

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Dalam pengerjaan judul ini, penelitian terdahulu digunakan sebagai acuan untuk memberikan ide dan referensi yang dapat membantu dalam pembuatan sistem. Terdapat lima penelitian sebelumnya yang diambil dalam kurun waktu lima tahun terakhir. Penelitian pertama dilakukan oleh William Lourensius, dkk. (2023) yang berjudul “Pembangunan Sistem *Marketplace Event* Berbasis Web dan Mobile”. Pengembangan sistem dalam penelitian ini dilakukan dengan menggunakan *framework* Laravel sebagai *back-end*, Vue.js sebagai *front-end (website)* dan Flutter sebagai *front-end (mobile)*. Sistem yang dirancang dari penelitian ini adalah aplikasi pengelolaan *event* berbasis *website* serta sistem pembelian tiket berbasis *mobile*. Tujuan dari penelitian ini adalah untuk membuat sistem *marketplace event* yang dinamis untuk memudahkan pengelolaan acara.

Penelitian selanjutnya dilakukan oleh Dian Jois, dkk. (2024) yang berjudul “Perancangan Sistem Informasi *Event Organizer* Berbasis Aplikasi Mobile dengan Menggunakan Metode *Prototype*”. Penelitian ini membahas mengenai permasalahan dimana masyarakat kesulitan untuk mencari tempat mengembangkan bakatnya. Sehingga, melalui penelitian ini, penulis membuat *prototype* Sistem *Event Organizer* untuk mempertemukan orang yang memiliki bakat yang tidak tersalur serta pembuat *event* dalam satu aplikasi. Adapun metode yang digunakan adalah *Prototype* dan hasilnya diimplementasikan dengan menggunakan *framework* React Native. Prototipe yang dibuat diujikan dengan menggunakan *User Experience Questionnaire* (UEQ).

Kemudian, terdapat penelitian terdahulu yang dilakukan oleh Raul Irawan Hermanto dan Malabay (2021) yang berjudul “Rancang Bangun Aplikasi Pemesanan Tiket *Event* di Kota Jakarta Berbasis Website”. Penelitian ini bertujuan untuk membangun aplikasi yang menghubungkan penyelenggara *event* dan pengunjung dalam proses jual beli tiket *event*. Melalui *platform* yang dibuat, pihak EO dapat mempromosikan *event* sementara pengunjung dapat melakukan pemesanan tiket melalui *website*. Sistem dikembangkan dengan menggunakan metode *waterfall* dengan *framework* Laravel serta memanfaatkan *database* MySQL. Selain fitur pemasaran dan pembelian tiket, terdapat pula implementasi QR Code

yang menjamin keamanan tiket yang dibeli serta fitur Gateway Midtrans. Sistem diuji dengan menggunakan metode pengujian *black box*.

Penelitian berikutnya dilakukan oleh Yosafat Hizkia Pesik, dkk. (2022) yang berjudul “Perancangan dan Pembuatan Aplikasi Manajemen Acara Berbasis Mobile Menggunakan Flutter”. Penelitian ini menghasilkan sebuah aplikasi manajemen acara berbasis *mobile* untuk memudahkan proses pengelolaan *event* yang sebelumnya manual. Sistem dibangun dengan menggunakan metode *Waterfall* serta diuji dengan menggunakan metode *black box testing*. Adapun *framework* yang digunakan adalah Flutter dengan *database* Firebase. Aplikasi yang dirancang menawarkan fitur menambah *event*, mencari *event*, mendaftar *event*, dan mengundang teman untuk mengikuti suatu acara.

