimpulan	<p>Dari penelitian ini, dapat disimpulkan bahwa pembangunan sistem pemilihan dan penetapan harga untuk menentukan vendor dalam proses pengadaan barang dan jasa logistik berbasis web telah berhasil dilakukan. Sistem ini mencakup proses registrasi vendor, pengolahan data administrasi vendor, seleksi administrasi vendor, penetapan harga, dan penentuan vendor. Rekomendasi untuk pengembangan sistem ini termasuk perbaikan tampilan agar lebih menarik dan pengembangan lebih lanjut untuk hasil penelitian yang lebih baik di masa depan.</p>

I. Tinjauan Studi 9

Tabel 2. 13 Tinjauan Studi 9

No	Data Jurnal / Makalah	Keterangan
1	Judul	Rancang bangun sistem pengidentifikasi <i>Travel Bag</i> pada kelompok biro perjalanan umroh/haji berbasis <i>web</i>
2	Jurnal	Jurnal Teknologi Informasi dan Komunikasi STMIK Subang e-ISSN: 2252-4517 p-ISSN:
	Volume, Nomor & Halaman	Volume 15 Nomor 1,
4	Bulan & Tahun	April 2020
5	Penulis	Timbo Faritcan Parlaungan Siallagan, Dede Wisnu
6	Penerbit	STMIK Subang
7	Tujuan Penelitian	Merancang perangkat sistem yang mampu mencatat data tas dan pemiliknya, serta memprosesnya secara cepat dan tepat. Hal ini bertujuan untuk mengatasi masalah pengelolaan data kepemilikan tas yang masih menggunakan metode cetak manual, kesulitan dalam analisis data lampau, serta risiko data hilang dan rusak karena penggunaan media kertas. Dengan demikian, sistem ini diharapkan dapat meningkatkan efisiensi dan keamanan dalam identifikasi travel bag pada kelompok biro perjalanan umroh/haji.
8	Lokasi & Subjek Penelitian	Penelitian dalam artikel tersebut dilakukan di STMIK Subang. Subjek penelitian berfokus pada perancangan dan pembuatan sistem pengidentifikasi travel bag pada kelompok biro perjalanan umroh/haji berbasis web.
9	Metode yang digunakan	Metode Pemrograman CRUD ( <i>Create Read Update Delete</i> ) berbasis IOT ( <i>Internet Of Things</i> ) dengan menggunakan mikrokontroler Esp8266.
10	Perancangan Sistem	<ol style="list-style-type: none"> <li>1. Database MySQL.</li> <li>2. Pemrograman <i>CRUD (Create Read Update Delete)</i> menggunakan Bahasa pemrograman PHP.</li> <li>3. Pengujian dengan akses program browser untuk menampilkan laporan isi basis data berdasarkan parameter tertentu.</li> <li>4. Merancang antarmuka sistem yang terdiri dari header, menu, konten, dan footer.</li> </ol>

		5. Perancangan arsitektur sistem jaringan komputer lokal nirkabel untuk jalannya sistem.
11	Hasil Penelitian	<p>Hasil dari penelitian ini adalah:</p> <ol style="list-style-type: none"> <li>1. Pencatatan data tas dan pemiliknya dilakukan secara otomatis, mempermudah proses identifikasi tas pada peserta umrah.</li> <li>2. Sistem pengidentifikasi travel bag berbasis web menggunakan <i>mikrokontroler</i> Esp8266 dapat membantu biro perjalanan umroh/haji dalam pencatatan data dan identifikasi tas peserta secara digital.</li> <li>3. Resiko tas tertukar menjadi minim karena data tersimpan secara digital dan dapat diakses dengan mudah.</li> <li>4. Sistem ini memberikan kemudahan dalam proses identifikasi travel bag yang lebih cepat dan aman.</li> </ol>
12	Kekuatan Penelitian	<p>Kekuatan penelitian ini adalah :</p> <ol style="list-style-type: none"> <li>1. Penggunaan teknologi mikrokontroler Esp8266 yang memungkinkan pencatatan data secara otomatis.</li> <li>2. Sistem berbasis web yang memudahkan akses dan pengelolaan data secara digital.</li> <li>3. Penekanan pada keamanan data dengan penyimpanan digital yang aman.</li> </ol>
13	Kelemahan Penelitian	<p>Kelemahan penelitian ini adalah:</p> <ol style="list-style-type: none"> <li>1. Tidak disebutkan secara rinci mengenai skala implementasi dan uji coba sistem pada skala yang lebih luas.</li> <li>2. Tidak ada pembahasan mengenai ketersediaan sumber daya dan biaya implementasi sistem secara menyeluruh.</li> <li>3. Tidak ada analisis mendalam mengenai potensi hambatan atau tantangan yang mungkin dihadapi dalam implementasi sistem.</li> </ol>
14	Kesimpulan	<p>Kesimpulan dari artikel tersebut adalah bahwa rancang sistem pengidentifikasi travel bag pada kelompok biro perjalanan umroh/haji berbasis web menggunakan mikrokontroler Esp8266 dapat mempermudah proses identifikasi tas peserta umrah secara otomatis dan efisien, serta mengurangi risiko kesalahan identifikasi dan kehilangan data tas</p>

J. Tinjauan Studi 10

Tabel 2. 14 Tinjauan Studi 10

No	Data Jurnal / Makalah	Keterangan
1	Judul	Pemanfaatan JSON untuk menampilkan data <i>Realtime</i> covid-19 dengan model <i>view presenter</i>
2	Jurnal	Jurnal TEKNOINFO ISSN: 2615-224X
	Volume, Nomor & Halaman	Volume 16 Nomor 1, 144-149
4	Bulan & Tahun	2022
5	Penulis	Rysa Sahrial, Deri Fikri Fauzi, Eva Susilawati
6	Penerbit	Universitas Putra Indonesia
7	Tujuan Penelitian	Tujuan penelitian dari artikel tersebut adalah untuk memanfaatkan JSON dalam menampilkan data COVID-19 secara real-time dengan menggunakan <i>Model View Presenter (MVP)</i> . Penelitian ini bertujuan untuk mencari solusi dalam pencegahan COVID-19 dengan memanfaatkan teknologi informasi serta mengembangkan sistem informasi yang mampu menampilkan data penyebaran COVID-19 secara akurat dan <i>real-time</i> .
8	Lokasi & Subjek Penelitian	Penelitian dilakukan di Universitas Putra Indonesia, dengan alamat Jl. Dr. Muwardi 66, Cianjur. Subjek penelitian melibatkan pengembangan aplikasi informasi penyebaran COVID-19 dengan memanfaatkan JSON dan Model View Presenter (MVP) sebagai arsitektur pengembangan. Penelitian ini difokuskan pada pemanfaatan teknologi informasi untuk menyajikan data COVID-19 secara real-time kepada pengguna melalui aplikasi yang dikembangkan.
9	Metode yang digunakan	Metode <i>Hierarchy plus Input-Process-Output (HIPO)</i> . HIPO digunakan untuk mengembangkan sistem informasi yang mampu menampilkan data COVID-19 secara <i>real-time</i> dengan memanfaatkan JSON dan <i>Model View Presenter (MVP)</i>
10	Perancangan Sistem	<ol style="list-style-type: none"> <li>1. Menggunakan JSON &amp; <i>Model View Presenter (MVP)</i>.</li> <li>2. Arsitektur MVP terdiri dari tiga lapisan dasar: Model, View, dan Presenter.</li> <li>3. Model digunakan untuk menunjukkan objek dan data dalam aplikasi Android.</li> </ol>

		<ol style="list-style-type: none"> <li>4. Pengembangan sistem menggunakan metode <i>Hierarchy plus Input-Process-Output (HIPO)</i>.</li> <li>5. Sistem dirancang untuk menampilkan data COVID-19 secara <i>real-time</i> dengan akurat dan efisien</li> </ol>
11	Hasil Penelitian	<p>Hasil dari penelitian ini adalah:</p> <ol style="list-style-type: none"> <li>1. Penelitian berhasil mengimplementasikan penggunaan JSON untuk pertukaran data API rest dengan metode <i>Model View Presenter (MVP)</i>.</li> <li>2. Penggunaan tiga metode pemanggilan data JSON berbeda dalam sistem informasi penyebaran COVID-19.</li> <li>3. Pengujian menunjukkan perbandingan kinerja berdasarkan <i>memory</i> dan <i>CPU</i> dari tiga metode pemanggilan data JSON.</li> <li>4. Hasil pengujian menunjukkan efisiensi dan keefektifan penggunaan JSON dalam aplikasi informasi COVID-19</li> </ol>
12	Kekuatan Penelitian	<p>Kekuatan penelitian ini adalah :</p> <ol style="list-style-type: none"> <li>1. Penelitian menggunakan metode <i>Model View Presenter (MVP)</i> yang memisahkan antara tampilan, logika bisnis, dan data.</li> <li>2. Penggunaan JSON sebagai format pertukaran data ringan dan mudah diinterpretasikan oleh manusia dan komputer.</li> <li>3. Pengembangan sistem menggunakan metode <i>Hierarchy plus Input-Process-Output (HIPO)</i> untuk merancang aplikasi informasi COVID-19.</li> <li>4. Pengujian yang dilakukan memberikan pemahaman yang baik tentang kinerja aplikasi berdasarkan penggunaan <i>memory</i> dan <i>CPU</i>.</li> </ol>
13	Kelemahan Penelitian	<p>Kelemahan penelitian ini adalah:</p> <ol style="list-style-type: none"> <li>1. Penelitian hanya fokus pada implementasi penggunaan JSON dalam aplikasi informasi COVID-19 tanpa mempertimbangkan faktor-faktor lain yang mungkin memengaruhi kinerja sistem.</li> <li>2. Pengujian hanya dilakukan pada tiga metode pemanggilan data JSON tanpa mempertimbangkan variasi lain yang mungkin mempengaruhi performa aplikasi.</li> <li>3. Tidak ada pembahasan mendalam tentang potensi masalah keamanan data dalam pertukaran informasi COVID-19 menggunakan JSON</li> </ol>

