

BAB II

TINJAUAN LITERATUR

2.1 Penelitian Sebelumnya

Jurnal “Aplikasi Pembelajaran Bahasa Jepang Dasar (Hiragana dan Katakana) Untuk Pelajar Berbasis Android” menjadi referensi dalam penelitian ini karena membahas pengembangan aplikasi pembelajaran bahasa Jepang yang berfokus pada huruf Hiragana dan Katakana. Penelitian ini menggunakan metode System Development Life Cycle (SDLC), yang memberikan pendekatan sistematis dalam pengembangan aplikasi (Rasyiq et al., 2023).

Jurnal “Optimalisasi Pembelajaran Klausa Bahasa Jepang Melalui Perancangan Website sebagai Media Pembelajaran di LPK IHMI Cirebon” menjadi referensi utama dalam penelitian ini karena memiliki relevansi yang erat dengan topik yang dibahas. Penelitian tersebut membahas implementasi website berbasis WordPress sebagai sarana pembelajaran bahasa Jepang, khususnya dalam memahami klausa bahasa Jepang, yang masih menjadi tantangan bagi banyak pembelajar. Studi ini menjadi landasan dalam penelitian mengenai manfaat Progressive Web Application (PWA) dalam pembelajaran bahasa Jepang bagi mahasiswa Universitas Muhammadiyah Bengkulu, sekaligus menjadi penerapan teknologi pembelajaran berbasis web (Ardiansyah & Purnamasari, 2024).

Jurnal “Simpuru: Gamifikasi Pembelajaran Bahasa Jepang dalam Aplikasi Berbasis Web” menjadi referensi dalam penelitian ini karena memiliki relevansi yang kuat dengan penggunaan teknologi dalam pembelajaran bahasa Jepang. Penelitian ini mengintegrasikan gamifikasi ke dalam sistem pembelajaran berbasis web untuk mengatasi tantangan yang sering dihadapi oleh mahasiswa, seperti demotivasi, kebosanan, dan kesulitan memahami huruf kanji. Dengan adanya fitur gamifikasi, pelajar dapat lebih aktif dalam belajar melalui sistem poin, avatar, leaderboard, dan berbagai insentif lainnya yang meningkatkan motivasi belajar (Pangaribuan et al., 2022).

Jurnal “Aplikasi Belajar Bahasa Jepang Berbasis Android” menjadi referensi penting dalam penelitian ini karena membahas penggunaan teknologi mobile dalam pembelajaran bahasa Jepang. Penelitian dalam jurnal ini menggunakan metode Extreme Programming sebagai pendekatan pengembangan perangkat lunak, sementara penelitian ini mengadopsi metode waterfall yang lebih terstruktur dalam setiap tahapannya. Selain itu, jurnal ini juga menunjukkan bagaimana mobile learning dapat meningkatkan motivasi dan keterlibatan mahasiswa, yang sejalan dengan tujuan penelitian ini dalam penerapan teknologi PWA terhadap pembelajaran bahasa Jepang (Rusli & Nurahman, 2018).

Jurnal “Pengembangan Aplikasi Belajar Bahasa Jepang Berbasis Website” menjadi referensi dalam penelitian ini karena membahas pengembangan aplikasi ManGo yang bertujuan memberikan layanan

pendidikan online berbasis web bagi pengguna yang ingin mempelajari bahasa Jepang. Pada penelitian ini menerapkan metode Agile Scrum, ManGo memungkinkan pengelolaan materi secara sistematis, serta menyediakan fitur kuis interaktif untuk menguji pemahaman pengguna terhadap huruf Hiragana, yang sejalan dengan tujuan penelitian dalam mengembangkan Progressive Web Application (PWA) untuk pembelajaran bahasa Jepang (Sisephaputra et al., 2023).

Pada penelitian sebelumnya yang dijadikan sebagai referensi dalam penelitian ini memiliki keterkaitan langsung dengan proses penelitian penulis. Setiap penelitian sebelum ini mencakup aspek penggunaan teknologi mobile dan web untuk pembelajaran, optimalisasi interaktivitas melalui gamifikasi, serta metode pengembangan sistem yang terstruktur. Dengan mengacu pada penelitian tersebut, penelitian ini dapat merancang sistem pembelajaran bahasa Jepang yang lebih modern, fleksibel, dan efektif, sekaligus memastikan bahwa aplikasi yang dikembangkan memiliki antarmuka yang responsif, fitur interaktif, serta mampu meningkatkan motivasi dan pemahaman mahasiswa terhadap materi yang dipelajari.