Penelitian terakhir dilakukan oleh Ibrahim A. Abu-ALSondos, dkk. (2020) yang berjudul “*Organizing Event Ubiquitous With a Proposed Event Mobile Application in Bahrain*”. Penelitian ini membahas mengenai masyarakat serta wisatawan di Bahrain yang memiliki banyak minat dan hobi tetapi tidak tahu dimana mencari acara atau festival yang berkaitan dengan minat mereka. Oleh karena itu, penelitian ini berfokus untuk mengembangkan sebuah aplikasi *mobile* untuk membagikan beragam acara yang ada di Bahrain untuk menghubungkan orang-orang dengan minat yang sama dan mengabari wisatawan mengenai acara yang ada di Bahrain. Pengembangan aplikasi dilakukan dengan menggunakan metode *agile* dan memanfaatkan *software* Appy Pie. Terdapat fitur untuk pengelolaan dan pencarian *event*, *live chat*, pengiriman saran dan *request* atau permintaan acara, dan lain sebagainya.

Berdasarkan penelitian terdahulu, kebanyakan sistem yang dirancang hanya memanfaatkan salah satu *platform* antara *mobile* dan *website*. Dari 5 penelitian yang disebutkan, terdapat 1 penelitian yang membangun sistem *marketplace event* berbasis *mobile* dan web. Pada penelitian kedua terdapat pembangunan sistem baik bagi partisipan maupun EO namun hanya berbasis *mobile*. Penelitian ketiga dikembangkan untuk kedua pihak partisipan dan EO tetapi hanya dikembangkan pada *platform website*. Penelitian keempat menghasilkan sistem berbasis *mobile* bagi masyarakat untuk mencari acara. Oleh karena itu, perancangan sistem ini dilakukan berfokus kepada 3 *role* yaitu masyarakat, admin, dan EO. Sistem untuk masyarakat akan dirancang berbasis *mobile* untuk sementara sistem untuk EO dan admin akan dirancang berbasis *website*. *Website* yang dirancang akan mencakup aspek-aspek seperti sistem pemasaran kegiatan dan *dashboard* pembelian

serta peserta untuk EO, pengelolaan sistem untuk admin, dan fitur pencarian dan pendaftaran kegiatan bagi masyarakat.

Tabel 2.1 Penelitian Terdahulu

Judul	Metode	Tools	Hasil	Kekurangan
Pembangunan Sistem <i>Marketplace Event</i> Berbasis <i>Web</i> dan <i>Mobile</i>	Tidak disebutkan	<i>Framework:</i> Laravel, Vue, Flutter; <i>Database:</i> MySQL	Sistem informasi aplikasi pengelolaan <i>event</i> berbasis <i>website</i> , dibangun dengan <i>framework</i> Laravel dan Vue.js, serta sistem pembelian tiket acara berbasis <i>mobile</i> yang dibangun dengan menggunakan <i>framework</i> Laravel. Sistem ini memudahkan EO untuk mengelola sebuah acara dan memudahkan pencarian dan pembelian tiket.	Sistem informasi yang dibuat masih tidak memiliki fitur untuk pembayaran.
Perancangan Sistem Informasi <i>Event Organizer</i> Berbasis Aplikasi	<i>Prototyping, User Experience Questionnaire</i> (UEQ)	<i>Framework:</i> React Native; <i>Software:</i> Adobe XD (untuk <i>prototype</i>)	Menggunakan metode <i>prototype</i> untuk menghasilkan perancangan desain aplikasi	Pengembangan sistem hanya dilakukan berbasis <i>mobile</i> baik bagi <i>user</i>

Judul	Metode	Tools	Hasil	Kekurangan
<i>Mobile dengan Menggunakan Metode Prototype</i>			<i>Event Organizer</i> berbasis <i>mobile</i> untuk mempertemukan orang yang memiliki bakat serta pembuat <i>event</i> . Rancangan yang dibuat telah dilakukan pengujian dengan menggunakan UEQ dan diperoleh hasil yang bagus.	maupun EO. Adapun fitur yang diperuntukan EO hanya terbatas pada pembuatan dan pengelolaan <i>event</i> .
Rancang Bangun Aplikasi Pemesanan Tiket <i>Event</i> di Kota Jakarta Berbasis <i>Website</i>	<i>Waterfall, Black Box Testing</i>	Bahasa pemrograman: PHP (Web); <i>Framework</i> : Laravel (<i>back-end</i>), Bootstrap (UI); <i>Database</i> : MySQL	Aplikasi pemesanan tiket <i>event</i> berbasis <i>website</i> yang dapat membantu pihak EO untuk mempromosikan tiket <i>event</i> serta memudahkan <i>user</i> untuk menemukan <i>event</i> di sekitar wilayah kota Jakarta. Terdapat fitur pembelian tiket secara	Sistem yang dikembangkan hanya berbasis <i>website</i> dan tidak dikembangkan untuk <i>platform mobile</i> sehingga target pengguna perlu menggunakan PC atau mode <i>desktop</i> pada browser.