14	Kesimpulan	Berdasarkan penelitian yang dilakukan, pemanfaatan JSON untuk pertukaran data API rest dengan metode <i>Model View Presenter (MVP)</i> terbukti sangat mudah diimplementasikan meskipun menggunakan tiga metode pemanggilan data JSON yang berbeda. Pengujian menunjukkan perbandingan kinerja berdasarkan penggunaan <i>memory</i> dan CPU dari tiga metode tersebut. Dengan demikian, penggunaan JSON dalam aplikasi informasi COVID-19 dapat dianggap efisien dan efektif
----	------------	--

K. Tinjauan Studi 11

Tabel 2. 15 Tinjauan Studi 11

No	Data Jurnal / Makalah	Keterangan
1	Judul	<i>Comparison of memory usage between REST API in Javascript and Golang</i>
2	Jurnal	MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer ISSN: 2476-9843 DOI: 10.30812/matrik.v22il.1325
	Volume, Nomor & Halaman	Volume 22 Nomor 1, 229-240
4	Bulan & Tahun	November 2022
5	Penulis	Hafizd Ardainsyah, Agung Fatwanto
6	Penerbit	Universitas Islam Negeri Sunan Kalijaga
7	Tujuan Penelitian	Untuk membandingkan penggunaan memori antara REST API yang diimplementasikan menggunakan Javascript dan Golang. Penelitian ini bertujuan untuk memberikan pemahaman yang lebih baik mengenai perbedaan dalam penggunaan sumber daya memori antara kedua teknologi tersebut.
8	Lokasi & Subjek Penelitian	Penelitian dilakukan di Universitas Islam Negeri Sunan Kalijaga, Yogyakarta, Indonesia. Subjek penelitian melibatkan perbandingan penggunaan memori antara REST API yang dibuat menggunakan Javascript dan Golang.
9	Metode yang digunakan	Metode yang digunakan adalah metodologi kuantitatif eksperimental, REST API Javascript dan Golang.
10	Perancangan Sistem	<ol style="list-style-type: none"> <li>1. Pembuatan dua REST API menggunakan Javascript dan Golang.</li> <li>2. Pengumpulan data menggunakan dataset dari Google Cloud.</li> <li>3. Penggunaan protokol GET dalam protokol HTTP.</li> </ol>

		<p>4. Pengukuran penggunaan memori oleh REST API Javascript dan Golang menggunakan browser dengan kemampuan inspeksi.</p> <p>5. Analisis perbedaan penggunaan memori antara kedua teknologi menggunakan uji statistik seperti uji Wilcoxon, uji T untuk data berpasangan, dan uji kesetaraan.</p>
11	Hasil Penelitian	<p>Hasil dari penelitian ini adalah:</p> <ol style="list-style-type: none"> <li>1. Setelah dilakukan uji Wilcoxon, uji T untuk data berpasangan, dan uji kesetaraan, terdapat perbedaan dalam penggunaan memori antara Javascript dan Golang.</li> <li>2. Meskipun terdapat perbedaan, perbedaan tersebut dianggap tidak signifikan secara praktis.</li> <li>3. Hasil analisis statistik menunjukkan bahwa secara statistik, Javascript dan Golang menggunakan sumber daya memori secara berbeda.</li> <li>4. Terdapat perbedaan signifikan dalam penggunaan memori antara Javascript dan Golang untuk data offline USA Person dan Wikipedia, namun tidak signifikan untuk data online USA Person.</li> <li>5. Untuk data online Wikipedia, tidak terdapat perbedaan signifikan dalam penggunaan memori antara Javascript dan Golang.</li> </ol>
12	Kekuatan Penelitian	<p>Kekuatan penelitian ini adalah :</p> <ol style="list-style-type: none"> <li>1. Penggunaan metodologi kuantitatif eksperimental yang memungkinkan analisis statistik yang kuat.</li> <li>2. Penggunaan uji kesetaraan, uji T untuk data berpasangan, dan uji Wilcoxon untuk menguji perbedaan penggunaan memori antara Javascript dan Golang.</li> <li>3. Penggunaan sampel data dari Google Cloud yang dapat diandalkan.</li> </ol>
13	Kelemahan Penelitian	<p>Kelemahan penelitian ini adalah:</p> <ol style="list-style-type: none"> <li>1. Penggunaan sampel data yang mungkin tidak mencakup semua skenario penggunaan yang mungkin terjadi.</li> <li>2. Perlunya penelitian lanjutan dengan sampel data yang lebih besar untuk mengonfirmasi temuan.</li> <li>3. Faktor-faktor lain di luar penggunaan memori yang mungkin memengaruhi performa REST API tidak dibahas secara mendalam.</li> </ol>

14	Kesimpulan	Kesimpulan dari artikel tersebut adalah bahwa meskipun terdapat perbedaan dalam penggunaan memori antara Javascript dan Golang, perbedaan tersebut dianggap tidak signifikan secara praktis. Diperlukan penelitian lanjutan dengan sampel data yang lebih besar untuk mengonfirmasi temuan tersebut.
----	------------	--

L. Tinjauan Studi 12

**Tabel 2. 16** Tinjauan Studi 12

No	Data Jurnal / Makalah	Keterangan
1	Judul	<i>Design and Implementation of REST API for Academic Information System</i>
2	Jurnal	<i>IOP Conference Series: Materials Science and Engineering</i> ISSN: 1757-8981
3	Volume, Nomor & Halaman	Volume 875 Nomor 1, 1-7
4	Bulan & Tahun	Juni 2020
5	Penulis	A A Prayogi, M Niswar, Indrabayu, M Rijal
6	Penerbit	IOP Publishing Ltd
7	Tujuan Penelitian	Untuk mengembangkan sebuah prototipe REST API pada sistem informasi akademik dan menganalisis kinerjanya dalam hal waktu respons, throughput, dan kehilangan paket data di bawah koneksi konkuren. Penelitian ini bertujuan untuk meningkatkan pertukaran data antar sistem informasi akademik guna memperbaiki layanan akademik dan keamanan sistem secara keseluruhan
8	Lokasi & Subjek Penelitian	Penelitian dilakukan di Universitas Hasanuddin, Makassar, Indonesia. Subjek penelitian melibatkan pengembangan prototipe REST API untuk sistem informasi akademik dengan fokus pada kinerja implementasi NodeJS dan PHP dalam mengelola data pegawai universitas
9	Metode yang digunakan	Prototipe REST API yang menggunakan dua teknologi server yaitu, NodeJS dan PHP. CRUD (Create, Read, Update Delete)
10	Perancangan Sistem	<ol style="list-style-type: none"> <li>1. Pengembangan prototipe REST API untuk sistem informasi akademik</li> <li>2. Penggunaan dua teknologi server utama: NodeJS dan PHP</li> <li>3. Implementasi CRUD (Create, Read, Update, Delete) untuk interaksi dengan data</li> </ol>

		<ol style="list-style-type: none"> <li>4. Penyediaan endpoint REST API untuk ekstraksi data pegawai dari tabel profilpegawai</li> <li>5. Penggunaan database sebagai repositori data dengan satu tabel profilpegawai</li> <li>6. Pembuatan dua jenis endpoint REST API: GET ALL dan GET PARAMS</li> <li>7. Penggunaan Apache JMeter untuk mensimulasikan hingga 1000 koneksi konkuren</li> <li>8. Analisis kinerja sistem melalui pengukuran waktu respons, throughput, dan kehilangan paket data</li> </ol>
11	Hasil Penelitian	<p>Hasil dari penelitian ini adalah:</p> <ol style="list-style-type: none"> <li>1. Waktu respons rata-rata untuk implementasi NodeJS lebih baik daripada PHP, dengan rentang 138ms hingga 156ms untuk NodeJS dan 244ms hingga 266ms untuk PHP.</li> <li>2. Implementasi NodeJS mencapai throughput 100% untuk hingga 1000 koneksi konkuren, sementara PHP hanya mencapai 48.70% throughput pada jumlah koneksi yang sama.</li> <li>3. Hasil pengukuran kinerja menunjukkan bahwa implementasi REST API NodeJS secara konsisten memiliki kinerja yang lebih baik dibandingkan dengan implementasi PHP</li> </ol>
12	Kekuatan Penelitian	<p>Kekuatan penelitian ini adalah :</p> <ol style="list-style-type: none"> <li>1. Penelitian dilakukan dengan menggunakan dua teknologi server yang umum digunakan, yaitu NodeJS dan PHP.</li> <li>2. Penggunaan metode pengukuran kinerja yang komprehensif melalui analisis waktu respons, throughput, dan kehilangan paket data.</li> <li>3. Penggunaan simulasi hingga 1000 koneksi konkuren memberikan gambaran yang baik tentang kinerja sistem dalam situasi yang padat.</li> <li>4. Hasil penelitian memberikan pemahaman yang jelas tentang perbandingan kinerja antara implementasi NodeJS dan PHP dalam konteks pengembangan REST API untuk sistem informasi akademik.</li> </ol>
13	Kelemahan Penelitian	<p>Kelemahan penelitian ini adalah:</p> <ol style="list-style-type: none"> <li>1. Penelitian hanya dilakukan pada satu tabel data yang relatif sederhana, sehingga hasilnya mungkin tidak sepenuhnya</li> </ol>

		<p>mencerminkan kinerja dalam skenario yang lebih kompleks.</p> <p>2. Tidak ada analisis yang mendalam tentang faktor-faktor lain yang dapat memengaruhi kinerja sistem, seperti ukuran database yang lebih besar atau kompleksitas endpoint yang lebih tinggi.</p> <p>3. Penelitian tidak mencakup aspek keamanan atau skalabilitas sistem, yang juga penting dalam pengembangan aplikasi web yang sebenarnya.</p>
14	Kesimpulan	<p>Kesimpulan dari artikel tersebut adalah bahwa dalam pengembangan REST API untuk sistem informasi akademik, implementasi NodeJS cenderung menunjukkan kinerja yang lebih baik daripada implementasi PHP dalam hal waktu respons dan throughput, terutama saat menangani banyak koneksi konkuren. Diperlukan penelitian lebih lanjut untuk menguji kinerja NodeJS dan PHP dalam skenario yang lebih kompleks dan dengan operasi I/O yang lebih intensif</p>

M. Tinjauan Studi 13

Tabel 2. 17 Tinjauan Studi 13

No	Data Jurnal / Makalah	Keterangan
1	Judul	<i>Web Service for Academic Information Systems</i>
2	Jurnal	<i>IOP Conference Series: Materials Science and Engineering</i> ISSN: 1757-8981
3	Volume, Nomor & Halaman	Volume 879 Nomor 1, 1-7
4	Bulan & Tahun	Juni 2020
5	Penulis	A R Kurniawan, L Warlina
6	Penerbit	IOP Publishing Ltd
7	Tujuan Penelitian	Menciptakan sistem informasi akademik berbasis layanan web/API untuk kemudahan akses lintas platform
8	Lokasi & Subjek Penelitian	Penelitian dilakukan di Departemen Manajemen dan Departemen Perencanaan Wilayah dan Kota, Universitas Komputer Indonesia. Subjek penelitian tersebut adalah penciptaan sistem informasi akademik berbasis layanan web/API
9	Metode yang digunakan	Metode yang digunakan adalah analisis deskriptif dan pendekatan model waterfall

10	Perancangan Sistem	<ol style="list-style-type: none"> <li>1. Tujuan: Menciptakan sistem informasi akademik berbasis layanan web/API.</li> <li>2. Metode: Analisis deskriptif dan pendekatan model waterfall.</li> <li>3. Manfaat: Memudahkan akses dan aplikasi sistem informasi lintas platform.</li> <li>4. Kontribusi: Berpotensi membantu dalam regulasi manajemen data di bidang pendidikan tinggi.</li> </ol>
11	Hasil Penelitian	<p>Hasil dari penelitian ini adalah:</p> <ol style="list-style-type: none"> <li>1. Sistem informasi akademik dapat diakses dan diaplikasikan secara online melalui berbagai platform.</li> <li>2. Penggunaan API memudahkan pengembangan sistem di universitas yang maju.</li> <li>3. Berpotensi memberikan kontribusi dalam regulasi manajemen data di bidang pendidikan tinggi.</li> </ol>
12	Kekuatan Penelitian	<p>Kekuatan penelitian ini adalah :</p> <ol style="list-style-type: none"> <li>1. Memanfaatkan API untuk memudahkan akses sistem informasi lintas platform.</li> <li>2. Menggunakan metode analisis deskriptif dan pendekatan model waterfall..</li> </ol>
13	Kelemahan Penelitian	<p>Kelemahan penelitian ini adalah:</p> <ol style="list-style-type: none"> <li>1. Tidak disebutkan secara eksplisit dalam kutipan yang disediakan.</li> </ol>
14	Kesimpulan	<p>Penelitian tersebut bertujuan untuk menciptakan sistem informasi akademik berbasis layanan web atau API, dengan menggunakan metode analisis deskriptif dan pendekatan model waterfall dalam pengembangan perangkat lunak. Hasil penelitian diharapkan dapat membantu dalam manajemen data akademik dan berpotensi memberikan kontribusi dalam regulasi manajemen data di bidang pendidikan tinggi.</p>

N. Tinjauan Studi 14

**Tabel 2. 18** Tinjauan Studi 14

No	Data Jurnal / Makalah	Keterangan
1	Judul	<i>Implementation of REST API in Web Service System for Medical Resume Provision</i>
2	Jurnal	<i>International Journal of Healthcare and Information Technology</i> ISSN: 3025-6933
3	Volume, Nomor & Halaman	Volume 1 Nomor 1, 34-48
4	Bulan & Tahun	Agustus 2023

5	Penulis	Mochammad Choirur Roziqin, Muhammad Syekhoni Noor, Abdullah Iskandar, Ananda Yuliantika
6	Penerbit	P3M Politeknik Negeri Jember
7	Tujuan Penelitian	Untuk mengembangkan sistem layanan <i>web</i> yang berfungsi sebagai penyedia resume medis melalui <i>RESTful API</i> , dengan fokus pada pertukaran data <i>resume</i> medis antar rumah sakit untuk meningkatkan kualitas layanan dan akurasi diagnosis berdasarkan data yang tersedia
8	Lokasi & Subjek Penelitian	Penelitian tersebut dilakukan di Departemen Kesehatan Politeknik Negeri Jember dan Waseda University. Subjek penelitian melibatkan pengembangan sistem layanan <i>web</i> untuk penyediaan <i>resume</i> medis melalui <i>RESTful API</i>
9	Metode yang digunakan	Menggunakan metode <i>RESTful API</i>
10	Perancangan Sistem	<ol style="list-style-type: none"> <li>1. Pengembangan sistem menggunakan <i>Web Service</i> berbasis <i>RESTful API</i>.</li> <li>2. Implementasi sistem untuk terhubung dengan sistem rekam medis lain.</li> <li>3. Fasilitasi pertukaran data <i>resume</i> medis antar rumah sakit.</li> <li>4. Meningkatkan kualitas layanan dan akurasi diagnosis.</li> <li>5. Penggunaan Visual Studio Code dan database MySQL.</li> <li>6. Skema permintaan data <i>resume</i> medis melalui <i>HTTP Request</i>.</li> <li>7. Pengolahan data <i>resume</i> medis dari rumah sakit dan pengembalian dalam format JSON.</li> <li>8. Penggunaan kredensial sistem untuk otorisasi akses.</li> <li>9. Verifikasi informasi kunjungan pasien dan pengiriman permintaan data ke rumah sakit.</li> <li>10. Pengembalian data <i>resume</i> medis dalam format JSON kepada rumah sakit yang meminta</li> </ol>
11	Hasil Penelitian	Hasil dari penelitian ini adalah: <ol style="list-style-type: none"> <li>1. Pengembangan sistem <i>web service</i> untuk menyediakan <i>resume</i> medis pasien.</li> <li>2. Fasilitasi pertukaran data <i>resume</i> medis antara rumah sakit menggunakan <i>RESTful API</i>.</li> <li>3. Solusi untuk masalah ketidaksesuaian informasi pasien dan data klinis.</li> </ol>

		4. Peningkatan kualitas layanan dan akurasi diagnosis berdasarkan data yang tersedia
12	Kekuatan Penelitian	Kekuatan penelitian ini adalah : <ol style="list-style-type: none"> <li>1. Pengembangan sistem <i>web service</i> yang inovatif untuk menyediakan <i>resume</i> medis pasien.</li> <li>2. Pemanfaatan <i>RESTful API</i> untuk pertukaran data medis antar rumah sakit.</li> <li>3. Solusi yang dapat meningkatkan kualitas layanan dan akurasi diagnosis.</li> </ol>
13	Kelemahan Penelitian	Kelemahan penelitian ini adalah: <ol style="list-style-type: none"> <li>1. Tidak disebutkan secara rinci tentang uji coba atau validasi sistem yang dikembangkan.</li> <li>2. Tidak ada pembahasan mendalam mengenai potensi hambatan atau tantangan dalam implementasi sistem</li> </ol>
14	Kesimpulan	Dari artikel tersebut, dapat disimpulkan bahwa pengembangan sistem <i>web service</i> menggunakan arsitektur <i>RESTful API</i> untuk menyediakan <i>resume</i> medis pasien merupakan solusi inovatif yang dapat meningkatkan pertukaran data medis antar rumah sakit, memperbaiki ketidaksesuaian informasi pasien, dan meningkatkan kualitas layanan serta akurasi diagnosis berdasarkan data yang tersedia.

O. Tinjauan Studi 15

Tabel 2. 19 Tinjauan Studi 15

No	Data Jurnal / Makalah	Keterangan
1	Judul	<i>CRUD Operation on WordPress Database using C# and REST API</i>
2	Jurnal	<i>International Journal of Applied Engineering and Management Letters (IAEML)</i> ISSN: 2581-7000
3	Volume, Nomor & Halaman	Volume 7 Nomor 4, 131-138
4	Bulan & Tahun	November 2023
5	Penulis	Sudip Chakraborty, P.S. Aithal
6	Penerbit	<i>Srinivas University</i>
7	Tujuan Penelitian	untuk memberikan informasi referensi tentang penggunaan database WordPress sebagai penyimpanan data dalam penelitian, serta memberikan contoh praktis agar peneliti dapat mengintegrasikan database WordPress dengan mudah dan usaha yang minimal.

8	Lokasi & Subjek Penelitian	Lokasi penelitian dilakukan di <i>Institute of Computer Science and Information Sciences, Srinivas University, Mangalore, India</i> . Subjek penelitian melibatkan pengembangan modul untuk operasi CRUD menggunakan C# dan REST API
9	Metode yang digunakan	Menggunakan metode CRUD dan REST API
10	Perancangan Sistem	<ol style="list-style-type: none"> <li>1. Sistem menggunakan perangkat lunak "Local" untuk instalasi WordPress.</li> <li>2. Dibuat rute kustom pada situs web WordPress untuk percobaan.</li> <li>3. Pengembangan modul CRUD menggunakan C# dan REST API.</li> <li>4. Aplikasi C# memiliki elemen GUI untuk interaksi dengan pengguna.</li> <li>5. Penggunaan modul "WordPress_REST.cs" untuk komunikasi dengan database WordPress.</li> <li>6. Terdapat fungsi terpisah untuk aktivitas <i>CREATE, READ, UPDATE, dan DELETE</i>.</li> <li>7. Modul "project_BP.cs" digunakan untuk pembangun objek JSON spesifik proyek.</li> <li>8. Penggunaan "Global.cs" untuk menyimpan variabel yang digunakan di berbagai modul.</li> <li>9. Elemen GUI seperti <i>Datagrid, list box, tombol, dan textbox</i> digunakan untuk interaksi dan tampilan data</li> </ol>
11	Hasil Penelitian	<p>Hasil dari penelitian ini adalah:</p> <ol style="list-style-type: none"> <li>1. Penelitian telah diuji menggunakan implementasi praktis dengan kinerja yang baik.</li> <li>2. Terdapat sedikit latensi jaringan dalam pengamatan.</li> <li>3. Waktu pembaruan konten tergantung pada bandwidth situs web, lalu lintas jaringan, spesifikasi sistem, dan jumlah aplikasi yang berjalan</li> </ol>
12	Kekuatan Penelitian	<p>Kekuatan penelitian ini adalah :</p> <ol style="list-style-type: none"> <li>1. Penelitian memberikan pendekatan yang berbeda dalam menyimpan data penelitian di dalam database situs web WordPress.</li> <li>2. Penggunaan REST API memungkinkan komunikasi dengan server WordPress untuk melakukan operasi CRUD.</li> <li>3. Penelitian memberikan panduan praktis dan kode implementasi yang tersedia untuk memudahkan integrasi.</li> </ol>
13	Kelemahan Penelitian	Kelemahan penelitian ini adalah:

		<ol style="list-style-type: none"> <li>1. Tidak disertakan penanganan kesalahan dalam kode, yang dapat mempengaruhi kehandalan sistem .</li> <li>2. Kinerja sistem dipengaruhi oleh faktor-faktor seperti bandwidth situs web, lalu lintas jaringan, dan spesifikasi sistem</li> </ol>
14	Kesimpulan	<p>Kesimpulan dari artikel tersebut adalah bahwa penggunaan REST API saat ini populer dan mudah diimplementasikan untuk berkomunikasi dengan server situs web WordPress dan melakukan operasi CRUD. Penelitian ini memberikan informasi praktis bagi peneliti yang ingin mengintegrasikan operasi database melalui REST API</p>

Penggunaan teknologi *RESTful API* dalam berbagai aplikasi seperti pengelolaan prestasi mahasiswa, pemesanan produk, dan integrasi data COVID-19 berhasil memberikan solusi yang efisien dan terintegrasi. Penerapan *framework* seperti Laravel dan Spring Boot dalam pengembangan sistem telah menunjukkan keberhasilan yang signifikan dengan hasil pengujian yang baik, termasuk waktu respon yang cepat dalam pertukaran data JSON. Perbandingan antara REST dan SOAP juga menunjukkan bahwa REST lebih unggul dalam pengiriman permintaan dan tanggapan. Kesimpulan ini menegaskan bahwa RESTful API tidak hanya inovatif dalam memodernisasi sistem tradisional tetapi juga efektif dalam meningkatkan efisiensi operasional dan memfasilitasi interaksi data lintas platform yang lancar.

## 2.6 Kerangka Pemikiran

Tabel 2. 20 Kerangka Pemikiran

Kondisi	Indikator	Hasil
<ul style="list-style-type: none"> <li>• Proses pencatatan biodata <i>database customer</i> masih dilakukan secara manual melalui <i>whatsapp</i> maupun <i>e-mail</i></li> <li>• Proses pemesanan barang oleh <i>customer</i> dilakukan secara tradisional dan rentan terjadinya data yang hilang dan kurang tepat</li> <li>• Pembatasan piutang <i>customer</i> yang rentan dan masih mudah lolos <i>order</i></li> </ul>	<ul style="list-style-type: none"> <li>• Menganalisa serta merancang <i>Web Service</i> yang mendukung dalam melakukan kegiatan transaksi</li> <li>• Merancang seluruh elemen yang dibutuhkan dalam pemesanan barang</li> <li>• Menentukan batasan piutang yang diperbolehkan untuk <i>customer</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>Web Service Aura Portal</i> yang dapat digunakan untuk mendaftarkan biodata <i>customer</i> baru</li> <li>• Pemesanan barang yang lebih mudah, lebih cepat, dan dapat dilakukan kapan saja</li> <li>• Pembatasan dengan melarang <i>customer</i> agar tidak dapat memesan barang kembali apabila sudah melebihi batas umur piutang yang telah ditentukan.</li> </ul>

## BAB III

### METODOLOGI PENELITIAN

#### 3.1. Gambaran Umum Perusahaan

##### 3.1.1. Sejarah Perusahaan

Aura Unggul didirikan pada November 2009 dengan tujuan untuk memenuhi kebutuhan akan *packaging* dengan kualitas terbaik dan harga yang terjangkau. Awal mulanya Aura Unggul melakukan produksi berupa *carton packaging* seperti *carton box*, dan partisi.

Pada tahun 2013, PT. Aura Unggul mulai memproduksi OPP *packaging tape* atau yang biasa dikenal dengan lakban berbahan OPP (Bening & Coklat). Lalu, pada tahun 2017 Aura Unggul mulai memproduksi *double tape*, *cloth tape* (lakban kain), *masking tape* (lakban kertas), dan *stretch film* (plastik *wrapping*).

Lalu pada tahun 2022 PT. Aura Unggul juga membuka divisi untuk produk baru yaitu *papercore*, dimana ini merupakan salah satu bahan dasar yang digunakan untuk memperlancar dalam menjalankan produksinya dalam kategori lakban. Hingga pada akhirnya, tahun 2023 PT. Aura Unggul mulai memasarkan produk dari *papercore* itu sendiri untuk perusahaan – perusahaan produksi lakban lain, kain, serta berbagai macam produk yang menggunakan *papercore* sebagai media produksinya.

Hingga saat ini PT. Aura Unggul sudah memiliki 3 buah pabrik serta 1 buah kantor pusat yang tersebar di daerah Kab. Tangerang serta Kota Tangerang. Dan, dikarenakan jarak antar cabang dari PT. Aura Unggul yang dikategorikan cukup jauh, maka dari itu sangat riskan untuk terjadinya kesalahan hingga hilang data dalam pertukaran dokumen antar cabang yang sedang terjadi.

### 3.1.2. Visi dan Misi PT. Aura Unggul

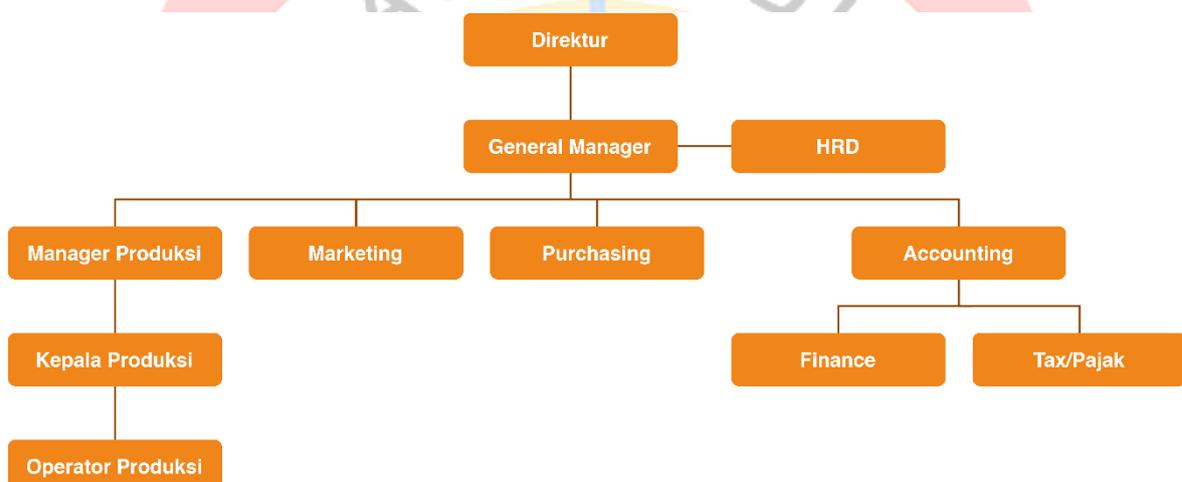
#### 3.1.2.1. Visi Perusahaan

Menjadi *sourcing packaging* terbaik bagi semua pelanggan kami.

#### 3.1.2.2. Misi Perusahaan

Selalu memberikan produk sesuai kebutuhan pelanggan dengan kualitas dan harga yang terbaik

### 3.1.3. Struktur Organisasi PT. Aura Unggul



**Gambar 3. 1** Struktur Organisasi Aura (Sumber: Direktur PT. Aura Unggul, 2023)

### 3.1.4. Tugas dan Wewenang

#### 3.1.4.1. Direktur

1. Mengidentifikasi, mengembangkan, dan menetapkan kebijakan perusahaan.
2. Mengawasi dan mengelola seluruh kegiatan operasional perusahaan, termasuk administrasi dan pengadaan barang.
3. Mengawasi dan mengembangkan seluruh sumber pendapatan dan asset perusahaan.
4. Menjadi pemimpin dan bertanggung jawab terhadap perusahaan.

5. Mewujudkan visi dan misi perusahaan.
6. Bertanggung jawab atas keuntungan dan kerugian yang mungkin dialami perusahaan.
7. Mempunyai wewenang untuk mengangkat dan memberhentikan karyawan perusahaan.
8. Mewakili perusahaan dalam kegiatan yang berhubungan dengan perusahaan.

#### **3.1.4.2. General Manager**

1. Menyusun strategi penjualan dan pemasaran.
2. Mengelola dan mengawasi anggaran pengeluaran perusahaan untuk semua kegiatan, termasuk operasional, produksi, dan persediaan.
3. Memberikan arahan kepada kepala departemen dengan memberikan petunjuk dan tujuan khusus terkait program kerja.
4. Menyampaikan perintah atau kebijakan yang dibuat oleh eksekutif untuk dilaksanakan kepada kepala departemen.
5. Mengawasi dan mengevaluasi setiap departemen perusahaan dan memastikan bahwa aktivitas yang dilakukan sudah sesuai.
6. Mengawasi seluruh kegiatan pengembangan karyawan mulai dari rekrutmen hingga pelatihan karyawan.
7. Melakukan penilaian kinerja karyawan untuk membantu mengetahui tercapai tidaknya tujuan perusahaan.

### 3.1.4.3. HRD (*Human Resource Development*)

1. Bertanggung jawab atas tugas administrative seperti pengelolaan data karyawan hingga *payroll*.
2. Menjamin kebijakan perusahaan berlaku bagi seluruh karyawan perusahaan.
3. Bertanggung jawab untuk mengelola hubungan karyawan.
4. Menarik dan mempertahankan karyawan yang berprestasi.
5. Mengawasi kebutuhan rekrutmen saat ini dan masa depan.
6. Bertanggung jawab atas pengelolaan pelatihan, pembelajaran dan pengembangan yang bertujuan untuk meningkatkan keterampilan kerja.
7. Menangani masalah kompensasi dan tunjangan karyawan.
8. Bertanggung jawab untuk menyiapkan dokumen yang dibutuhkan untuk penilaian kinerja karyawan dan memastikan bahwa setiap kinerja dievaluasi.

### 3.1.4.4. *Manager Produksi*

1. Merencanakan dan menjadwalkan produksi untuk menjalankan sistem produksi secara keseluruhan.
2. Bertanggung jawab memastikan dan mengkoordinasikan ketersediaan bahan produksi tepat waktu, termasuk pengendalian dan pengawasan kualitas bahan baku yang dibeli untuk keperluan produksi.
3. Memantau proses produksi. Hal ini mencakup pengawasan operator, proses dan kualitas hasil produksi untuk memastikan

bahwa proses telah berjalan sesuai SOP (*Standard Operating Procedure*), serta memastikan keselamatan kerja.

4. Bertanggung jawab untuk melakukan control kualitas produk.
5. Bertanggung jawab atas pemeliharaan dan kelancaran pengoperasian peralatan produksi.
6. Bertanggung jawab untuk menjamin kerahasiaan produksi.

#### **3.1.4.5. Kepala Produksi**

1. Bertanggung jawab mengawasi seluruh aktivitas proses produksi, mulai dari bahan baku hingga produk jadi.
2. Memastikan kualitas bahan baku selama proses dan produk jadi sesuai dengan standar perusahaan.
3. Memastikan konsistensi penerapan SOP kerja dan keselamatan kerja di bagian produksi.
4. Berkoordinasi dengan bagian lain yang terkait untuk menjamin kelancaran produksi.
5. Bertanggung jawab atas pemeliharaan mesin produksi yang ada.
6. Bertanggung jawab untuk membuat laporan kinerja dan hasil produksi secara berkala.

#### **3.1.4.6. Operator Produksi**

1. Menyiapkan dan mengoperasikan peralatan produksi dan mematuhi kebijakan keselamatan perusahaan.
2. Memastikan proses kerja telah dilakukan sesuai dengan instruksi yang diberikan.
3. Bertanggung jawab memastikan seluruh proses produksi berjalan lancar sesuai SOP yang berlaku.

4. Bertanggung jawab untuk mencapai tujuan produksi yang ditentukan.
5. Membantu mengatur alur produksi, memperbaiki sistem mesin, dan melaporkan kemungkinan perbaikan yang dapat mengganggu alur produksi.
6. Bertanggung jawab atas perawatan rutin mesin untuk menjamin kapasitas dan kualitas hasil produksi tetap terjaga sesuai dengan kualitas yang diharapkan.
7. Mengawasi penggunaan mesin dan peralatan produksi yang digunakan.
8. Mengawasi proses pengemasan, pengiriman dan melaporkan hasil produksi.
9. Memastikan lingkungan kerja selalu bersih dan rapi.

#### **3.1.4.7. Marketing**

1. Melaksanakan kebijakan manajemen mengenai pelayanan dan pemasaran produk perusahaan.
2. Mengadakan penawaran produk, baik produk lama maupun produk baru perusahaan.
3. Membuat kontrak dengan konsumen yang potensial.
4. Mengadakan penjualan produk dan menjalin hubungan yang baik dengan pelanggan.
5. Memberikan informasi kepada pelanggan mengenai SOP perusahaan dan persyaratan pemesanan produk.\
6. Menanggapi keluhan mengenai produk atau layanan perusahaan.

7. Memeriksa semua surat penjualan yang masuk dan keluar dari perusahaan.
8. Memeriksa laporan dan data terkait sistem *pre-order* penerimaan barang sesuai dengan kontrak penjualan.
9. Berkoordinasi dengan bagian produksi dan pengiriman untuk memastikan jumlah dan hasil produksi yang diterima pelanggan sesuai dengan *pre-order* sebelumnya.
10. Mengevaluasi status pembayaran dan menanggapi keterlambatan pembayaran.

#### **3.1.4.8. Purchasing**

1. Melakukan riset guna mencari *supplier* potensial.
2. Membandingkan penawaran *supplier*.
3. Melakukan negosiasi penawaran dari semua pihak sebelum melakukan perjanjian kontrak.
4. Memastikan kualitas produk yang dibeli sesuai dengan standar perusahaan, terutama bahan baku produksi.
5. Memastikan semua pengiriman barang yang dibeli tiba tepat waktu.
6. Memastikan Tingkat persediaan dan pemeliharaan kualitas produk yang dibeli.
7. Berkoordinasi dengan pihak Gudang untuk memastikan produk bahan baku yang dibeli disimpan dengan aman.
8. Membuat laporan terkait produk yang dibeli selama periode waktu tertentu, seperti laporan pembelain dan retur barang jika ada.

#### **3.1.4.9. *Accounting***

1. Membuat pembukuan keuangan, mulai dari enraca laba rugi hingga pengeluaran dan pemasukan perusahaan.
2. Membuat rekapan dan jurnal operasional perusahaan.
3. Melakukan penginputan data keuangan ke dalam sistem perusahaan.
4. Membuat pembukuan dari transaksi perusahaan yang terjadi dan dicatat dengna baik.
5. Melakukan rekonsiliasi dan penyesuaian data.
6. Membuat laporan keuangan perusahaan yang akan diberikan kepada pimpinan perusahaan.
7. Menghitung gaji karyawan perusahaan.
8. Melakukan pengarsipan dokumen keuangan perusahaan.

#### **3.1.4.10. *Finance***

1. Melakukan semua instruksi dari koordinator keuangan.
2. Menerima berbagai pembayaran dan setoran tagihan sebagai hasil dari transaksi yang berkaitan dengan operasional perusahaan.
3. Memastikan dan merekap seluruh transaksi perusahaan.
4. Melakukan pengarsipan dan mencatat seluruh bukti transaksi-transaksi yang berkaitan dengan kegiatan perusahaan.
5. Membantu Menyusun berbagai dokumen terkait kegiatan akuntansi dan keuangan perusahaan.
6. Membantu membuat laporan dengna baik dan benar.
7. Membantu dalam membayar tagihan kepada pihak eksternal yang telah melengkapi data dan persyaratan.

#### 3.1.4.11. *Tax/Pajak*

1. Melakukan tinjauan terkait sistem pajak perusahaan.
2. Harus selalu *update* terkait kebijakan baru perpajakan yang berlaku pada ruang lingkup perusahaan.
3. Bertanggung jawab atas pengarsipan, pengelolaan, dan menyiapkan dokumen perpajakan perusahaan dengan rapi.
4. Memperbarui *database* pajak perusahaan agar tersip dengan rapi.
5. Menghitung pajak perusahaan yang harus dibayar secara pasti.
6. Membayar pajak menggunakan *billing* dan melaporkannya dengan tepat waktu.

### 3.2. **Prosedur Sistem Berjalan**

#### 3.2.1. **Prosedur Penambahan *Customer* Baru**

1. Bagian *marketing* mencari *customer* baru dan melakukan hubungan komunikasi awal serta negosiasi dengan *customer* tersebut.
2. Setelah *customer* selesai bernegosiasi, selanjutnya *marketing* akan mengirimkan formulir data *customer* dalam format *Ms. Word Document* yang nantinya harus dilengkapi oleh *customer* tersebut.
3. Setelah *customer* sudah melengkapi form tersebut, form akan di kirimkan kembali kepada *marketing* yang nantinya akan di setuju dan di tandatangani oleh *General Manager* untuk dapat melanjutkan ke proses selanjutnya.
4. Setelah di tandatangani oleh *general manager*, maka dari itu tahap selanjutnya yaitu *marketing* akan memasukkan data *customer* tersebut kedalam *database customer* yang masih menggunakan *Ms. Excell* dan juga akan di *input* kedalam *database system accounting* yang sudah ada.

### 3.2.2. Prosedur Pemesanan

1. Tim *marketing* melakukan negosiasi dengan *customer* terlebih dahulu terkait pemesanan seperti; spek barang yang dipesan, harga yang diberikan, jumlah pemesanan, tanggal kirim, serta perjanjian pembayaran (T.O.P)
2. Setelah *deal*, maka *customer* di persilahkan untuk melakukan pembuatan *Purchase Order*, dan apabila *customer* tidak ingin atau tidak dapat membuat *Purchase Order*, maka tim *marketing* dapat membuat Konfirmasi Order *Customer* sebagai patokan dalam pembuatan surat jalan dan penagihan nantinya.
3. Setelah Pesanan penjualan di dapat, maka *marketing* akan melakukan *input data* kedalam sistem *accounting* untuk di proses oleh pabrik mulai dari pengiriman hingga surat jalan.
4. Setelah surat jalan terbit, maka barang akan dikirimkan oleh *driver* menuju ke *customer*.
5. Setelah barang sampai, *customer* akan menandatangani surat jalan yang menyatakan bahwa barang telah diterima dengan selamat, dan surat jalan akan dibawa kembali oleh *driver* untuk menuju ke proses selanjutnya.
6. Selanjutnya, *marketing*, akan memproses surat jalan tersebut untuk diberikan ke bagian *finance*

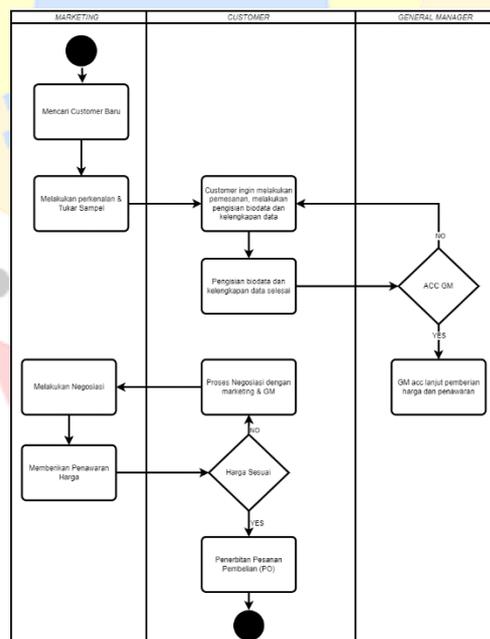
### 3.2.3. Prosedur Penagihan

1. Bagian *Finance* menerima surat jalan yang diberikan oleh *marketing*, dan sudah dilampirkan dengan *purchase order* atau konfirmasi order dari *customer*.

2. Setelah menerima surat jalan yang diberikan, selanjutnya *finance* akan melakukan *input data* pada sistem dan membuat *invoice* berdasarkan surat jalan dan lembar pemesanan yang diberikan.
3. Setelah tagihan terbit, tagihan tersebut akan diberikan kepada bagian pajak untuk dibuatkna faktur pajak yang nantinya tagihan tersebut akan di tanda tangani terlebih dahulu oleh *general manager*.
4. Setelah lampiran tagihan telah lengkap dan sudah di tandat tangani, maka tagihan tersebut akan dikirimkan melalui ekspedisi kepada *customer*.
5. Bagian AR akan konfirmasi tagihan tersebut apabila sudah diterima, lalu akan melakukan penagihan kepada *customer* apabila *customer* tersebut telah melewati batas waktu dan jumlah tagihan yang telah disepakati sebelumnya dengan pihak *marketing*.

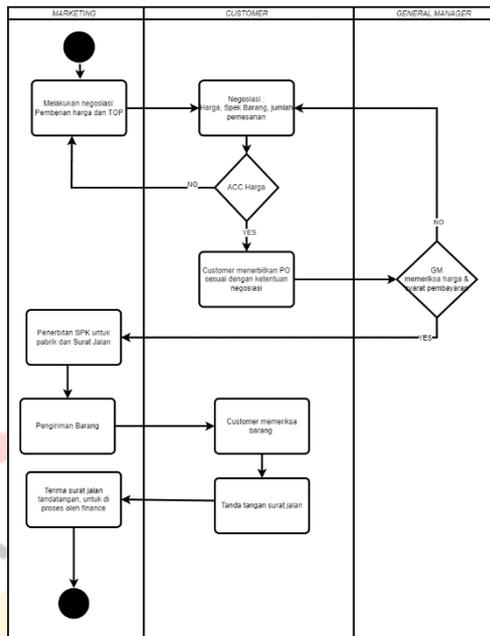
### 3.3. Activity Diagram

#### 3.3.1. Sistem Penambahan *Customer* Baru



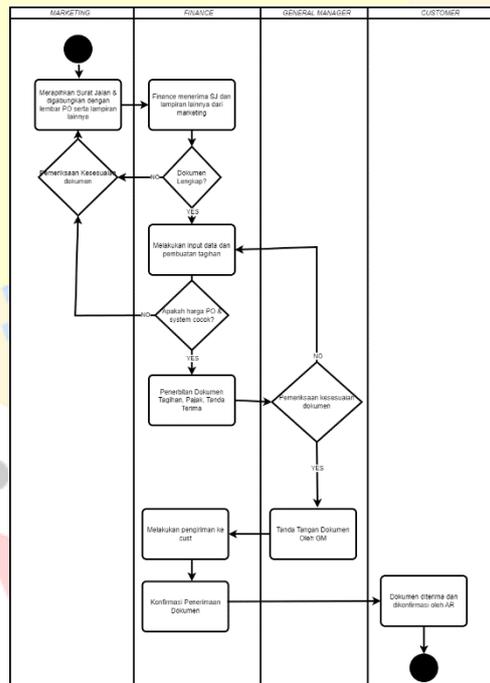
Gambar 3. 2 Activity Diagram sistem penambahan *customer* baru

### 3.3.2. Sistem Pemesanan



Gambar 3. 3 Activity Diagram sistem pemesanan barang

### 3.3.3. Sistem Penagihan



Gambar 3. 4 Activity Diagram sistem penagihan

### 3.4. Dokumentasi *Input* dan *Output*

#### 3.4.1. Dokumen Masukan

Dokumen masukan merupakan dokumen yang dibutuhkan untuk diolah dalam proses untuk menghasilkan output yang diinginkan. Berikut rincian dokumen masukan yang dibutuhkan:

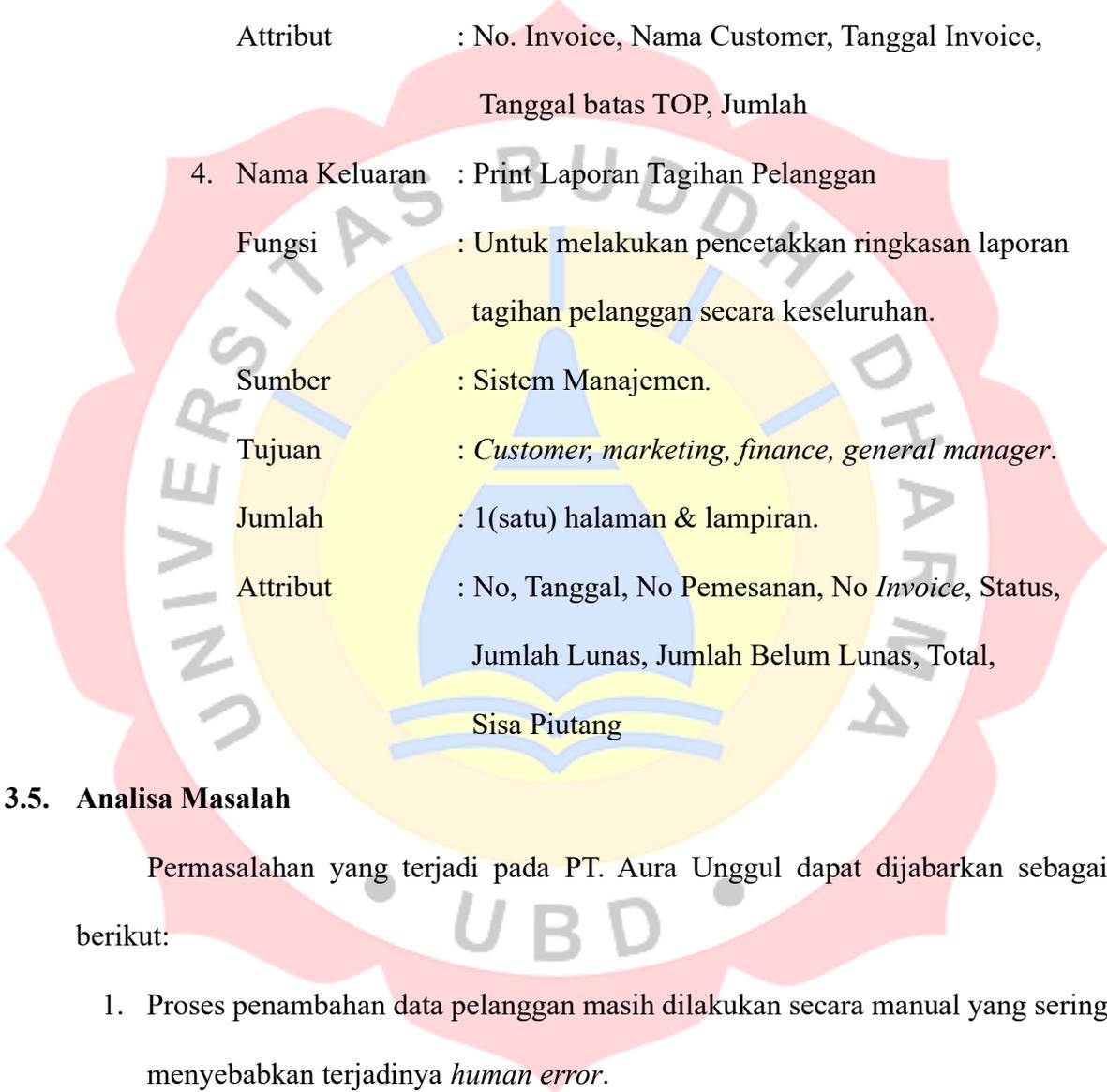
1. Nama Masukan : Biodata *Customer*  
Fungsi : Untuk menyimpan data pelanggan  
Sumber : *Customer*  
Tujuan : *Marketing & General Manager*  
Jumlah : 1(satu) lembar  
Atribut : ID, iduser, nama\_perusahaan, nama\_pemilik, no\_npwp, Alamat\_npwp, alamat\_pengiriman, alamat\_tagihan, bidang\_usaha, cp, no\_telp, email, sales, top, acc
2. Nama Masukan : Form *Order*  
Fungsi : Untuk menyimpan data pemesanan pelanggan  
Sumber : *Customer*  
Tujuan : *Marketing & General Manager*  
Jumlah : 1(satu) lembar & lampiran  
Atribut : id, userid, nama, Alamat, no\_telp, no\_po, tanggal, marketing, top, tgl\_kirim, acc, upload\_po
3. Nama Masukan : Form *Invoice*  
Fungsi : Untuk menyimpan data tagihan dan jumlah piutang  
Sumber : *Finance*  
Tujuan : *Customer*

Jumlah : 1(satu) lembar & lampiran  
Attribut : id, nama\_cust, no\_inv, tgl\_inv, tgl\_top, total,  
upload\_inv, upload\_fp, upload\_sj

### 3.4.2. Dokumen Keluaran

Dokumen keluaran adalah dokumen hasil pencatatan atau laporan serta dokumen yang mendukung kegiatan manajemen. Rincian yang diperlukan untuk dokumen keluaran adalah sebagai berikut:

1. Nama Keluaran : List Biodata *Customer*  
Fungsi : Untuk menampilkan biodata pelanggan yang sudah dimasukkan oleh pelanggan.  
Sumber : Sistem Manajemen  
Tujuan : *Marketing, Finance, General Manager.*  
Jumlah : 1(satu) halaman  
Attribut : ID, iduser, nama\_perusahaan, nama\_pemilik, no\_npwp, Alamat\_npwp, alamat\_pengiriman, alamat\_tagihan, bidang\_usaha, cp, no\_telp, email, sales, top, acc
2. Nama Keluaran : List Pesanan  
Fungsi : Untuk menampilkan data pesanan yang sudah masuk.  
Sumber : *Customer.*  
Tujuan : *Marketing, Finance, General Manager.*  
Jumlah : 1(satu) halaman & lampiran.  
Attribut : id, userid, nama, Alamat, no\_telp, no\_po, tanggal, marketing, top, tgl\_kirim, acc, upload\_po.

- 
3. Nama Keluaran : List *Invoice*
- Fungsi : Untuk menampilkan informasi tagihan pelanggan.
- Sumber : *Finance*.
- Tujuan : *Customer*.
- Jumlah : 1(satu) halaman & lampiran.
- Attribut : No. Invoice, Nama Customer, Tanggal Invoice, Tanggal batas TOP, Jumlah
4. Nama Keluaran : Print Laporan Tagihan Pelanggan
- Fungsi : Untuk melakukan pencetakan ringkasan laporan tagihan pelanggan secara keseluruhan.
- Sumber : Sistem Manajemen.
- Tujuan : *Customer, marketing, finance, general manager*.
- Jumlah : 1(satu) halaman & lampiran.
- Attribut : No, Tanggal, No Pemesanan, No *Invoice*, Status, Jumlah Lunas, Jumlah Belum Lunas, Total, Sisa Piutang

### 3.5. Analisa Masalah

Permasalahan yang terjadi pada PT. Aura Unggul dapat dijabarkan sebagai berikut:

1. Proses penambahan data pelanggan masih dilakukan secara manual yang sering menyebabkan terjadinya *human error*.
2. Proses penambahan data pelanggan yang terkesan Panjang, dan tidak efektif serta efisien dikarenakan dilakukan menggunakan tukar data file (*Ms. Office*) secara manual.

3. Seringnya terjadi *misscommunnication* antar divisi yang disebabkan karena data pelanggan yang tidak *update* secara otomatis.
4. Proses pemesanan dan pertukaran PO yang dilihat kurang efektif dikarenakan melalui pertukaran manual atau hanya sekedar pesan saja.
5. Proses pertukaran data PO antar divisi yang kurang lengkap, atau salah harga.
6. Proses penagihan yang masih dilakukan secara *manual* (By telpon) sehingga terjadinya *miscommunication* yang disebabkan pelanggan tidak menjawab telpon tersebut.
7. Proses pembatasan piutang pelanggan yang sudah jatuh tempo dilakukan kurang maksimal, sehingga pelanggan yang belum membayar tagihan sebelumnya masih akan dapat melakukan pemesanan.

### 3.6. Identifikasi Kebutuhan Sistem

Dari masalah yang ada, maka dapat disimpulakn bahwa kebutuhan sistem yang dibutuhkan sebagai berikut:

1. Sistem manajemen pelanggan dapat melakukan *login* dan *logout*.
2. Sistem manajemen pelanggan dapat melakukan pembatasan hak akses menu dan data pada tiap *user*.
3. Sistem manajemen pelanggan menggunakan keamanan berupa token yang sudah disetting sedemikian rupa akan melakukan pembaharuan token tiap 1 jam.
4. Sistem manajemen pelanggan dapat melakukan penambahan, perubahan, penghapusan data pelanggan.
5. Sistem manajemen pelanggan dapat menampilkan biodata pelanggan secara rinci dan lengkap.
6. Sistem manajemen pelanggan dapat melakukan penambahan, perubahan, penghapusan pada pemesanan oleh pelanggan.

7. Sistem manajemen pelanggan juga dapat menampilkan rincian pemesanan pelanggan secara lengkap dan jelas.
8. Sistem manajemen pelanggan dapat melakukan proses penambahan, perubahan, serta penghapusan data tagihan.
9. Sistem manajemen pelanggan dapat menampilkan jumlah hutang, sisa hutang, maupun Riwayat tagihan pelanggan secara rinci.
10. Sistem manajemen pelanggan dapat melakukan pembatasan pada order yang sudah melebihi batas TOP yang telah ditentukan.
11. Sistem manajemen pelanggan juga dapat melakukan ACC pelanggan, order, piutang yang hanya dapat dilakukan oleh administrator.

### 3.7. Metode REST API

*Representational State Transfer (REST) Application Programming Interface (API)* merupakan metode pendekatan arsitektur yang digunakan didalam *software development* untuk mengizinkan komunikasi antara berbagai sistem dan perangkat lunak melalui protokol HTTP, yang beroperasi berdasarkan prinsip REST, memiliki pendekatan yang sederhana dan ringkas untuk mentransfer data melalui jaringan.

Dalam koneksi *website* Aura Unggul *Portal* yang akan digunakan sebagai media penghubung antara *customer* dan *marketing* PT. Aura Unggul, membutuhkan adanya *web service* berbasis REST API yang mencakup fitur-fitur seperti *session login*, hak akses pengguna, *control* CRUD untuk beberapa menu seperti biodata pelanggan, menu pemesanan, serta menu penagihan dan piutang dengan cara kerjanya sebagai berikut:

### 3.7.1. *Session Login*

**Endpoint :** */sessions*

- **Metode HTTP :** POST
- **Deskripsi:** Untuk melakukan *login* pada aplikasi, pengguna akan melakukan pengisian *form* pada halaman *login* yang berisikan *userid* beserta *password*, yang selanjutnya API akan mengirimkan permintaan *login* dengan metode POST ke *endpoint /sessions* dengan kredensial yang dienkripsi.
- **Respons:** *Server* akan melakukan verifikasi kecocokan kredensial (*username* dan *password*), apabila data cocok dan ada dalam *database user*, maka dari itu *server* akan mengembalikan respon dengan memberikan token akses (dalam bentuk gabungan huruf dan angka secara acak dan random) jika autentikasi berhasil. Jika gagal, maka *server* akan memberikan respon berupa kode kesalahan dan pesan singkat dimana terjadinya kesalahan tersebut.

### 3.7.2. Mengambil *userid* pengguna

**Endpoint:** */getpermissions*

- **Metode HTTP:** GET
- **Deskripsi:** Setelah pengguna melakukan *login* dan berhasil masuk dengan menggunakan token, maka selanjutnya untuk menentukan hak akses pengguna terhadap fitur-fitur tertentu dalam sistem, terlebih dahulu *webservice* pada aplikasi akan secara otomatis mengirimkan permintaan GET ke *endpoint /getpermissions?token={token}*, di mana *{token}* adalah token yang dibawa saat melakukan *login* sistem oleh pengguna,

untuk mendapatkan *userid* dari pengguna yang sedang login dan melihat hak akses dari pengguna tersebut.

- **Respons:** *Server* akan melakukan validasi token akses dan mencocokkan data dari pengguna dengan data *userid* yang ada di *database sessions* pengguna yang nantinya akan mengembalikan dengan memberikan *userid* yang tepat untuk proses selanjutnya.

### 3.7.3. Mengambil Hak Akses Pengguna

**Endpoint:** */users*

- **Metode HTTP:** GET
- **Deskripsi:** Setelah mendapatkan *response* dari *endpoint* sebelumnya, maka sistem akan secara otomatis mengirimkan permintaan GET ke *endpoint /order?id={userid}*, di mana *{user\_id}* adalah ID pengguna yang sedang *login* yang telah di dapatkan dari *endpoint* sebelumnya.
- **Respons:** API akan mengembalikan berupa hasil dari pengecekan data *users* beserta data informasi *login users* yang berisikan nama lengkap, serta hak akses *user* yang nantinya akan dicocokkan dengan ketentuan pada aplikasi dan mengarahkan pengguna ke halaman menu yang tepat.

### 3.7.4. Mengambil Biodata Customer

**Endpoint:** */customers*

- **Metode HTTP:** GET
- **Deskripsi:** Untuk mengambil biodata pelanggan tertentu, aplikasi akan mengirimkan permintaan GET ke *endpoint /customer?iduser={userid}*, dimana *{userid}* merupakan ID yang dimiliki oleh *customer* yang tersimpan melalui file *cookie* dari *web browser* yang digunakan untuk mengakses beberapa menu.

- **Respons:** *Server* akan mengembalikan data biodata pelanggan dalam format JSON, sesuai dengan ketentuan yang diminta {userid}.

### 3.7.5. Pengisian Biodata *Customer*

**Endpoint:** /cust

- **Metode HTTP:** POST
- **Deskripsi:** Apabila pelanggan baru pertama kali *login*, atau *userid* yang digunakan oleh pelanggan belum memiliki biodata yang tercatat pada *database*, maka dari itu sistem secara otomatis akan mengarahkan halaman awal pelanggan menuju form pengisian biodata yang nantinya akan dikirimkan ke *database* menggunakan permintaan metode POST ke *endpoint* /cust dengan menyertakan *userid* dari perekam biodata tersebut dan detail biodata dalam format JSON
- **Respons:** *Server* akan memproses permintaan, memvalidasi data, dan mengembalikan status biodata (misalnya ID atau nama biodata telah sukses dibuat).

### 3.7.6. Memperbarui Biodata *Customer*

**Endpoint:** /cust

- **Metode HTTP:** PATCH
- **Deskripsi:** Apabila biodata pelanggan sudah ada, tetapi terjadi kesalahan atau ada *update* informasi tentang data pribadi pelanggan, maka disediakan fitur *form* untuk melakukan *update data*, yang nantinya akan dikirimkan ke *database* menggunakan metode PATCH ke *endpoint* /cust?id={id}, di mana {id} adalah ID *database* dari biodata *customer* yang ingin dirubah, dengan menyertakan *userid* dari pengguna dan detail biodata dalam format JSON

- **Respons:** *Server* akan memproses permintaan, memvalidasi data, dan mengembalikan status biodata (misalnya ID atau Nama biodata telah sukses di perbarui)

### 3.7.7. Menghapus Biodata *Customer*

**Endpoint:** /cust

- **Metode HTTP :** DELETE
- **Deskripsi:** Untuk menghapus biodata pelanggan, hanya *administrator* saja yang memiliki hak akses untuk menghapus biodata pelanggan apabila terjadi kesalahan dalam pengisian ataupun hal lainnya yang mengakibatkan pelanggan tersebut dihapus dari data pelanggan. *Administrator* hanya perlu menekan tombol *radio button* bergambar *delete* yang telah disediakan, lalu sistem akan secara otomatis mengirimkan permintaan DELETE ke *endpoint* /cust?id={id}, di mana {id} merupakan ID dari data terkait yang ingin dihapus.
- **Respons:** *Server* akan mengembalikan *respons* berupa status bahwa penghapusan telah berhasil dilakukan.

### 3.7.8. Mengambil Data Pemesanan

**Endpoint:** /order

- **Metode HTTP:** GET
- **Deskripsi:** Untuk mengambil data pesanan yang telah dimasukan sebelumnya, menggunakan metode HTTP: GET ke *endpoint* /order?userid={userid}, di mana {userid} adalah ID pengguna dari pemesan yang terkait yang melakukan *input data* maupun dengan melalui filter nama pelanggan (apabila login melalui *marketing* atau *administrator*).