2.2 Progressive Web Application

Progressive Web Application (PWA) merupakan sebuah standar baru yang direkomendasikan oleh grup *Web Fundamentals Google* sebagai usaha untuk menghubungkan kesenjangan antara aplikasi *native* dan aplikasi *website* dengan mengenalkan fitur seperti layar utama, dukungan *offline*, dan

background synchronization. PWA menyatukan pengalaman pengguna aplikasi *web* dan aplikasi *native* seperti kecepatan akses, dapat diakses meskipun tidak terhubung internet, dijalankan secara *fullscreen* mempunyai fitur *push notification*, dan bisa dipasang di *home screen*. Sebuah aplikasi web bisa disebut sebagai PWA, ada beberapa syarat, yaitu disajikan melalui *Hypertext Transfer Protocol Secure* (HTTPS) untuk alasan keamanan, dilengkapi dengan *manifest* aplikasi web untuk mendeklarasikan *metadata* aplikasi, dan menggunakan setidaknya satu *service worker* (Mochamad Ocktafian Setiadi et al., 2023).

2.3 Service Worker

Service worker adalah kode Javascript yang dijalankan oleh *browser* di latar belakang, dan terpisah dengan skrip lainnya. *Service worker* akan mengembalikan aset website yang sudah disimpan sebelumnya, hal ini membuat sebuah website dapat bekerja layaknya aplikasi native Android dan *desktop*. Dengan kemampuan menyimpan *asset website* secara lokal, *service worker* juga meningkatkan kecepatan akses sebuah website dan menghemat data.

2.4 Software Development Life Cycle (SDLC)

Software Development Life Cycle (SDLC) merupakan sebuah metode pengembangan perangkat lunak secara sistematis dan memungkinkan menyelesaikan software dalam batas waktu dan menjaga kualitas sesuai standar. Model SDLC Waterfall merupakan sebuah proses untuk

mengembangkan perangkat lunak yang secara umum banyak digunakan (Mahfudh et al., 2022).

2.5 Bahasa Jepang

Bahasa Jepang merupakan salah satu bahasa asing yang banyak dipelajari di Indonesia. Menurut Japan Foundation, Indonesia menjadi pembelajar bahasa Jepang terbanyak kedua di dunia, sebanyak 872.441 orang.

Bahasa Jepang adalah bahasa yang memiliki ragam keunikan. Menurut Sudjianto, bahasa Jepang adalah bahasa yang dinyatakan dengan tulisan yang menggunakan huruf-huruf seperti *kanji*, *hiragana*, *katakana*, dan *romaji*. *Hiragana* merupakan tata bahasa Jepang asli, sedangkan *katakana* merupakan huruf kata serapan. *Kanji* sendiri merupakan huruf serapan dari bahasa Cina.

Tata bahasa Jepang sangat berbeda dengan bahasa Inggris atau bahasa lainnya. Oleh karena itu, penting untuk mempelajari tata bahasa dasar seperti partikel, konjugasi, kata kerja, dan bentuk kata sifat terlebih dahulu.

2.6 Laravel

Laravel adalah suatu kerangka kerja yang berbasis bahasa pemrograman PHP. Laravel juga terdiri dari konsep MVC (*Model*, *View*, *Controller*) yang merupakan sebuah konsep modern yang memisahkan tampilan (*front-end*) dan juga bagian pengelolaan data atau sering disebut *controller* (*back-end*). Laravel memiliki beberapa fitur seperti, rute atau routing, fitur ini mengatur request yang telah ditentukan berdasarkan kebutuhan, lalu Laravel juga memiliki *query builder* dan ORM yang

memberikan kemudahan dalam melakukan operasi *database* dan mendukung berbagai *database*. Lalu fitur composer dapat mempermudah pengembang saat membutuhkan modul atau library tambahan dan dapat melakukan pembaruan pada modul tersebut. Selain itu, masih ada fitur-fitur lain yang mempermudah proses perancangan aplikasi (Fikri Ahmad Fauzi & Fajar Darmawan, 2023).

2.7 Algoritma Fisher-Yates Shuffle

Algoritma *fisher yates shuffle* merupakan algoritma pengacakan yang metodenya digunakan untuk menghasilkan permutasi acak dari himpunan terhingga dengan bahasa lain. Algoritma ini digunakan untuk mengacak himpunan tersebut (Rohmah et al., 2020).