Judul	Metode	Tools	Hasil	Kekurangan
			online dimana setiap tiket akan memiliki QR Code yang unik untuk registrasi ulang.	
Perancangan dan Pembuatan Aplikasi Manajemen Acara Berbasis <i>Mobile</i> Menggunakan Flutter	<i>Waterfall, Black Box Testing</i>	<i>Framework: Flutter; Database: Firebase</i>	Aplikasi manajemen acara berbasis <i>mobile</i> untuk mempermudah pengguna dalam mencari dan berpartisipasi serta membuat dan memasarkan <i>event</i> . Aplikasi dikembangkan dengan <i>framework</i> Flutter dan digunakan Firebase sebagai <i>database</i> . Adapun fitur yang ditawarkan adalah: <i>login</i> dan registrasi, modifikasi profil, pencarian <i>event</i> , pendaftaran <i>event</i> dimana pengguna	Sistem yang dikembangkan hanya berbasis <i>mobile</i> dengan 1 <i>user role</i> saja yakni pengguna. EO tidak memiliki <i>role</i> yang berbeda sehingga fitur yang ada tidak terlalu berfokus untuk manajemen <i>event</i> melainkan lebih banyak berfokus ke partisipasi dan pencarian <i>event</i> .

Judul	Metode	Tools	Hasil	Kekurangan
			dapat mengundang teman untuk berpartisipasi, pembayaran dan pencetakan tiket, pembuatan <i>event</i> , dan bertukar pesan. Hasil pengujian <i>black box</i> menyatakan bahwa semua skenario aktifitas pengujian berhasil.	
Organizing Event Ubiquitous With a Proposed Event Mobile Application in Bahrain	<i>Agile Methodology</i>	<i>Software: Appy Pie (mobile), Photoshop (UI)</i>	Aplikasi <i>mobile</i> untuk masyarakat dan wisatawan di Bahrain yang berfokus untuk memasarkan <i>event-event</i> yang ada serta mempertemukan orang-orang dengan minat yang sama untuk merencanakan sebuah <i>event</i> atau pertemuan. <i>Interface</i> aplikasi dirancang	Meskipun fiturnya lengkap dan menarik, tetapi tidak terdapat implementasi fitur untuk registrasi acara sehingga akan sulit bagi panitia acara untuk mengelola registrasi ulang pada saat hari acara. Selain itu, akan lebih baik jika pembuat

Judul	Metode	Tools	Hasil	Kekurangan
			dengan menggunakan Photoshop dan aplikasi dibangun dengan menggunakan Appy Pie.	acara (Admin) dapat melihat hasil penjualan <i>event</i> yang diadakan.
Rancang Bangun Sistem Pemasaran dan Registrasi Acara menggunakan Metode <i>Rapid Application Development</i> (Penelitian saat ini)	<i>Rapid Application Development, Black Box Testing</i>	Bahasa pemrograman: Flutter; <i>Database</i> : Firebase; <i>Design</i> : Figma.	Hasil dari judul ini adalah untuk merancang sistem pemasaran acara bagi EO yang berbasis <i>website</i> dan pencarian acara bagi masyarakat berbasis <i>mobile</i> . Sistem mencakup fitur-fitur untuk pencarian acara, pembelian tiket, dan pengingat bagi pengguna. Bagi EO, sistem menawarkan menu penjualan tiket, registrasi peserta, serta <i>dashboard</i> penjualan.	Belum dilakukan pengujian.

2.2 Landasan Teori

2.2.1 Acara / Event

Acara atau *event* adalah sebuah kejadian khusus dan penting yang terjadi di tempat dan waktu tertentu, memiliki awalan dan akhiran (Getz, 2007). Terdapat beragam jenis acara namun secara garis besar, menurut Noor (2009) terdapat empat jenis *event*, yaitu *leisure event*, *personal event*, *cultural event*, dan *organizational event*. *Leisure event* berkaitan dengan kegiatan hiburan atau olahraga, *personal event* melibatkan kepentingan pribadi atau keluarga, *cultural event* berhubungan dengan nilai dan budaya, sedangkan *organizational event* diselenggarakan untuk tujuan organisasi seperti konferensi atau pameran.

2.2.2 Event Organizer

Event Organizer adalah profesi yang merencanakan sebuah *event* dan mengumpulkan audiensi dengan tujuan selebrasi, edukasi, pemasaran, dan reuni (Goldblatt, 2013). Saat ini perkembangan penyelenggaraan acara di Indonesia semakin meningkat terutama pasca pandemi dimana terdapat beragam *event* yang diselenggarakan mulai dari seminar daring atau webinar, konser musik, pameran seni, dan lain sebagainya terutama di kota-kota besar. Hal ini menjelaskan bahwa kebutuhan akan penyelenggaraan acara semakin meningkat.

2.2.3 Sistem Pemasaran

Menurut KBBI, sistem adalah perangkat unsur yang saling berkaitan sehingga membentuk suatu totalitas. Sementara, menurut Stanton dalam Swastha & W. (2002), pemasaran adalah serangkaian kegiatan usaha yang meliputi perencanaan, penetapan harga, promosi, dan distribusi barang atau jasa untuk memenuhi kebutuhan pelanggan. Dengan demikian, sistem pemasaran merupakan keseluruhan kegiatan yang terstruktur dan saling terhubung dalam merencanakan, mempromosikan, serta mendistribusikan produk atau jasa guna mencapai kepuasan pelanggan.

Berdasarkan (Rizkiawan, 2021), sistem pemasaran yang paling sederhana terdiri dari dua unsur yang saling berinteraksi, yaitu organisasi yang dipasarkan dan pasar yang dituju. Kedua unsur tersebut dihubungkan oleh dua aliran, yaitu:

- 1) Aliran organisasi pemasaran yang mendistribusikan barang

dan jasa kepada konsumen.

- 2) Aliran informasi dimana perusahaan memperkenalkan produknya kepada konsumen melalui promosi atau iklan.

Dalam penelitian ini, *Event Organizer* ialah organisasi atau pihak yang akan mendistribusikan acara kepada konsumen sementara Masyarakat adalah konsumen atau target pasar yang dituju.

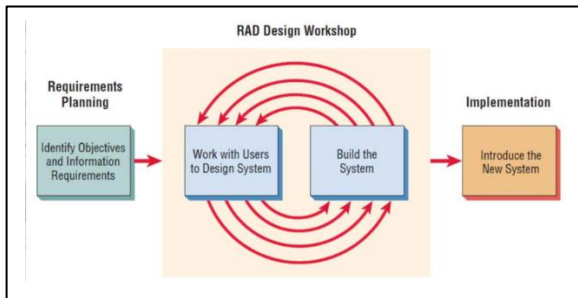
2.2.4 Registrasi Acara

Registrasi dalam KBBI diartikan sebagai sebuah pencatatan atau pendaftaran. Dalam suatu acara, proses registrasi merupakan tahapan awal yang penting dimana dilakukan pendaftaran atau pencatatan peserta yang hadir. Menurut Gunawijaya dalam (Widiasury dkk., 2020), registrasi dibagi menjadi dua tahap:

- 1) Tahap Pendataan: Peserta memberikan informasi lengkap mengenai data pribadi serta program acara yang ingin diikuti. Penyelenggara memperoleh informasi lengkap dari setiap peserta untuk memastikan kebutuhan masing-masing peserta.
- 2) Tahap Pelaksanaan: Seluruh peserta melakukan registrasi, baik yang sudah terdaftar maupun yang belum. Tahap ini mencakup pendaftaran ulang (*re-registration*) bagi peserta yang sudah terdaftar dan pendaftaran di tempat (*on-site registration*) bagi yang belum terdaftar.

2.2.5 *Rapid Application Development* (RAD)

Rapid Application Development (RAD) adalah model proses pengembangan perangkat lunak yang bersifat incremental terutama untuk waktu pengerjaan yang pendek (Sukanto & Shalahuddin, 2016). Metode ini cocok untuk pengembangan perangkat lunak yang menekankan kecepatan dan efisiensi melalui penggunaan prototipe dan iterasi yang cepat. Tujuannya adalah untuk menghasilkan sistem yang memenuhi standar kualitas secara singkat dengan keterlibatan pengguna secara aktif selama proses pengembangan. Pengerjaan sistem dengan metode ini umumnya hanya berlangsung selama 60-90 hari.



Gambar 2.1 Tahapan Metode RAD
Sumber: Kendall & Kendall (2010)

Menurut Kendall & Kendall (2010), terdapat tiga tahapan dalam RAD yang diilustrasikan pada Gambar 2.1, yaitu:

- 1) Perancangan Kebutuhan (*Requirements Planning*): Tahapan ini merupakan tahapan perencanaan yang melibatkan pertemuan langsung antara klien dan pengembang untuk mengetahui tujuan dari sistem yang akan dibuat serta mengidentifikasi kebutuhan sistem.
- 2) Tahap Desain (*Design Workshop*): Pada tahapan ini, terjadi proses perancangan dan perbaikan yang digambarkan dalam sebuah *workshop*. Analis dan pengembang bekerja sama membangun dan merancang prototipe dan kebutuhan sistem sebagai representasi visual kepada pengguna. Kemudian, pengguna akan memberikan umpan balik dan pengembang akan melakukan perbaikan. Proses ini akan terus diulang hingga desain telah sesuai dengan keinginan pengguna.
- 3) Implementasi (*Implementation*): Pada tahapan ini, desain sistem yang telah dibuat akan dikembangkan menjadi sistem yang fungsional. Selain pengembangan, tahapan ini juga melibatkan pengujian dimana sistem yang sudah selesai diuji kepada pengguna.

2.2.6 Aplikasi *Mobile*

Menurut KBBI, aplikasi dalam konteks informatika adalah program komputer atau perangkat lunak yang didesain untuk mengerjakan tugas tertentu. Darmayuda (2014) membagi aplikasi menjadi dua jenis, yaitu aplikasi terkoneksi yang memerlukan koneksi *database* secara berkelanjutan dan aplikasi terputus yang hanya terhubung ke *database* saat proses pengambilan atau penyimpanan data.

Sementara itu, Turban (2012) mendefinisikan aplikasi *mobile* sebagai istilah yang digunakan untuk mendeskripsikan aplikasi internet yang berjalan pada *smartphone* atau perangkat *mobile* lainnya, membantu pengguna terhubung dengan layanan internet yang biasanya diakses melalui PC. Dalam penelitian ini, aplikasi yang dikembangkan merupakan sistem pemasaran dan registrasi acara yang berfungsi sebagai media promosi sekaligus *platform* pendaftaran dan pembelian tiket secara real-time melalui koneksi database.