- **Respons:** *Server* akan mengembalikan *respons* berupa status data ditemukan serta memberikan data yang diminta.

### 3.7.9. Menambahkan Data Pemesanan

**Endpoint:** /order

- **Metode HTTP:** POST
- **Deskripsi:** *customer* akan diberikan berupa *form* untuk menambahkan pemesanan pada menu *order* melalui tombol *radio button* yang telah disediakan. Setelah itu sistem akan mengirimkan permintaan POST ke *endpoint* /order dengan menyertakan *Userid* dari *customer* yang melakukan *input data* dan detail pemesanan dalam format JSON.
- **Respons:** *Server* akan memproses data, memvalidasi data, serta melakukan penambahan data pada *database* lalu mengembalikan *respons* kepada pelanggan yang telah menambahkan data dengan keterangan bahwa data telah ditambahkan dan mengembalikan *user* menuju ke halaman menu *order*.

### 3.7.10. Memperbarui Data Pemesanan

**Endpoint:** /order

- **Metode HTTP:** PATCH
- **Deskripsi:** Pelanggan yang sudah melakukan *input data* pemesanan, tetapi masih mengalami kesalahan *input data* ataupun ingin melakukan perubahan rincian pemesanan dapat melakukan pembaruan terhadap data pemesanan yang ada. Pada menu *order*, *customer* akan di sediakan tombol *radio button* yang berfungsi untuk melakukan *update* data secara spesifik dengan mengirimkan permintaan PATCH ke *endpoint* /order?id={id}, di mana {id} merupakan ID dari data terkait yang ingin

dilakukan perubahan dengan detail perubahan pemesanan dalam format JSON.

- **Respons:** *Server* akan melakukan proses data, memvalidasi kecocokan data, serta melakukan perubahan data pada *database* lalu mengembalikan *response* kepada *customer* berupa status perubahan data serta data terbaru yang telah dirubah.

### 3.7.11. Menghapus Data Pemesanan

**Endpoint:** /order

- **Metode HTTP:** DELETE
- **Deskripsi:** Apabila karna faktor tertentu *customer* diharuskan menghapus data pemesanan, maka hanya *user* dengan level *administrator* saja yang dapat melakukan penghapusan data pada menu pemesanan dengan mengirimkan permintaan DELETE ke *endpoint* /order?id={id}, di mana {id} merupakan ID dari data pemesanan yang akan dihapus.
- **Respons:** *Server* akan melakukan proses data, dan memvalidasi kecocokan dari *request* API baik sistem dengan *database*, lalu akan mengembalikan *respons* berupa pesan singkat status penghapusan bahwa data telah berhasil dihapus.

### 3.7.12. Mengambil Data Penagihan

**Endpoint:** /inv

- **Metode HTTP:** GET
- **Deskripsi:** Untuk mengambil informasi tagihan terkait dengan pemesanan tertentu berdasarkan nama *user*, aplikasi akan mengirimkan

permintaan GET ke *endpoint* /inv?nama\_cust={nama\_cust}, dimana {nama\_cust} adalah nama dari *customer* pemesanan yang terkait

- **Respons:** *Server* akan megembalikan informasi perihal tagihan dalam format JSON, termasuk detail seperti total tagihan, tanggal tagihan, informasi faktur pajak, surat jalan, lampiran *invoice*, tanggal jatuh tempo dan informasi terkait lainnya.

### 3.7.13. Menambahkan Data Penagihan

**Endpoint:** /inv

- **Metode HTTP:** POST
- **Deskripsi:** Untuk melakukan penambahan data penagihan, hanya *user* dengan level *administrator* saja yang dapat melakukan penambahan penagihan, dengan mengirimkan permintaan POST ke *endpoint* /inv, dengan menyertakan detail dari tagihan dalam format JSON.
- **Respons:** *Server* akan melakukan proses permintaan, validasi data, dan mengembalikan status dari hasil *input* data penagihan.

### 3.7.14. Memperbarui Data Penagihan

**Endpoint:** /inv

- **Metode HTTP:** PATCH
- **Deskripsi:** Untuk melakukan perubahan data apabila terjadi perubahan ataupun kesalahan dalam pengisian data penagihan sebelumnya, maka dapat dilakukan perubahan data yang hanya dapat dilaksanakan oleh *user* yang memiliki level *administrator*, dengan mengirimkan permintaan PATCH ke *endpoint* /inv?id=[id], di mana [id] merupakan ID dari data yang mengalami permasalahan dan harus melakukan perubahan data, dengan menyertakan detail dari perubahan data dalam format JSON.

- **Respons:** *Server* akan melakukan proses permintaan, validasi kecocokan data, serta mengembalikan status dari hasil perubahan data.

### 3.7.15. Menghapus Data Penagihan

**Endpoint:** /inv

- **Metode HTTP:** DELETE
- **Deskripsi:** Apabila terjadi suatu hal yang mengakibatkan data penagihan harus dihapus, maka hanya user yang memiliki hak akses *level administrator* saja yang dapat melakukan proses permintaan DELETE ke *endpoint* /inv?id={id}, di mana {id} merupakan ID dari data yang akan dihapus.
- **Respons:** *Server* akan melakukan proses permintaan, serta melakukan konfirmasi penghapusan data, dan mengembalikan status dari penghapusan data apabila telah dikonfirmasi oleh pengguna untuk melakukan penghapusan data.

### 3.7.16. Melakukan Proses Logout

**Endpoint:** /sessions

- **Metode HTTP:** DELETE
- **Deskripsi:** Apabila *user* telah selesai menggunakan sistem, maka diharapkan untuk melakukan proses *logout* untuk menghapus data *login* dan *sessions* yang tersimpan, dengan mengirimkan permintaan DELETE ke *endpoint* /sessions?id={sessionid}, di mana {sessionid} merupakan ID dari *sessions* yang tersimpan saat *user* melakukan *login* sebelum penggunaan sistem, serta melakukan *sessions destroy* untuk menghapus *cookie* dari *login* pengguna.

- **Respons:** *Server* akan melakukan proses permintaan, melakukan validasi kebenaran data, serta konfirmasi *logout user*, lalu melakukan penghapusan data *sessions* pada *tblsessions* serta pada *cookie* yang dipakai oleh *user*.

Adapun dapat di simpulkan alur kerja secara keseluruhan menjadi :

1. **Autentikasi :** Sistem aplikasi pengguna melakukan *login* dengan mengirimkan kredensial.
2. **Otentikasi:** *Server* memverifikasi kredensial dan mengembalikan token akses.
3. **Hak Akses:** Sistem aplikasi pengguna menggunakan hak akses token untuk meminta dan memvalidasi data hak akses pengguna.
4. **Operasi Data:** Sistem aplikasi pengguna juga menggunakan *endpoint* yang sesuai untuk mengambil (GET), membuat (POST), merubah (PATCH), menghapus (DELETE) data seperti biodata customer, pemesanan, dan penagihan dengan melalui pengecekan hak akses dari masing-masing pengguna untuk mengatasi pengguna dalam penggunaan *endpoint* yang kurang tepat.
5. **Respon:** *Server* merespon dengan data yang diminta atau status operasi yang berhasil ataupun gagal.

REST API dalam konteks *website* dengan bantuan sistem *webservice* ini memanfaatkan prinsip-prinsip HTTP dan format data yang ringan seperti JSON untuk memudahkan komunikasi antara klien (aplikasi) dan *server (database)*. Hal ini memungkinkan aplikasi untuk melakukan operasi-operasi terkait dengan data *customer* dan manajemen pemesanan secara efektif, efisien, dan terstruktur.

### 3.8. Requirement Elicitation

RE (*Requirement Elicitation*) merupakan kebutuhan dan keinginan dari calon pengguna program, sehingga nantinya program akan dibuat sesuai dengan kebutuhan pengguna serta dapat optimalisasi penggunaan program tersebut. Setelah melakukan wawancara serta observasi kepada responden yang menginginkan sebuah aplikasi yang dapat digunakan untuk membantu dalam manajemen data pelanggan serta pembatasan transaksi sesuai dengan kesepakatan awal dengan pelanggan, maka kebutuhan responden untuk aplikasi ini dapat dijabarkan sebagai berikut:

**Tabel 3. 1** Tabel *Requirement Elicitation*

NO	Analisa Kebutuhan Sistem	Keterangan
	Saya ingin sistem dapat:	
1	Tampilan yang mudah digunakan dan dimengerti	
2	Dapat diakses secara <i>realtime</i> , dimanapun dan kapanpun	
3	Dapat mengelola biodata dari pelanggan (nama, alamat, no telp, dsb)	
4	Dapat digunakan untuk <i>upload</i> lampiran dari biodata pelanggan (NPWP, KTP, serta surat yang diperlukan lainnya)	
5	Dapat digunakan untuk mengelola transaksi pelanggan	
6	Dapat digunakan untuk <i>upload</i> dan menyimpan data lampiran <i>purchase order</i> dari pelanggan	
7	Dapat digunakan untuk mengelola <i>invoice</i> tagihan pelanggan	
8	Dapat melakukan <i>upload</i> lampiran <i>invoice</i> pelanggan (Faktur Pajak, Surat jalan, <i>Invoice</i> )	
9	Dapat menampilkan jumlah piutang yang masih ada dan jatuh tempo	

10	Dapat melakukan pembatasan piutang pelanggan	
11	Harus ada catatan dari pemesanan pelanggan	
12	Harus ada catatan dari tagihan pelanggan	
13	Memiliki fitur <i>multiuser</i>	
14	Memiliki fitur <i>admin</i>	

### 3.9. Jadwal Penelitian

**Tabel 3. 2** Tabel Jadwal Penelitian (*Gant Chart*)

Kegiatan	2024															
	Maret			April				Mei				Juni				
	3	4	5	1	2	3	4	1	2	3	4	1	2	3	4	
Pengumpulan Data																
Observasi																
Tinjauan Pustaka																
Wawancara																
Pembuatan Sistem Usulan																
Testing																
Evaluasi																
Penilaian																
Dokumentasi																