2.8 Flowchart

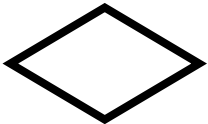
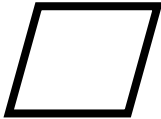
Flowchart atau bagan alur merupakan sebuah diagram yang menampilkan langkah-langkah dan keputusan untuk melakukan sebuah proses dari suatu program. Setiap langkah digambarkan dalam bentuk diagram yang dihubungkan dengan garis atau arah panah (Saputra et al., 2024).

Flowchart berperan penting dalam memutuskan sebuah langkah atau fungsionalitas dari sebuah proyek pembuatan program yang melibatkan banyak orang sekaligus. Selain itu dengan menggunakan bagan alur proses dari sebuah program akan lebih jelas, ringkas, dan mengurangi kemungkinan untuk salah penafsiran. Penggunaan flowchart dalam dunia pemograman juga

merupakan cara yang bagus untuk menghubungkan antara kebutuhan teknis dan non-teknis.

Berikut ini adalah simbol-simbol yang digunakan dalam *Flowchart*:

Tabel 2.1 Simbol-simbol Flowchart

Simbol	Fungsi	Keterangan
	Terminasi	Menunjukkan awal atau akhir sebuah proses
	Proses atau langkah	Merupakan serangkaian operasi yang mengubah nilai, bentuk, atau lokasi data
	Garis alir	Menunjukkan arah aliran proses atau algoritma
	Keputusan	Memperlihatkan operasi kondisi untuk menentukan salah satu dari dua jalur yang akan diambil oleh program
	Input/Output	Menunjukkan suatu proses input yang berasal dari pengguna atau output ke layar.

2.9 UML (Unified Modelling Language)

Unified Modelling Language (UML) adalah bahasa pemodelan visual yang dirancang khusus untuk pengembangan dan analisis sistem berorientasi objek dan desain (Prayitno et al., 2023).

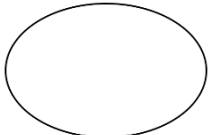
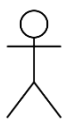
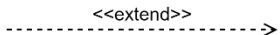
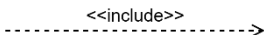
Berikut ini adalah pemodelan yang termasuk ke dalam UML:

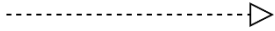
1. Use Case Diagram

Use Case Diagram merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. *Use Case Diagram* mendeskripsikan sebuah interaksi satu atau lebih aktor dengan sistem informasi yang akan dibuat.

Berikut ini adalah simbol-simbol yang dipakai dalam *Use Case Diagram* beserta dengan fungsinya:

Tabel 2.2 Simbol-simbol Use Case diagram

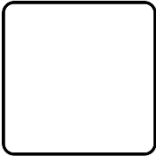
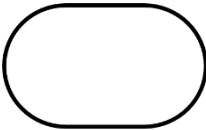
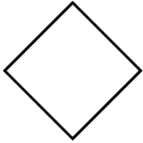
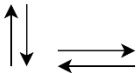
Simbol	Fungsi	Keterangan
	Use Case	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.
 Actor	Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan use case
	Asosiasi	Abstraksi dari penghubung antara aktor dengan use case
	Ekstensi	Menunjukkan bahwa suatu use case merupakan tambahan fungsional dari use case lainnya jika suatu kondisi terpenuhi
	Include	Menunjukkan bahwa suatu use case seluruhnya merupakan fungsionalitas dari use case lainnya

Simbol	Fungsi	Keterangan
	Generalisasi	Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan use case.

2. Activity Diagram

Activity Diagram adalah suatu diagram yang menggambarkan konsep aliran data, kontrol, aksi terstruktur serta dirancang dengan baik dalam suatu sistem.

Tabel 2.3 Simbol-simbol Activity diagram

Simbol	Nama	Keterangan
	Activity	Memperlihatkan bagaimana masing masing kelas antarmuka saling berinteraksi satu sama lain
	Action	State dari sistem yang mencerminkan eksekusi dari suatu aksi
	Initial node	Bagaimana objek dibentuk atau diawali
	Activity final node	Bagaimana objek dibentuk dan diakhiri
	Decision	Digunakan untuk menggambarkan bagaimana suatu keputusan/tindakan yang harus diambil pada kondisi tertentu
	Line Connector	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya