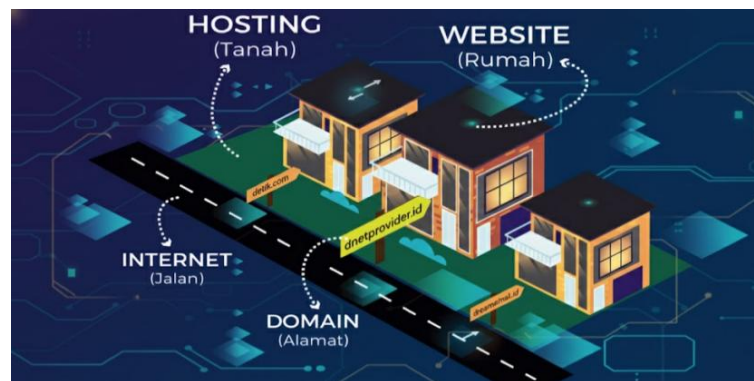


BAB II

LANDASAN TEORI

2.1. Konsep Dasar Website

Website ialah sekumpulan halaman yang menyajikan berbagai jenis informasi[9], Website pada dasarnya adalah sekumpulan halaman yang saling terhubung dan berisi berbagai jenis informasi, mulai dari teks, gambar, animasi, audio, video, hingga gabungan dari semua format tersebut [9]. World Wide Web, atau yang lebih dikenal dengan istilah Web, merupakan sebuah layanan penyedia informasi yang memanfaatkan konsep *hyperlink* sebagai penghubung antar konten. Konsep tautan ini memudahkan pengguna sering disebut *surfer* untuk menjelajahi berbagai informasi di internet dengan lebih cepat dan efisien. Melalui Web, pengguna dapat Memberikan penekanan pada kata, frasa, maupun gambar dalam sebuah dokumen untuk kemudian dihubungkan ke elemen lain, seperti dokumen berbeda, potongan teks tertentu, klip video, hingga berkas audio [10].



Gambar 2.1. Komponen Website

Teknologi ini kini berkembang sangat cepat dan menjadi salah satu media komunikasi yang paling banyak digunakan. Hampir seluruh aktivitas manusia tidak terlepas dari bantuan perangkat komputer yang memudahkan proses

pengolahan data, pencarian informasi, hingga penyebaran pesan secara luas. Kehadiran Website berfungsi sebagai media promosi yang efektif, karena memungkinkan individu maupun perusahaan menampilkan produk, layanan, serta identitas mereka kepada khalayak yang lebih luas tanpa batasan ruang dan waktu. Komputer dan Website juga membuat masyarakat dapat memperoleh, mengelola, dan membagikan informasi dengan jauh lebih cepat, praktis, dan terstruktur dibandingkan metode konvensional.

Komponen utama yang membentuk sebuah Website antara lain domain, hosting, dan konten. Domain merupakan alamat atau nama khusus yang digunakan agar pengguna lebih mudah menemukan dan mengakses situs tersebut. Hosting berperan sebagai lokasi penyimpanan seluruh data Website, seperti *file script*, *database*, maupun berkas pendukung lainnya. Sementara itu, konten adalah bagian yang memuat informasi yang ingin disampaikan kepada pengunjung; tanpa konten, keberadaan Website tidak akan Memberikan nilai atau manfaat apa pun. Secara teknis, sebuah Website tersusun dari sejumlah halaman yang berada dalam satu domain atau subdomain yang saling terorganisasi. Pada umumnya, saat pengguna pertama kali membuka sebuah situs, mereka akan diarahkan ke halaman beranda (*homepage*) dengan memasukkan URL melalui peramban Web. Halaman utama ini biasanya menjadi pintu masuk menuju berbagai halaman lain yang masih berada dalam satu struktur situs. Setiap halaman tersebut saling terhubung sehingga memudahkan pengunjung untuk menelusuri isi Website secara menyeluruh [11].

2.1.1. Sejarah Website

Awal mula perkembangan Website bermula dari gagasan seorang ilmuwan asal Inggris bernama Tim Berners-Lee. Ketertarikannya pada dunia komputasi tidak lepas dari latar belakang keluarganya, sebab kedua orang tuanya juga berprofesi sebagai ilmuwan komputer pada masa-masa awal perkembangan teknologi tersebut. Berners-Lee menciptakan konsep Website dengan tujuan mempermudah para peneliti di tempat ia bekerja untuk mengakses, membagikan, serta bertukar informasi secara lebih cepat dan terstruktur. Tonggak penting dalam sejarah ini terjadi pada 30 April 1993, ketika CERN laboratorium fisika terkenal yang berlokasi di Swiss secara resmi mengumumkan bahwa teknologi Website dapat digunakan secara bebas oleh publik.



Gambar 2.2. Evolusi Tampilan Website dari Teks ke Interaktif

Sebelum peluncuran resmi Website untuk publik, Tim Berners-Lee pada tahun 1990 telah merumuskan tiga fondasi utama yang menjadi dasar terbentuknya teknologi Web modern, yaitu:

1. HTML (*HyperText Markup Language*), Bahasa markup yang digunakan untuk membangun struktur dan tampilan halaman Web sehingga informasi dapat disajikan secara terorganisasi.

2. URI (*Uniform Resource Identifier*) / URL (*Uniform Resource Locator*), Sistem penamaan atau alamat unik yang dipakai untuk mengenali dan mengakses setiap sumber daya di dalam Web.
3. HTTP (*HyperText Transfer Protocol*), Protokol komunikasi yang memungkinkan pengambilan, pengiriman, dan pertukaran data antar halaman Web sehingga seluruh situs dapat diakses dan saling terhubung.

Peluncuran Website untuk publik pada tahun 1993 menandai awal percepatan kemajuan teknologi digital. Konsep dasar yang diperkenalkan oleh Tim Berners-Lee yaitu HTML, URI/URL, dan HTTP, kemudian dikembangkan menjadi fondasi Web modern. Sejak itu, Website berperan penting sebagai media penyebaran informasi dan komunikasi global, sekaligus mengubah cara manusia bekerja, belajar, dan berinteraksi di era digital [12].

2.1.2. Jenis Website

Secara umum jenis Web berdasarkan penampilan dan respon dibagi menjadi 3 jenis, yaitu:

1. Website statis, yaitu jenis situs yang tampilannya cenderung tetap dan jarang mengalami perubahan. Jika ada pembaruan, biasanya hanya pada desain visual, bukan pada isi informasinya. Konten di dalamnya bersifat tetap dan tidak sering dimodifikasi. Contoh umum Website statis ialah halaman profil perusahaan atau organisasi.
2. Website dinamis, merupakan situs yang kontennya dapat berubah dan diperbarui secara berkala sesuai kebutuhan, perkembangan bisnis, atau tren yang sedang berlangsung. Tampilan dan fiturnya lebih interaktif

karena sering dilengkapi fasilitas seperti kolom komentar, ruang obrolan, hingga sistem manajemen konten. Contoh Website dinamis antara lain blog, portal berita *online*, toko daring (e-commerce), dan berbagai jenis sistem informasi.

3. Website interaktif, dirancang agar pengguna dapat saling berkomunikasi, Memberikan respon, dan terlibat langsung dalam aktivitas di dalam situs. Jenis Website ini biasanya termasuk ke dalam platform media sosial yang mengutamakan interaksi antar pengguna, seperti Facebook, Instagram, Twitter, dan platform sosial lainnya.

Dengan memahami berbagai jenis Website, mulai dari statis, dinamis, hingga interaktif, pengembang dan pengguna dapat menyesuaikan pilihan situs sesuai kebutuhan dan tujuan. Website statis cocok untuk penyampaian informasi yang tetap, Website dinamis mendukung pembaruan konten secara rutin, sedangkan Website interaktif mendorong keterlibatan aktif pengguna [13]. Pemilihan jenis Website yang tepat akan membantu efektivitas penyampaian informasi, interaksi, dan promosi secara *online*.

2.1.3. Fungsi dan Manfaat Website

Web memiliki berbagai fungsi umum yang mendukung berbagai kebutuhan. Berikut adalah beberapa fungsi umum Web:

1. Sumber Informasi : Website berfungsi sebagai sumber informasi yang luas dan mudah diakses. Website ini juga menyediakan informasi tentang produk, layanan, kegiatan, dan berita terkini kepada pengguna

2. Komunikasi dan Interaksi: Yang memungkinkan komunikasi dan interaksi antara pengguna dan pemilik Website
3. Media Promosi: Website berfungsi sebagai platform promosi dimana perusahaan dapat menjual produk dan jasa secara *online*.

Dengan memahami berbagai fungsi dan manfaat Website, baik sebagai sumber informasi, media komunikasi, maupun sarana promosi, pemilik situs dapat memanfaatkannya secara optimal sesuai kebutuhan. Website tidak hanya mempermudah penyampaian informasi, tetapi juga meningkatkan interaksi dengan pengguna dan memperluas jangkauan promosi secara digital. Penggunaan Website yang tepat akan mendukung efektivitas kegiatan informasi, komunikasi, dan pemasaran di era modern [14].

2.2. Sistem Berbasis Web *Broadcast*

Broadcast merupakan istilah dalam bahasa Inggris yang erat kaitannya dengan dunia siaran, terutama dalam bidang komunikasi dan teknologi. Istilah ini merujuk pada proses penyampaian atau penyaluran suatu informasi kepada banyak penerima secara bersamaan. Contohnya, dalam aplikasi pesan instan seperti WhatsApp terdapat fitur *broadcast* yang memungkinkan pengguna mengirim pesan secara serentak ke banyak kontak, sehingga informasi dapat tersebar dengan cepat dan efisien [15]. *Broadcast message* merupakan jenis pesan yang dapat dikirimkan ke banyak penerima secara bersamaan tanpa perlu mengirimkannya satu per satu secara manual. Metode ini dianggap sangat efektif, terutama ketika digunakan untuk kegiatan pemasaran melalui WhatsApp, karena

proses penyampaian informasinya jauh lebih cepat dan efisien dibandingkan dengan beberapa platform media sosial lain, seperti Instagram [16].

Fitur ini sangat bermanfaat dalam konteks komunikasi bisnis maupun layanan publik karena mempermudah distribusi informasi tanpa perlu mengirim pesan satu per satu secara manual. Selain itu, pesan *broadcast* hanya diterima oleh penerima yang telah menyimpan nomor pengirim, sehingga Memberikan perlindungan privasi sekaligus mengurangi risiko spam. Pemanfaatan fitur *broadcast* dalam strategi pemasaran memungkinkan perusahaan menjangkau pelanggan secara efisien, menjaga interaksi langsung, dan meningkatkan respons pengguna terhadap pesan yang dikirim [15].



Gambar 2.3. Mengirim Pesan Massal dengan Mudah

Selain meningkatkan efisiensi dalam penyampaian informasi, penerapan sistem *broadcast* yang terintegrasi dengan WhatsApp juga Memberikan manfaat dalam pengelolaan komunikasi digital perusahaan. Dengan dukungan fitur seperti label kontak, katalog produk, serta riwayat percakapan yang tersimpan otomatis, perusahaan dapat mengatur komunikasi pelanggan dengan lebih tersusun. WhatsApp tidak hanya berfungsi sebagai sarana promosi, tetapi juga sebagai media pengarsipan data pelanggan yang membantu pelaku usaha menjaga

konsistensi layanan dan meningkatkan kepercayaan pengguna. Melalui penerapan sistem ini, komunikasi menjadi lebih efektif, pesan lebih tepat sasaran, dan seluruh aktivitas interaksi dapat dilihat secara transparan, sehingga mendukung profesional dalam pelayanan bisnis digital [17].

Konten pesan *broadcast* perlu dirancang agar *informatif* dan menarik perhatian penerima. Informasi yang disampaikan sebaiknya padat, jelas, dan mudah dipahami agar tujuan komunikasi tercapai dengan efektif. Penggunaan elemen multimedia seperti gambar, video, dan desain visual yang interaktif dalam pesan *broadcast* dapat meningkatkan keterlibatan *audiens*, terutama di kalangan generasi muda yang terbiasa dengan media digital. Oleh karena itu, pesan yang bersifat personal serta menampilkan elemen visual yang menarik cenderung lebih efektif dalam menarik perhatian dan membangun interaksi dengan penerima [18].

Perancangan sistem *Web Broadcast* merupakan langkah yang bertujuan untuk mengirimkan pesan atau informasi secara massal kepada banyak pengguna secara otomatis dan terstruktur. Sistem ini biasanya terintegrasi dengan *Application Programming Interface (API)* seperti WhatsApp Business API untuk mengotomatisasi proses pengiriman pesan. Dengan adanya sistem ini, perusahaan dapat menyampaikan informasi promosi, notifikasi layanan, atau pemberitahuan kepada pelanggan dengan tingkat efisiensi yang tinggi tanpa harus melakukan pengiriman pesan satu per satu secara manual [19].

Sistem *broadcast* berbasis Web dirancang dengan pola *client server*, di mana pengguna mengakses fitur aplikasi melalui tampilan antarmuka pada *browser*, sedangkan sisi *server* berfungsi untuk menjalankan logika bisnis dan mengatur proses pengiriman pesan ke setiap penerima. Di dalam sistem ini

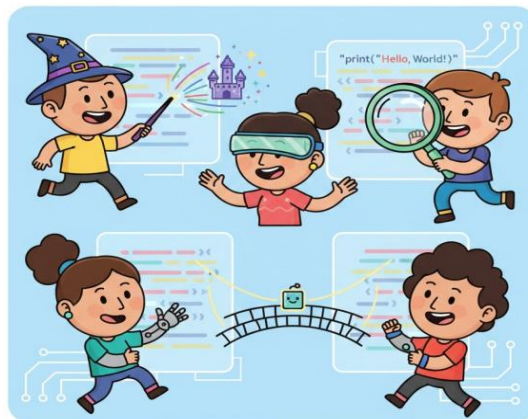
disertakan basis data yang berperan dalam menyimpan data penting seperti daftar penerima, riwayat pengiriman, serta status setiap pesan agar seluruh aktivitas dapat terpantau dengan baik. Proses pengiriman dilakukan melalui integrasi API (*Application Programming Interface*), yang memungkinkan aplikasi berkomunikasi langsung dengan berbagai platform digital seperti WhatsApp dan sebagainya. Penerapan arsitektur *client-server* pada sistem *broadcast* dengan memanfaatkan mekanisme Redis Pub/Sub terbukti mampu mempercepat proses distribusi pesan secara waktu nyata dan mencatat hasil pengiriman secara otomatis. Pendekatan ini meningkatkan ketepatan, efisiensi, serta stabilitas sistem dalam mendukung pengiriman pesan massal secara terkoordinasi [20].

Dalam pembuatan sistem *broadcast*, faktor performa, keamanan, dan skalabilitas menjadi fokus utama. Nah sistem harus mampu menangani pengiriman pesan dalam jumlah besar secara bersamaan tanpa menurunkan kecepatan kinerja. Oleh karena itu, dibutuhkan pemilihan bahasa pemrograman dan arsitektur sistem yang efisien agar proses *broadcast* berjalan stabil dan aman [21].

2.2.1. Bahasa Pemrograman dalam Pembuatan Sistem Web *Broadcast*

Pemrograman Web tersusun dari dua istilah, yaitu pemrograman dan Web. Pemrograman merujuk pada proses merancang serta menyusun rangkaian perintah secara sistematis untuk menyelesaikan suatu masalah dengan memanfaatkan bahasa pemrograman tertentu. Sementara itu, Web merupakan wadah yang berisi berbagai bentuk informasi mulai dari teks, gambar, audio, hingga animasi yang dapat diakses melalui jaringan internet dengan mengikuti tautan yang tersedia

menggunakan *browser*. Dengan demikian, pemrograman Web dapat diartikan sebagai aktivitas menuliskan instruksi menggunakan bahasa pemrograman yang dirancang untuk dijalankan melalui Web *browser* yang terhubung ke internet. *Output* dari proses ini adalah terciptanya sebuah aplikasi berbasis Web yang dapat digunakan kapan pun dan di mana pun oleh perusahaan, organisasi, maupun lembaga lain sesuai kebutuhan [22].



Gambar 2.4. Bahasa Pemrograman Sulap Dunia Digital

Bahasa pemrograman ialah sarana yang digunakan untuk Memberikan serangkaian perintah kepada komputer agar sistem dapat mengolah data dan menghasilkan *Output* sesuai yang diinginkan oleh programmer. Pemrograman Web dapat dipandang sebagai proses pembuatan atau pengembangan aplikasi berbasis Web dengan memanfaatkan bahasa pemrograman tertentu, sehingga Website mampu mengelola data dan menampilkan informasi sesuai kebutuhan pemilik atau pengelolanya [13]. Dalam perancangan Web, bahasa pemrograman umumnya dikelompokkan ke dalam dua kategori utama, yaitu *frontend* dan *backend*, yang masing-masing memiliki peran dan fungsi spesifik dalam membangun sebuah Website.

2.2.2. Bahasa Pemrograman *Front-End*

Bahasa pemrograman *Front-End* merupakan jenis bahasa yang digunakan khusus untuk mengatur tampilan dan interaktivitas sebuah halaman Web. Pengembang *Front-End* bertugas menciptakan pengalaman pengguna yang menarik, responsif, dan nyaman diakses melalui *browser*. Dengan penggunaan bahasa pemrograman *Front-End*, sebuah situs Web dapat tampil lebih dinamis, interaktif, dan mudah digunakan oleh pengunjungnya [23]. Beberapa bahasa pemrograman utama yang digunakan dalam *Front-End* Web development adalah:

a. HTML (*HyperText Markup Language*)



Gambar 2.5. Kode HTML Sederhana

HTML menjadi fondasi utama setiap halaman Web. Bahasa ini memungkinkan pengembang untuk menyusun struktur dan konten halaman dengan menggunakan berbagai elemen, seperti teks, gambar, dan atribut tertentu. HTML berperan dalam menentukan teks, gambar, tautan, serta komponen lain yang membentuk sebuah halaman Web [24].

b. CSS (Cascading Style Sheets)



Gambar 2.6. *Before & After* Tampilan Halaman Web

CSS digunakan untuk mengatur dan memperindah tampilan halaman Web. Dengan CSS, pengembang bisa menentukan warna, ukuran dan jenis font, tata letak, serta berbagai efek visual lainnya. CSS bekerja beriringan dengan HTML untuk Memberikan kontrol lebih besar terhadap desain dan estetika halaman Web [25].

c. JavaScript



Gambar 2.7. JavaScript

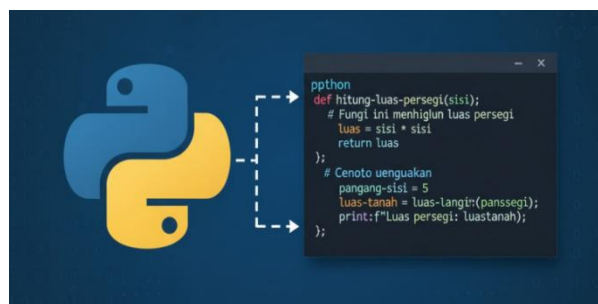
JavaScript pertama kali diperkenalkan pada tahun 1995 sebagai sarana untuk menambahkan program pada halaman Web melalui *browser Netscape Navigator*. Meskipun awalnya dirancang untuk *browser*, JavaScript kini juga digunakan di berbagai platform lain. Beberapa basis data, seperti *MongoDB* dan *CouchDB*, memanfaatkan JavaScript sebagai bahasa *scripting* dan *query*. Selain itu, platform pengembangan *desktop* dan *server*, terutama proyek *NODE.js*,

menyediakan lingkungan untuk menjalankan JavaScript di luar *browser*. Bahasa pemrograman ini memungkinkan pembuatan efek animasi, validasi formulir, navigasi dinamis, dan interaksi dengan pengguna secara *real-time* [26].

2.2.3. Bahasa Pemrograman *Back-end*

Bahasa pemrograman *back-end* adalah jenis bahasa yang digunakan untuk membangun logika dan fungsi di sisi *server* pada sebuah aplikasi Web. Tugas utamanya meliputi pengelolaan data, menambahkan, memperbarui, atau menghapus data pada sistem atau basis data, serta menangani proses di balik layar untuk mengolah *database* dan *server* yang tidak terlihat oleh pengguna. Berbeda dengan *Front-End* yang fokus pada tampilan dan interaktivitas di *browser*, pemrograman *back-end* berjalan di *server* untuk memastikan Website berjalan secara dinamis, aman, dan efisien [27]. Beberapa bahasa pemrograman populer yang sering digunakan untuk pengembangan *back-end* antara lain:

a. Python

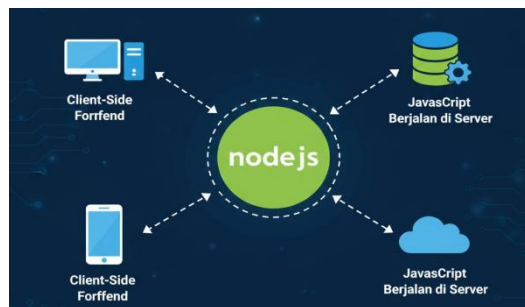


Gambar 2.8. Bahasa Pemrograman Python

Python menjadi pilihan utama karena dukungan pustaka seperti Flask yang memungkinkan integrasi *RESTful* API serta otomatisasi pengiriman pesan. Selain itu, Python dikenal memiliki sintaks yang sederhana dan kemampuan pengolahan

data yang efisien, sehingga sangat cocok untuk mengelola logika bisnis dalam sistem *broadcast*. Penggunaan Python dengan Flask mampu meningkatkan efektivitas pengembangan aplikasi berbasis Web, termasuk dalam mengatur komunikasi data antara *server* dan klien secara terstruktur dan aman [28].

b. NODE JS



Gambar 2.9. NODE.JS, JavaScript *Server-Side*

NODE.js merupakan *cross-platform runtime environment* yang memungkinkan eksekusi kode JavaScript di sisi *server*. Sejak diperkenalkan pada tahun 2009, NODE.js menggunakan V8, yaitu mesin JavaScript yang juga digunakan oleh Google Chrome. NODE.js juga sering digunakan sebagai bahasa pemrograman dalam pengembangan sistem *broadcast* berbasis Web. NODE.js merupakan lingkungan eksekusi JavaScript di sisi *server* yang menggunakan model event-driven dan non-blocking I/O, sehingga mampu menangani ribuan koneksi secara bersamaan tanpa menurunkan kinerja *server*. Kemampuan ini menjadikannya ideal untuk sistem *broadcast*, di mana pesan harus dikirim ke banyak penerima dalam waktu singkat. NODE.js umumnya dipadukan dengan *framework* seperti Express.js untuk membangun REST API yang menghubungkan *server* dengan layanan eksternal, misalnya WhatsApp Business API. Arsitektur *event-driven* dan *non-blocking I/O* pada NODE.js mampu menjaga stabilitas dan

performa tinggi saat menangani banyak koneksi sekaligus, yang sangat relevan untuk aplikasi *broadcast* berskala besar [29].

d. PHP (*Hypertext Preprocessor*)



Gambar 2.10. PHP (*Hypertext Preprocessor*)

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman berbasis skrip untuk Web *Server-Side*, yang digunakan untuk membuat dokumen HTML secara dinamis (*on the fly*). Artinya, dokumen HTML yang dihasilkan berasal dari suatu aplikasi, bukan dibuat langsung menggunakan editor teks atau editor HTML. Awalnya, PHP dikenal dengan nama PHP/FI, di mana PHP berarti *Personal Home Page* dan FI adalah *Form Interface*. Bahasa ini pertama kali dikembangkan oleh Rasmus Lerdorf, dan awalnya difokuskan untuk memproses *Input* yang dikirim melalui *form* yang ditampilkan di *browser* Web [30].

2.2.4. Keunggulan dan Tantangan Sistem Web *Broadcast*

Keunggulan utama dari pengembangan sistem Web *broadcast* terletak pada kemampuannya untuk mengotomatisasi komunikasi secara massal. Sistem ini memungkinkan pengiriman pesan kepada banyak penerima secara bersamaan dalam waktu yang jauh lebih singkat dibandingkan dengan metode manual. Selain itu, sistem Web *broadcast* dapat menyediakan laporan pengiriman secara

langsung dan mendukung personalisasi pesan berdasarkan data penerima. Ini membantu perusahaan dalam mempertahankan hubungan dengan pelanggan dengan cara yang lebih efektif [19].

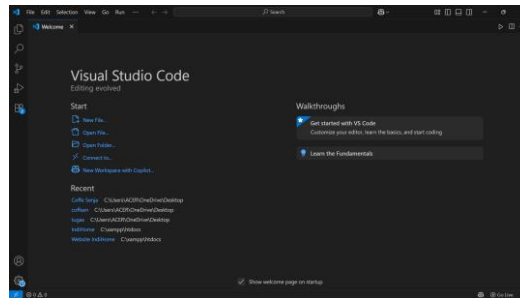
Namun, perancangan sistem *broadcast* juga menghadapi beberapa tantangan teknis. Salah satunya adalah batasan pengiriman pesan yang ditetapkan oleh platform seperti WhatsApp untuk mencegah spam, serta kebutuhan akan *server* dengan performa tinggi untuk menangani beban pengiriman secara massal. Selain itu, masalah keamanan data pelanggan menjadi perhatian utama, terutama dalam hal penyimpanan dan pengelolaan informasi pribadi yang diatur oleh kebijakan perlindungan data digital [21].

2.3. Tools Bahasa Pemrograman

Tools dalam pemrograman merupakan berbagai perangkat lunak yang membantu programmer dalam menjalankan setiap tahapan pembuatan aplikasi berbasis Web. Peralatan ini mencakup editor kode seperti Visual Studio Code, yang menyediakan fitur penyorot sintaks, sistem ekstensi, dan dukungan *Debugging* untuk mempermudah penulisan dan pengecekan program. Selain itu, *tools* seperti XAMPP juga digunakan untuk menyiapkan *server* lokal lengkap dengan Apache, MySQL, dan PHP, sehingga pengembang dapat menguji proyek secara *real-time* sebelum diterapkan ke *server* sesungguhnya. Kehadiran *tools* ini membuat proses penyusunan skrip menjadi lebih efisien, mengurangi kesalahan teknis, dan membantu pengembang menjaga kerapihan struktur proyek. Dengan memanfaatkan editor, *server* lokal, dan fitur pendukung lainnya, hasil cenderung

lebih stabil, mudah dikelola, dan cepat selesai karena alur kerja menjadi terarah dan terorganisir dengan baik [31].

2.3.1. Visual Studio Code (Vscode)



Gambar 2.11. Visual Studio Code

Visual Studio Code (VS Code) merupakan editor teks yang bersifat ringan namun memiliki kemampuan tinggi, dikembangkan oleh Microsoft dan dapat digunakan pada berbagai sistem operasi seperti Linux, macOS, serta Windows. Teks editor ini menyediakan dukungan bawaan untuk JavaScript, TypeScript, dan NODE.js, sementara bahasa pemrograman lain misalnya C++, C#, Python, Go, atau Java dapat ditambahkan melalui ekstensi yang tersedia di *marketplace* Visual Studio Code. VS Code pertama kali diperkenalkan pada 29 April 2015 dalam acara Build 2015. Versi pratinjaunya dirilis tidak lama setelah pengumuman tersebut. Pada 18 November 2015, perangkat lunak ini resmi dipublikasikan dengan Lisensi Expat, dan kode sumbernya disediakan melalui GitHub bersama pengumuman mengenai dukungan untuk ekstensi. Selanjutnya, pada 14 April 2016, Visual Studio Code berhasil melewati fase pratinjau publik dan diluncurkan secara resmi di situs Web [32]. Nah berikut fitur yang dimiliki Visual Studio Code (VSCODE) dan menjadi alasan mengapa teks editor ini sangat populer di kalangan pengembang. Beberapa fitur utamanya dapat dijelaskan sebagai berikut:

Tabel 2.1. Fitur-fitur Visual Studio Code

| No | Fitur | Deskripsi |
|----|------------------------------|--|
| 1 | <i>Basic Editing</i> | Fitur dasar pada Visual Studio Code menyediakan beragam perlengkapan untuk mengelola dan menyunting kode. Dalam kategori ini, pengguna dapat memanfaatkan fungsi seperti <i>Multiple Selection</i> , <i>Column Selection</i> , dan beragam pintasan keyboard. Selain itu, VSCODE juga mendukung <i>Auto Save</i> dan <i>Hot Exit</i> yang membuat pekerjaan tidak mudah hilang dan dapat dilanjutkan dengan cepat. |
| 2 | <i>Debugging</i> | Fitur penting lain yang sering menjadi alasan developer memilih VSCODE adalah kemampuan <i>Debugging</i> . Melalui fitur ini, proses memperbaiki, menjalankan, dan melakukan kompilasi ulang kode menjadi jauh lebih praktis, terutama ketika harus mengerjakan bagian program yang membutuhkan pengulangan atau <i>looping</i> . |
| 3 | <i>Extension Marketplace</i> | Salah satu kelebihan utama Visual Studio Code adalah tersedianya <i>Extension Marketplace</i> , yang tidak semua editor kode miliki. Melalui fitur ini, pengguna dapat menambahkan ekstensi atau plugin sesuai kebutuhan, termasuk dukungan tambahan untuk berbagai bahasa pemrograman dan alat pendukung lainnya |
| 4 | <i>IntelliSense</i> | Seperti halnya fitur Autocomplete atau Autocorrect pada ponsel, VSCODE juga memiliki kemampuan serupa yang disebut <i>IntelliSense</i> . Fitur ini secara otomatis Memberikan rekomendasi kata, fungsi, atau perintah berdasarkan kode yang sedang diketik pengguna, sehingga penulisan program menjadi |

| No | Fitur | Deskripsi |
|----|---------------------------|--|
| | | lebih cepat dan minim kesalahan. |
| 5 | GitHub <i>Integration</i> | Bagi programmer, GitHub tentu bukan hal baru. Platform tersebut sangat populer sebagai tempat pengelolaan dan kolaborasi proyek pengembangan perangkat lunak. VSCODE mempermudah integrasi dengan GitHub; pengguna hanya perlu menambahkan ekstensi GitHub <i>Pull Request</i> dan <i>Issues</i> untuk langsung mengelola repositori dari dalam editor |

Visual Studio Code menjadi salah satu editor kode yang paling banyak digunakan karena menyediakan fitur lengkap yang mendukung kenyamanan dan produktivitas pengembang [33]. Selain itu vscode juga memiliki kelebihan dan kekurangan, nah dapat dijelaskan dalam tabel berikut:

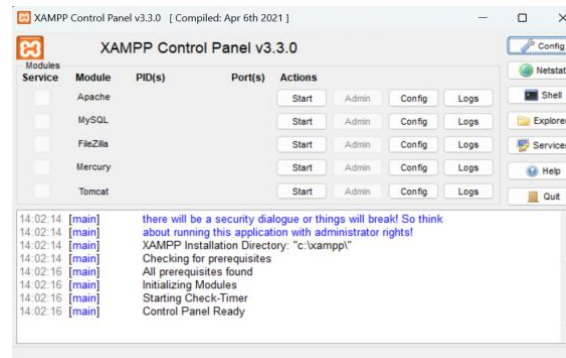
Tabel 2.2. Kelebihan & Kekurangan Visual Studio Code

| No | Kelebihan Visual Studio Code | Kekurangan Visual Studio Code |
|----|--|---|
| 1 | Gratis: Visual Studio Code unggul dari segi kelengkapan fitur. Hal ini dikarenakan terdapat banyak <i>Extension Marketplace</i> yang memungkinkan pengguna lebih leluasa menambahkan fitur tertentu dan semuanya bisa diakses secara gratis. | Ketergantungan pada Ekstensi: Pengguna sering harus mencari ekstensi yang tepat, dan hal ini dapat membingungkan atau menimbulkan konflik antar ekstensi. |
| 2 | Ringan: Meski memiliki banyak fitur, <i>Software</i> ini tetap ringan dan tidak membutuhkan perangkat berspesifikasi tinggi. | Kustomisasi yang Memerlukan Waktu: Pengaturan VS Code sesuai kebutuhan bisa memakan waktu, terutama bagi pengguna baru. |

| No | Kelebihan Visual Studio Code | Kekurangan Visual Studio Code |
|----|---|--|
| 3 | Mendukung banyak Bahasa pemrograman: VS Code dapat digunakan untuk berbagai bahasa berkat dukungan <i>Extension Marketplace</i> . | Performa pada proyek besar: Kinerja dapat melambat saat menangani proyek besar atau basis kode yang kompleks, terlebih jika terlalu banyak ekstensi terinstal. |
| 4 | Performa cepat: Mampu menangani berbagai tugas seperti pembuatan aplikasi Android, Website, dan iOS dengan cepat. | Bisa membingungkan bagi pemula: Banyaknya fitur dan opsi penyesuaian dapat terasa terlalu rumit untuk pengguna baru. |

Visual Studio Code (VS Code) adalah editor kode yang ringan namun sangat mumpuni, mendukung beragam bahasa pemrograman, dan dapat dijalankan pada sistem operasi Windows, Linux, maupun macOS. Berkat fitur unggulannya seperti pengeditan dasar, kemampuan *Debugging*, *IntelliSense*, integrasi dengan GitHub, serta koleksi ekstensi yang luas, VS Code mampu meningkatkan efisiensi dan produktivitas para pengembang. Meski demikian, performanya kadang bergantung pada ekstensi tambahan, dapat melambat ketika digunakan pada proyek berskala besar, dan banyaknya opsi konfigurasi bisa membingungkan bagi pengguna baru. Secara keseluruhan, kombinasi antara fitur lengkap, performa yang stabil, serta akses gratis menjadikan VS Code sebagai salah satu editor kode yang paling populer saat ini [34].

2.3.2. XAMPP



Gambar 2.12. XAMPP Control Panel

XAMPP yaitu perangkat lunak gratis (*free Software*) yang kompatibel dengan berbagai sistem operasi [35]. *Software* ini adalah gabungan dari beberapa program, yaitu:

1. X (Cross Platform), Kode ini berfungsi sebagai penanda bahwa suatu perangkat lunak dapat dijalankan di berbagai sistem operasi, seperti Windows, Linux, macOS, hingga Solaris.
2. A (Apache), Hal ini menunjukkan bahwa Web *server* yang digunakan bersifat gratis dan terbuka untuk dikembangkan oleh komunitas (*open source*), sehingga siapa pun dapat berkontribusi dalam membuat halaman Web yang berfungsi dengan benar.
3. M (MySQL/MariaDB), MySQL adalah perangkat lunak *server database* yang memanfaatkan bahasa *Structured Query Language* (SQL) untuk mengelola data dengan lebih rapi dan sistematis. Contoh penggunaannya adalah melakukan pemrosesan data, memperbaiki informasi, atau menghapus entri di dalam *database*. MySQL dapat dijalankan secara lokal tanpa memerlukan koneksi internet, sehingga pengembang dapat membangun aplikasi Web langsung di komputer mereka. Sementara itu,

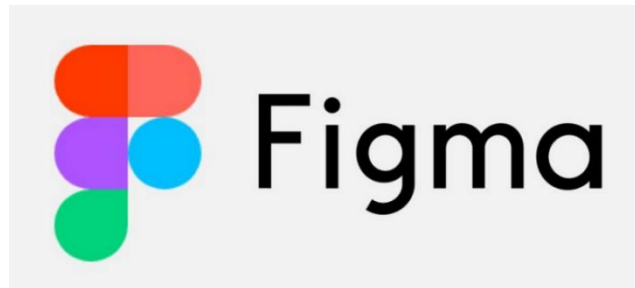
MariaDB merupakan sistem manajemen basis data yang dikembangkan secara mandiri sebagai kelanjutan dari MySQL.

4. P (PHP), merupakan bahasa pemrograman yang dirancang untuk kebutuhan pengolahan data di sisi *server* atau *back-end*, sehingga mampu membuat halaman Web menjadi interaktif melalui teknik *Server-Side* scripting. PHP juga mendukung berbagai sistem manajemen *database*, seperti Oracle, PostgreSQL, Microsoft *Access*, dan sejumlah platform lainnya.
5. P (Perl), adalah bahasa pemrograman serbaguna yang dapat digunakan untuk berbagai kebutuhan dan dapat berjalan pada banyak sistem operasi, sehingga menjadikannya sangat fleksibel dan populer. Bahasa ini juga dianggap sebagai pendahulu yang menunjukkan jejak perkembangan PHP. Saat ini, Perl banyak dimanfaatkan dalam pembuatan aplikasi, pengelolaan Web *server*, serta pengembangan situs berbasis *Content Management System* (CMS) seperti WordPress.

Dari beberapa program di atas jika disatukan membentuk nama XAMPP. Perangkat lunak ini bisa digunakan secara bebas tanpa harus membeli lisensi dari pembuatnya. XAMPP berfungsi sebagai localhost, yaitu *server* mandiri, yang memungkinkan pengguna untuk menguji fitur-fitur maupun menampilkan konten Website kepada orang lain tanpa memerlukan koneksi internet [36].

2.3.3. Figma

Figma merupakan salah satu perangkat desain prototipe UI/UX yang menyediakan berbagai fitur kolaboratif serta fleksibilitas tinggi sehingga banyak digunakan dalam proses perancangan Website modern [37].



Gambar 2.13. Figma

Sebelum pengembangan Website dilakukan, diperlukan tahap perancangan desain untuk menyesuaikan tampilan, fitur, dan fungsi yang akan digunakan. Perancangan ini dibuat berdasarkan kebutuhan pengguna agar dapat mempermudah mereka dalam mencapai tujuan penggunaan sistem. Proses tersebut dikenal dengan konsep UI/UX. *User Interface* (UI) berfokus pada tampilan visual yang menghubungkan sistem dengan pengguna, sedangkan *User Experience* (UX) berkaitan dengan pengalaman dan kenyamanan pengguna saat berinteraksi dengan aplikasi. Dalam proses desain, Figma sering digunakan karena menyediakan fitur prototyping yang memungkinkan desainer membuat simulasi tampilan aplikasi seperti penggunaan nyata. Selain itu, Figma mendukung kerja *real-time*, sehingga setiap perubahan dapat langsung diperbarui dan tersimpan secara otomatis [38].

2.4. WhatsApp Sebagai Kanal Komunikasi Bisnis

Media sosial ialah platform daring yang memungkinkan penggunanya untuk berpartisipasi, berbagi, dan menciptakan konten dengan mudah. Media sosial berfungsi sebagai sarana untuk memperkenalkan serta memasarkan produk, sekaligus memperluas interaksi sosial antar individu. Media sosial adalah media untuk berbagi informasi dalam bentuk gambar, video, teks, maupun audio kepada konsumen, sehingga memungkinkan terjadinya interaksi yang efektif antar pengguna. Secara keseluruhan, media sosial menghubungkan orang-orang di seluruh dunia dan memfasilitasi komunikasi serta hubungan sosial antarindividu [39].

Media sosial juga salah satu elemen penting dalam komunikasi bisnis di era digital. Peranannya sangat signifikan, mulai dari membangun citra merek hingga meningkatkan penjualan. Dalam konteks membangun dan meningkatkan kesadaran merek (*brand awareness*), media sosial memungkinkan perusahaan menjangkau calon pelanggan secara lebih luas dan menciptakan kesadaran merk secara efektif. Dengan menghadirkan konten yang menarik dan relevan, bisnis dapat menarik perhatian calon konsumen sekaligus memperkuat citra merek. Platform seperti WhatsApp, Instagram, TikTok, Shopee, dan Tokopedia sangat berperan dalam memperkuat visual merek serta menciptakan identitas yang konsisten. Selain itu, media sosial juga berfungsi untuk membangun hubungan dengan pelanggan (*customer engagement*). Media ini memungkinkan interaksi langsung, menciptakan komunikasi yang lebih personal, dan meningkatkan loyalitas konsumen. Pelaku usaha dapat merespons pertanyaan, menerima masukan, serta membangun kepercayaan pelanggan melalui interaksi positif di

media sosial. Media sosial juga menyediakan data berharga mengenai tren pasar, preferensi pelanggan, dan perilaku konsumen. Informasi ini dapat digunakan pengusaha untuk memahami pasar lebih baik, menyusun strategi pemasaran yang tepat, serta mengambil keputusan bisnis yang lebih akurat. Analisis sentimen di media sosial membantu perusahaan memahami persepsi publik terhadap merek dan produk, sementara alat analisis media sosial memungkinkan pelacakan kinerja kampanye serta evaluasi dampak komunikasi bisnis digital, dengan fokus pada interaksi pengguna dan konten yang disebarluaskan [40].



Gambar 2.14. WhatsApp Broadcast

WhatsApp adalah salah satu platform media sosial berbasis pesan yang digunakan untuk berkomunikasi, mulai dari bertukar *chat*, mengirim dokumen, foto, melakukan panggilan suara hingga *video call* melalui jaringan internet. Aplikasi ini masuk dalam kategori media sosial paling aktif, menempati posisi kedua setelah YouTube. Dalam laporan tersebut, YouTube digunakan oleh sekitar 88% pengguna internet di Indonesia, kemudian disusul oleh WhatsApp dengan 83%, dan Facebook berada di posisi ketiga dengan 80% pengguna. WhatsApp memiliki komunitas pengguna yang sangat luas dan saling terhubung, sehingga menarik perhatian Facebook untuk mengakuisisinya. Facebook akhirnya membeli aplikasi ini dengan nilai sekitar 19 miliar dolar AS karena ingin melengkapi layanannya yang sebelumnya belum memiliki basis data komunikasi berbasis

pesan seluler dan nomor telepon [41].

WhatsApp resmi diluncurkan pada tahun 2009 oleh mantan karyawan Yahoo!, Jan Koum dan Brian Acton. Awalnya hanya tersedia untuk pengguna iPhone, kemudian berkembang ke platform Android, Blackberry, dan Windows Phone. Aplikasi ini muncul sebagai alternatif pesan SMS yang mahal, dan dengan cepat menjadi salah satu aplikasi dengan jumlah pengguna terbanyak di dunia. Perkembangan WhatsApp telah mengubah komunikasi interpersonal di lingkungan kerja. Aplikasi ini memungkinkan karyawan berbagi informasi dengan cepat, berkoordinasi secara real time, dan mengatasi kendala jarak atau lokasi cabang yang tersebar. Grup obrolan WhatsApp mendukung kolaborasi antar tim atau departemen, memperkuat hubungan antar rekan kerja, serta memfasilitasi komunikasi yang lebih efektif dan kolaboratif [42].

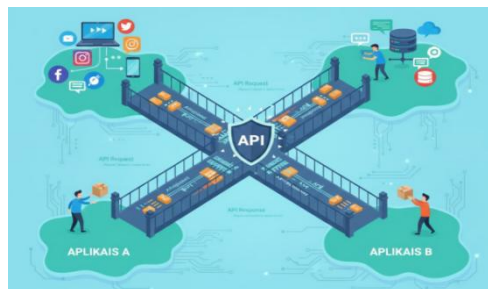
Beberapa fitur WhatsApp yang mendukung komunikasi dan kolaborasi di tempat kerja antara lain:

1. Obrolan Grup (*Group Chat*): Memudahkan koordinasi tim, berbagi informasi, dan pertukaran ide secara cepat.
2. Status: Memungkinkan karyawan membagikan pembaruan singkat berupa teks, foto, atau video terkait pekerjaan, sehingga menciptakan transparansi dan meningkatkan keterlibatan tim. Konten status otomatis hilang dalam 24 jam.
3. Panggilan Video (*Video call*): Memfasilitasi komunikasi visual secara gratis, mendukung pertemuan virtual, serta meningkatkan efisiensi kolaborasi.

4. Pesan Suara (*Voice Note*): Alternatif komunikasi selain teks yang memungkinkan pengiriman pesan secara rinci dan mendalam; pengguna bisa merekam, menjeda, dan memutar kembali pesan suara.
5. Berbagi *File* dan Dokumen: Mempermudah pertukaran dokumen, laporan, presentasi, dan foto dalam berbagai format untuk mendukung kerja sama tim.
6. *Polling*: Digunakan untuk pengambilan keputusan cepat melalui voting, sehingga setiap anggota dapat Memberikan suara yang memengaruhi hasil Keputusan.

Dengan fitur-fitur tersebut, WhatsApp menjadi alat komunikasi yang efektif, meningkatkan interaksi interpersonal, efisiensi kerja, dan kolaborasi tim di lingkungan kerja [42].

2.5. Integrasi API WhatsApp



Gambar 2.15. Jembatan Penghubung Antar Aplikasi

API merupakan singkatan dari *Application Programming Interface*. Secara sederhana, API dapat dipahami sebagai sekumpulan aturan serta prosedur yang memungkinkan berbagai aplikasi perangkat lunak untuk saling berkomunikasi. Menurut Ferry Gunawan, Grace Martha, dan G. Bororing, API adalah komponen dalam sebuah sistem perangkat lunak yang berisi himpunan fungsi, instruksi, serta

protokol yang menyediakan mekanisme agar sistem komputer dapat berinteraksi satu sama lain. Ketika Anda menjalankan suatu aplikasi di ponsel maupun komputer, aplikasi tersebut akan berhubungan dengan *server* melalui API. Aplikasi mengirimkan permintaan (*request*) ke *server*, kemudian *server* memberikan kembali data yang dibutuhkan. Contohnya, saat Anda melakukan pencarian di Google, aplikasi secara otomatis mengirimkan *request* ke *server* Google lewat API, lalu *server* mengembalikan hasil pencarian yang Anda butuhkan [43]. Dari segi arsitektur adapun jenis-jenis Api, yaitu sebagai berikut:

1. *RESTful* API bekerja dengan menerapkan konsep REST (*Representational State Transfer*), yaitu pendekatan di mana setiap permintaan yang dikirim ke API mewakili suatu tindakan tertentu. Contohnya, permintaan *GET* digunakan untuk memperoleh data, *POST* untuk menambahkan data baru, *PUT* untuk mengubah data yang sudah ada, dan *DELETE* untuk menghapus data.
2. Sementara itu, GraphQL (*Graph Query Language*) merupakan jenis API yang Memberi keleluasaan bagi pengguna untuk meminta data secara lebih spesifik sesuai kebutuhan, sehingga informasi yang diterima dapat lebih tepat dan efisien.

Integrasi WhatsApp API yaitu aplikasi atau sistem pihak ketiga untuk mengirim dan menerima pesan secara otomatis, sehingga komunikasi dengan pelanggan dapat berlangsung *real-time* tanpa intervensi manual. Sistem ini dapat digunakan untuk berbagai keperluan, seperti notifikasi transaksi, pengingat pembayaran, atau layanan pelanggan yang dijalankan secara otomatis. Selain itu, API ini mendukung pengembangan *chatbot*, yang mampu merespons pertanyaan

pelanggan secara instan dan konsisten, sehingga meningkatkan efisiensi layanan dan menjaga kepuasan pelanggan melalui respons yang cepat dan akurat, penerapan WhatsApp API dapat meningkatkan efektivitas komunikasi dibandingkan dengan metode manual, menunjukkan manfaat signifikan dalam pengelolaan interaksi bisnis digital [44].

Selain itu, integrasi WhatsApp API memungkinkan perusahaan untuk mengirim pesan massal yang disesuaikan dengan preferensi pelanggan. Dengan memanfaatkan data pelanggan, pesan yang dikirim dapat lebih relevan, seperti promosi produk yang sesuai minat pengguna, sehingga interaksi dengan pelanggan meningkat dan peluang konversi penjualan menjadi lebih tinggi. integrasi API mendukung otomatisasi layanan pelanggan, mempercepat respons, mengurangi beban staf, dan memantau efektivitas komunikasi secara *real-time* [45].

2.5.1. Cara Kerja Api



Gambar 2.16. Alur Kerja API (*Application Programming Interface*)

API menjalankan fungsinya dengan menerima suatu permintaan (*request*) dari sebuah sistem, kemudian mengirimkan kembali balasan (*response*) kepada sistem tersebut. Dalam prosesnya, API memanfaatkan protokol HTTP sebagai media pertukaran informasi sehingga data dapat dikirimkan melalui jaringan

internet. Bentuk data yang umum digunakan dalam proses komunikasi API adalah JavaScript Object Notation (JSON), karena format ini mudah dipahami oleh manusia maupun komputer, serta memiliki ukuran yang ringan sehingga efisien saat digunakan dalam pertukaran data.

2.5.2. Manfaat Penggunaan Api

Terdapat berbagai keuntungan yang bisa *diperoleh* ketika menggunakan API. Beberapa manfaat utamanya antara lain:

1. Integrasi Aplikasi API memungkinkan berbagai platform atau sistem untuk saling terhubung dan bertukar data. Dengan begitu, proses integrasi dapat dilakukan tanpa harus membangun ulang fitur yang sudah tersedia sebelumnya.
2. Efisiensi dalam Pengembangan Pemanfaatan API membantu pengembang karena tidak perlu membuat seluruh komponen dari nol. Fitur yang sudah disediakan API bisa langsung digunakan, sehingga waktu dan usaha dalam proses pembuatan aplikasi menjadi lebih singkat.
3. Kolaborasi Lintas Platform Dengan API, data atau layanan dapat diakses melalui berbagai perangkat seperti Website, aplikasi *mobile*, maupun *desktop* secara bersamaan. Ini memastikan setiap pengguna memperoleh informasi yang sama di semua platform.

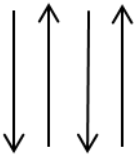


Integrasi API, khususnya WhatsApp API, Memberikan peran penting dalam meningkatkan efektivitas komunikasi digital. Penerapan WhatsApp API memungkinkan otomatisasi komunikasi dan promosi dengan pelanggan, termasuk pengiriman pesan *real-time*, notifikasi, pengingat, layanan pelanggan berbasis



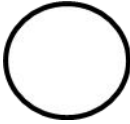
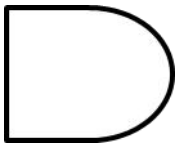



chatbot, hingga pengiriman pesan massal yang dipersonalisasi. Hal ini tidak hanya mempercepat proses komunikasi, tetapi juga meningkatkan efisiensi operasional, relevansi pesan, serta peluang konversi bisnis [46].

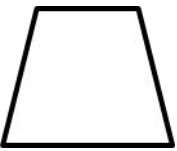

2.6. Flowchart

Flowchart merupakan bentuk penyajian algoritma yang dituliskan menggunakan simbol-simbol grafis. Diagram ini menampilkan alur kerja atau urutan proses suatu program, termasuk langkah-langkah, keputusan, serta hubungan antar proses. Penggunaan *Flowchart* membantu memeriksa apakah ada bagian analisis yang terlewat saat memecahkan suatu permasalahan. Selain itu, *Flowchart* juga memudahkan analisis maupun programmer dalam memecah persoalan menjadi bagian-bagian kecil serta mempertimbangkan berbagai alternatif proses yang dapat dilakukan [47].

Tabel 2.3. Simbol-simbol *Flowchart*

| NO | SIMBOL | DESKRIPSI |
|----|---|---|
| 1 |  | Simbol Arah Alur (<i>Flow Direction</i>) Digunakan untuk menghubungkan simbol satu dengan yang lain serta menunjukkan arah jalannya proses. |
| 2 |  | Terminal Digunakan sebagai titik awal dan akhir dari sebuah program. |
| 3 |  | <i>Input-Output</i> Menyatakan proses menerima masukan data atau menampilkan keluaran dari suatu kegiatan. |

| NO | SIMBOL | DESKRIPSI |
|----|---|--|
| 4 |  | <p>Keputusan (<i>Decision</i>) Menandakan adanya kondisi tertentu yang memerlukan pilihan atau cabang dari beberapa kemungkinan.</p> |
| 5 |  | <p>Proses yang Telah Ditentukan (<i>Predefined Process</i>) Simbol yang menggambarkan proses yang sudah didefinisikan sebelumnya, biasanya berkaitan dengan penyimpanan atau pemrosesan data.</p> |
| 6 |  | <p>Penghubung (<i>Connector</i>) Digunakan untuk menyambungkan proses yang berpindah halaman atau bagian.</p> |
| 7 |  | <p>Tunda (<i>Delay</i>) Menunjukkan adanya jeda waktu atau proses menunggu, seperti menanti dokumen disortir atau diproses.</p> |
| 8 |  | <p>Dokumen Ganda (<i>Multiple Documents</i>) Menyatakan lebih dari satu dokumen yang terlibat dalam satu proses, dan dilambangkan dalam satu simbol.</p> |
| 9 |  | <p>Dokumen (<i>Document</i>) Simbol ini digunakan untuk menggambarkan data berbentuk informasi tertulis atau tercetak.</p> |
| 10 |  | <p>Proses Tertentu (<i>Predefined Process</i>) Menandakan bahwa langkah-langkah atau proses yang dilakukan sudah ditentukan dalam bentuk prosedur.</p> |

| NO | SIMBOL | DESKRIPSI |
|----|---|---|
| 11 |  | Operasi Manual (<i>Manual Operation</i>) Melambangkan proses atau kegiatan yang dilakukan secara manual, tanpa bantuan alat otomatis. |
| 12 |  | Proses (<i>Process</i>) Menunjukkan adanya aktivitas komputasi atau pengolahan data yang dilakukan oleh sistem/program. |

Flowchart memiliki peran yang sangat krusial dalam penelitian “Rancang Bangun Web *Broadcast* WhatsApp pada PT. Infomedia Nusantara Terintegrasi API” karena berfungsi sebagai representasi visual dari alur kerja sistem yang dikembangkan. Pada proses pembuatan Web *broadcast* yang berintegrasi dengan WhatsApp API, terdapat tahapan-tahapan seperti memasukkan data pelanggan, memilih *template* pesan, mengirim permintaan ke API, hingga menerima respons dari *server*. Seluruh proses tersebut perlu digambarkan secara runtut dan jelas agar mudah dipahami, dan *Flowchart* menjadi media yang paling sesuai untuk menunjukkan urutan langkah serta hubungan antar proses dalam sistem tersebut.

2.7. UML (*Unified Modelling Language*)


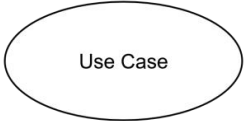
Unified Modeling Language (UML) merupakan sebuah “bahasa” standar yang digunakan dalam industri untuk memvisualisasikan, merancang, serta mendokumentasikan sistem perangkat lunak. UML menyediakan pendekatan baku dalam pembuatan model suatu sistem. Melalui UML, kita dapat memodelkan berbagai jenis aplikasi perangkat lunak yang dapat dijalankan pada beragam perangkat keras, sistem operasi, jaringan, maupun bahasa pemrograman apa pun.


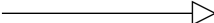
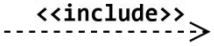

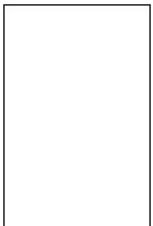
UML pertama kali dikembangkan oleh *Object Management Group*, dengan versi perdana 1.0 yang dirilis pada Januari 1997 [48].

2.7.1. Use Case Diagram

Use Case adalah gambaran fungsi sistem yang dilihat dari sudut pandang pengguna. *Use Case* menjelaskan proses yang dijalankan oleh sistem beserta komponen-komponen yang terlibat di dalamnya. Dalam penerapannya, *Use Case* disusun berdasarkan skenario yang berisi urutan aktivitas atau tahapan interaksi antara pengguna dengan sistem, maupun respons sistem terhadap tindakan pengguna. *Use Case* menggambarkan suatu tugas tertentu, seperti masuk ke sistem, membuat daftar dan aktivitas lainnya. Aktor merupakan entitas baik manusia maupun perangkat yang berhubungan langsung dengan sistem untuk menjalankan tugas-tugas tersebut[49].

Tabel 2.4. Simbol UML *Use Case* Diagram



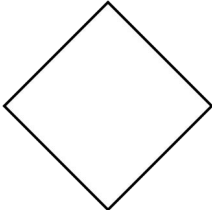
| NO | SIMBOL | DESKRIPSI |
|----|---|--|
| 1 |  | <p>Actor</p> <p>Entitas, Atau sistem lain yang berinteraksi dengan sistem informasi yang sedang dikembangkan, terletak di luar sistem itu sendiri. Meskipun simbol untuk <i>actor</i> seringkali berupa gambar orang, biasanya nama actor dinyatakan dengan kata benda di awal frasa.</p> |
| 2 |  | <p>Use Case</p> <p>Serangkaian Fungsionalitas Yang Disediakan Oleh Sistem, dibagi menjadi unit-unit yang saling bertukar pesan antara unit tersebut dan actor. Nama <i>Use Case</i> umumnya dimulai dengan kata kerja.</p> |

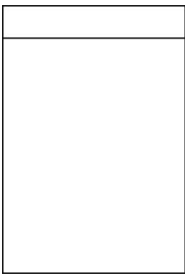

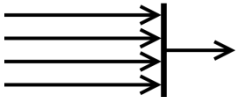

| NO | SIMBOL | DESKRIPSI |
|----|---|--|
| 3 |  | <p>Association</p> <p>Interaksi antara actor dan <i>Use Case</i> yang terlibat dalam <i>Use Case</i> tersebut—ini menunjukkan bahwa <i>Use Case</i> berhubungan dan berkomunikasi dengan actor.</p> |
| 4 |  | <p>Generalisasi</p> <p>Hubungan antara dua <i>Use Case</i> yang mencerminkan konsep generalisasi dan spesialisasi (umum dan khusus), di mana satu fungsi merupakan versi yang lebih umum dari yang lainnya.</p> |
| 5 |  | <p>Menggunakan (<i>Include</i>)</p> <p>Hubungan <i>Use Case</i> tambahan ke dalam sebuah <i>Use Case</i>, di mana <i>Use Case</i> yang ditambahkan diperlukan untuk menjalankan fungsionalitas atau sebagai syarat untuk melaksanakan <i>Use Case</i> tersebut.</p> |
| 6 |  | <p>Ekstensi (<i>Extend</i>)</p> <p>Hubungan <i>Use Case</i> tambahan terhadap sebuah <i>Use Case</i>, di mana <i>Use Case</i> yang ditambahkan bisa berdiri sendiri meskipun tanpa keberadaan <i>Use Case</i> tambahan tersebut. Nama depan dari <i>Use Case</i> yang ditambahkan akan sama dengan <i>Use Case</i> utama.</p> |
| 7 |  | <p>Sistem (<i>System Boundary</i>)</p> <p>Persegi panjang besar yang mengelilingi <i>Use Case</i> untuk mendefinisikan batas sistem.</p> |

2.7.2. Activity Diagram

Activity diagram digunakan untuk memvisualisasikan alur aktivitas dalam sistem yang sedang dikembangkan, termasuk bagaimana setiap alur dimulai, kemungkinan keputusan yang muncul, serta bagaimana alur tersebut berakhir. Diagram ini juga mampu menunjukkan proses-proses yang berjalan secara paralel pada beberapa eksekusi. Secara teknis, *Activity* diagram adalah jenis state diagram khusus, di mana sebagian besar state merepresentasikan tindakan (*action*), dan sebagian besar transisi dipicu oleh selesainya state sebelumnya (*internal processing*). Oleh karena itu, *Activity* diagram tidak menampilkan perilaku internal sistem atau interaksi antar subsistem secara rinci, melainkan lebih menekankan pada proses dan jalur aktivitas secara umum dari tingkat atas[50].

Tabel 2.5. Simbol UML *Activity* Diagram

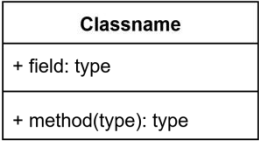


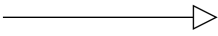


| NO | SIMBOL | DESKRIPSI |
|----|---|---|
| 1 |  | Start Status awal aktivitas sistem, diletakkan pada pojok kiri atas. |
| 2 |  | Aktivitas(Activity) Persegi panjang dengan sudut membulat. Menyatakan suatu kegiatan, proses, atau langkah yang terjadi dalam alur kerja. |
| 3 |  | Decision (Keputusan) Menunjukkan titik percabangan di mana alur kerja dapat mengikuti salah satu dari beberapa jalur berdasarkan suatu kondisi. |

| NO | SIMBOL | DESKRIPSI |
|----|---|---|
| 4 |  | Swimlane Pembagian <i>Activity</i> diagram untuk menunjukkan siapa melakukan apa |
| 5 |  | Fork NODE Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran yang di eksekusi secara bersamaan |
| 6 |  | Penggabungan (Join) Asosiasi yang menunjukkan penggabungan, di mana lebih dari satu aktivitas berkumpul menjadi satu. |
| 7 |  | End Lingkaran padat dengan lingkaran lain di luarnya. Menandakan akhir dari alur kerja. |

2.7.3. Class Diagram

Class merupakan spesifikasi yang, ketika diinstansiasi, akan membentuk sebuah objek, dan menjadi komponen utama dalam pengembangan serta desain berorientasi objek. *Class* mendeskripsikan kondisi atau atribut suatu sistem sekaligus menyediakan layanan atau metode untuk memanipulasi kondisi tersebut. *Class* diagram digunakan untuk menggambarkan struktur serta deskripsi dari *Class*, halaman, dan objek, termasuk hubungan di antara mereka, seperti *containment*, pewarisan, asosiasi, dan lainnya[51].

Tabel 2.6. Simbol UML *Class* Diagram

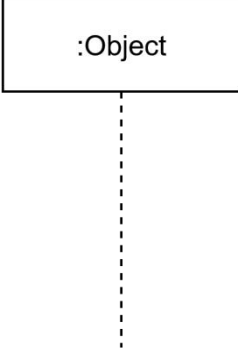

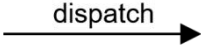
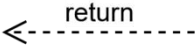
| NO | SIMBOL | DESKRIPSI |
|----|---|--|
| 1 |  | <p>Class</p> <p>Menampilkan atribut dan operasi dari sebuah objek.</p> |
| 2 |  | <p>Aggregation</p> <p>Hubungan “bagian dari” yang lemah (bagian yang bisa berdiri sendiri).</p> |
| 3 |  | <p>Composition</p> <p>Hubungan “bagian dari” yang kuat (bagian yang tidak bisa terpisah).</p> |
| 4 |  | <p>Generalisasi</p> <p>Relasi turunan (<i>Class</i> anak mewarisi <i>Class</i> induk)</p> |
| 5 |  | <p>Dependency</p> <p>Ketergantungan antar kelas (menggunakan jasa <i>Class</i> lain).</p> |
| 6 |  | <p>Association</p> <p>Hubungan umum antara dua kelas.</p> |

2.7.4. Sequence Diagram

Sequence diagram digunakan untuk memvisualisasikan interaksi antar objek di dalam dan sekitar sistem, termasuk pengguna, tampilan, dan komponen lainnya, dalam bentuk pesan yang ditampilkan terhadap garis waktu. Diagram ini memiliki dua dimensi: vertikal yang mewakili waktu, dan horizontal yang mewakili objek-objek yang terlibat. *Sequence* diagram biasanya dipakai untuk menunjukkan skenario atau urutan langkah-langkah yang terjadi sebagai respons terhadap suatu peristiwa, sehingga menghasilkan *Output* tertentu. Diagram ini

menampilkan pemicu aktivitas, proses dan perubahan internal yang terjadi, serta *Output* yang dihasilkan[52].

Tabel 2.7. Simbol UML *Sequence* Diagram


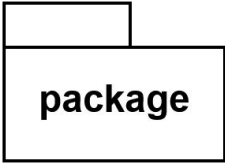



| NO | SIMBOL | DESKRIPSI |
|----|---|---|
| 1 |  | <p><i>LifeLine</i> Objek entity, antarmuka yang saling berinteraksi.</p> |
| 2 |  | <p><i>Activation Bar</i> Objek sedang menjalankan proses.</p> |
| 3 |  | <p>Message Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi. Pesan / pemanggilan metode.</p> |
| 4 |  | <p><i>Return Message</i> Mengembalikan nilai dari pemanggilan metode.</p> |

2.7.5. Component Diagram

Component diagram digunakan untuk memvisualisasikan struktur dan hubungan antar komponen perangkat lunak, termasuk ketergantungan (*Dependency*) di antara mereka. Komponen perangkat lunak merupakan modul yang berisi kode, baik berupa source code maupun binary code, termasuk *library* maupun *executable*, yang dapat muncul pada compile time, link time, maupun run

time. Biasanya, sebuah komponen terdiri dari beberapa *Class* dan/atau *PACKAGE*, namun juga bisa terbentuk dari komponen-komponen yang lebih kecil. Selain itu, komponen juga dapat berupa interface, yaitu sekumpulan layanan yang disediakan oleh satu komponen untuk digunakan oleh komponen lain[53].



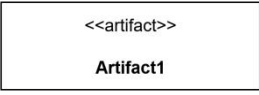

Tabel 2.8. Simbol UML *Component* Diagram

| NO | SIMBOL | DESKRIPSI |
|----|---|---|
| 1 |  | Component Bagian modular dari sistem (modul, service, library). |
| 2 |  | PACKAGE Sebuah bungkus dari satu atau lebih komponen. |
| 3 |  | Interface (Provided Interface) Layanan yang disediakan oleh komponen. |
| 4 |  | Required Interface Layanan yang dibutuhkan komponen dari pihak lain. |
| 5 |  | Dependency Ketergantungan antarkomponen. |

2.7.6. Collaboration Diagram

Collaboration diagram juga memodelkan interaksi antar objek seperti halnya *Sequence* diagram, namun fokusnya lebih pada peran masing-masing objek daripada urutan waktu pengiriman pesan. Setiap pesan diberikan nomor urut (*Sequence number*), di mana pesan pada level tertinggi diberi nomor 1, dan pesan-pesan pada level yang sama menggunakan prefiks yang serupa[54].

Tabel 2.9. Simbol UML *Collaboration* Diagram

| NO | SIMBOL | DESKRIPSI |
|----|---|---|
| 1 |  | <i>NODE</i> Menggambarkan aplikasi yang mampu mengeksekusi program. |
| 2 |  | <i>PACKAGE</i> Sebuah bungkus dari satu atau lebih komponen. |
| 3 |  | <i>Artifact</i> <i>File</i> yang di- <i>deploy</i> (misal .exe, .jar, <i>database</i>).. |
| 4 |  | <i>Communication Path</i> Jalur komunikasi antar <i>NODE</i> . |

UML memegang peranan penting dalam proses analisis dan perancangan sistem Web *Broadcast* WhatsApp yang terintegrasi dengan API. Setiap jenis diagram digunakan untuk menggambarkan kebutuhan sistem, alur kerja, struktur komponen, serta hubungan antar bagian dalam sistem. Karena sistem ini harus berinteraksi dengan WhatsApp API, UML membantu memvisualisasikan mekanisme komunikasi tersebut secara lebih jelas, sehingga memudahkan proses integrasi layanan API, pengembangan fitur *broadcast*, serta otomasi pengiriman pesan [55].

2.8. Hasil Penelitian Terdahulu

Beberapa penelitian relevan yang mendukung pengembangan sistem *broadcast* WhatsApp terintegrasi API antara lain:

1. **Analisis Fitur dan Fungsionalitas Aplikasi Whatsapp *Broadcast* untuk Penyebaran Informasi Pembayaran Pajak (Ritan, Deta, &**

Ishak, Journal of Artificial Intelligence and Digital Business (RIGGS), 2025), Kajian ini melakukan telaah mendalam terhadap efektivitas dan kapabilitas fitur WhatsApp *Broadcast* yang digunakan oleh UPTD Pendapatan Daerah Kabupaten Flores Timur untuk mendistribusikan kabar mengenai pembayaran pajak secara serentak kepada masyarakat. Dengan melibatkan survei kuantitatif terhadap 98 responden, temuan studi ini membuktikan bahwa fungsionalitas utama WhatsApp *Broadcast*, seperti pengiriman pesan massal, kemampuan mengunggah data penerima secara otomatis, dan penggunaan pesan *template*, dinilai sangat membantu petugas pelayanan pajak. Fitur-fitur tersebut memungkinkan penyampaian informasi penting berlangsung secara cepat dan personal, mencapai skor efektivitas rata-rata 3,25 pada skala Likert. Meskipun demikian, penelitian ini juga mencatat adanya tantangan teknis, yaitu masalah nomor telepon yang tidak aktif dan keterbatasan dalam melacak status pengiriman pesan. Hasil ini secara langsung mendukung pengembangan sistem *broadcast* WhatsApp API, karena menekankan bahwa fitur pengiriman massal dan otomatisasi sangat krusial dalam layanan publik atau bisnis [56].

2. **Implementasi Fitur Notifikasi Whatsapp API pada Sistem Manajemen Tugas Akhir (Anjasmara, Rosid, & Eviyanti, Physical Sciences, Life Science and Engineering, 2024)**, Penelitian ini memfokuskan pada pengembangan aplikasi Sistem Manajemen Tugas Akhir (SIMANTA) untuk mahasiswa, dengan tujuan utama mengotomatisasi pengiriman pesan pemberitahuan untuk setiap kegiatan

atau perubahan yang terjadi di dalam sistem tersebut. Menggunakan metodologi ADDIE yang dikenal karena sifatnya yang fleksibel, fitur Notifikasi WhatsApp API berhasil diimplementasikan ke dalam SIMANTA, memastikan bahwa mahasiswa dan pengguna terkait dapat menerima informasi penting dengan lebih gampang dan cepat terkait semua aksi yang mereka lakukan di sistem, seperti persetujuan, revisi, atau pengumuman. Keberhasilan sistem ini dalam mengirimkan pesan otomatis berdasarkan pemicu (trigger) sistem membuktikan kapabilitas API WhatsApp sebagai infrastruktur andal untuk layanan pesan notifikasi yang berskala massal di lingkungan akademik [44].

3. **Implementasi API WhatsApp pada Rancang Bangun Sistem Informasi Penjualan Berbasis Web (Nue, Chandra, & Baly Woda, Jurnal In Create, 2024)**, Karya ilmiah ini memaparkan tentang perancangan dan pembangunan Website untuk sistem informasi penjualan di *Berlian Caffe & Shop* Maumere yang mampu menangani transaksi baik secara luring (*offline*) maupun daring (*online*), di mana sebelumnya pencatatan dilakukan secara manual dan rentan kesalahan. Untuk meningkatkan efisiensi dan layanan pembeli, sistem ini diintegrasikan dengan API WhatsApp sehingga mampu mengirimkan rincian lengkap mengenai transaksi yang telah dilakukan, serta menyebarkan berita promosi terkini kepada pelanggan secara otomatis dan *real-time*. Penerapan sistem berbasis Web dengan API ini secara signifikan membantu pemilik, kasir, dan masyarakat umum dalam mengakses dan memproses informasi penjualan, sekaligus memastikan

komunikasi pemasaran berjalan lebih lancar dan efektif, yang sangat mendukung konsep pengiriman pesan massal (*broadcast*) dalam konteks bisnis [57].

2.9. Kerangka Penelitian

Tabel 2.10. Kerangka Penelitian

| NO | TAHAPAN | DESKRIPSI LOGIS | OUTPUT YANG DIHARAPKAN |
|----|--|---|--|
| 1 | Permasalahan Awal | PT. Infomedia Nusantara memerlukan sistem yang efisien dan skalabel untuk mengirimkan pesan massal (<i>broadcast</i>) dan notifikasi kepada pelanggan, yang saat ini masih dilakukan secara manual atau menggunakan sistem dengan keterbatasan. | Identifikasi kebutuhan fungsional dan non-fungsional dari sistem <i>broadcast</i> yang baru. |
| 2 | Variabel <i>Input</i> (Komponen Utama) | a. WhatsApp API: Teknologi utama untuk memungkinkan pengiriman pesan massal terprogram. b. <i>Web-Based System</i> : Platform antarmuka yang digunakan oleh admin/staf untuk mengelola data kontak, membuat pesan, dan | Ketersediaan API Key WhatsApp Business dan spesifikasi teknis untuk perancangan <i>database</i> serta antarmuka Web. |

| NO | TAHAPAN | DESKRIPSI LOGIS | OUTPUT YANG DIHARAPKAN |
|----|--|---|---|
| | | <p>menjalankan <i>broadcast</i>.</p> <p>c. <i>Database</i> Kontak: Tempat penyimpanan data pelanggan PT. Infomedia Nusantara yang akan menjadi target <i>broadcast</i>.</p> | |
| 3 | Proses Perancangan Sistem | <p>Metode yang digunakan untuk merancang dan membangun sistem (berkemungkinan menggunakan metode <i>Waterfall</i> seperti yang disinggung di salah satu jurnal referensi). Tahapan meliputi: Analisis Kebutuhan, Perancangan Sistem, Implementasi (Koding), dan Pengujian Sistem.</p> | <p>a. <i>Blueprint</i> Sistem (UML/DFD).</p> <p>b. Kode Program (PHP/Laravel/CodeIgniter/dll).</p> <p>c. Sistem Web <i>Broadcast</i> WhatsApp yang berfungsi.</p> |
| 4 | Variabel <i>Output</i> (Sistem Terintegrasi) | <p>Sistem Web <i>Broadcast</i> WhatsApp Terintegrasi API yang memiliki fitur-fitur seperti <i>upload</i> daftar kontak, <i>template</i> pesan, penjadwalan <i>broadcast</i>, dan laporan status pengiriman pesan.</p> | <p>Sebuah sistem yang teruji (Black Box Testing) dan tervalidasi yang mampu menggantikan proses <i>broadcast</i> manual.</p> |

| NO | TAHAPAN | DESKRIPSI LOGIS | OUTPUT YANG DIHARAPKAN |
|----|-------------------------|---|--|
| 5 | Pengukuran Keberhasilan | Pengujian fungsionalitas sistem (Black Box) untuk memastikan semua fitur berjalan sesuai rencana. Jika ada, uji efektivitas penggunaan oleh staf PT. Infomedia Nusantara. | a. Persentase Keberhasilan Fungsionalitas. b. Kesimpulan bahwa sistem <i>broadcast</i> terintegrasi API lebih efisien, cepat, dan terukur dibandingkan metode manual. |