2.2.7 Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis Linux yang mencakup sistem operasi, *middleware*, dan aplikasi (Safaat, 2011). Menurut Herlinah & Musliadi (2019), Android merupakan *platform* pemrograman yang dikembangkan oleh perusahaan Google untuk ponsel cerdas dan perangkat seluler lainnya. Android bisa dijalankan di berbagai perangkat ponsel cerdas yang dihasilkan oleh vendor yang berbeda-beda. Sejak pertama kali dirilis pada 23 September 2009 hingga saat ini, Android sudah berkembang dan merilis banyak versi yang masing-masing memiliki nama. Android menyediakan *platform* terbuka bagi para pengembang untuk membuat aplikasi mereka sendiri, memungkinkan akses yang luas terhadap kemampuan perangkat *mobile*.

2.2.8 Flutter

Flutter adalah *framework* pemrograman yang dibuat oleh Google untuk membuat aplikasi yang *cross platform* (Putra & Fu'adi, 2024). Dirilis pada 2017, Flutter tidak hanya dikembangkan untuk aplikasi berbasis *mobile* tetapi juga untuk aplikasi berbasis web dan desktop. Adapun keunggulan dari flutter adalah dapat digunakan untuk pengembangan multi-*platform* yaitu untuk Android dan IOS dengan kode atau codebase yang sama, sehingga tidak perlu membuat aplikasi yang berbeda untuk setiap *platform*. Selain itu, prototipe dan iterasi dapat dilakukan dengan lebih efisien karena kode dapat diubah dan dimuat ulang saat aplikasi sedang berjalan, sehingga aplikasi tidak perlu melalui proses build ulang apabila terdapat perubahan kode.

2.2.9 Website

Website adalah kumpulan halaman web yang saling terhubung dan dapat diakses melalui internet, berisi informasi dalam berbagai

format seperti teks, gambar, video, dan lainnya. Sementara, berdasarkan Hakim Lukmanul (2004), *website* adalah fasilitas internet yang menghubungkan dokumen dalam lingkup lokal maupun jarak jauh. Dokumen pada *website* disebut *web page*, dan *link* dalam *website* memungkinkan pengguna berpindah dari satu *page* ke *page* lain (*hypertext*), baik antara *page* yang disimpan dalam server yang sama maupun server di seluruh dunia.

2.2.10 PHP

Hypertext Preprocessor (PHP) merupakan salah satu bahasa pemrograman *open source* yang dikembangkan untuk pengembangan *website* dan dapat ditanamkan pada sebuah skrip HTML. Adapun bahasa ini berbasis kode-kode (*script*) yang digunakan untuk mengolah suatu data dan mengirimkannya kembali ke *web browser* menjadi kode HTML (Firman dkk., 2016). PHP banyak digunakan karena mudah dipelajari, lintas *platform*, dan mendukung integrasi dengan berbagai *database*, salah satunya MySQL. Dalam pengembangan sistem akan digunakan bahasa pemrograman PHP.

2.2.11 Laravel

Laravel merupakan salah satu *framework* berbasis PHP yang bersifat *open source* dan menggunakan konsep arsitektur *Model-View-Controller* (MVC). Konsep ini memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, *user interface*, dan bagian yang menjadi kontrol aplikasi (Gibran dkk., 2024). Laravel juga menyediakan beberapa fitur unggulan seperti sistem routing yang sederhana, *template engine* Blade untuk tampilan, ORM Eloquent untuk memanipulasi *database*, hingga *middleware* yang mendukung aspek keamanan dan otorisasi hak akses.

2.2.12 React.js

React.js merupakan *library* JavaScript yang dikembangkan oleh Facebook (Meta) untuk membangun antarmuka pengguna (*user interface*) yang interaktif dan dinamis untuk aplikasi web dan *mobile*. React memungkinkan pengembang untuk membuat *user interface* yang kompleks dengan set kode kecil terisolasi yang disebut “komponen”. React JS berbasis komponen yang mana dapat membuat beberapa komponen yang terenkapsulasi sehingga mengatur state-nya sendiri,

kemudian komponen tersebut digabungkan untuk membentuk user interfaces yang lebih kompleks. (Ubaform & Iswari, 2021). Dalam penelitian ini, React akan digunakan untuk membangun antarmuka *front-end* dari halaman web.

2.2.13 JSX

JSX atau JavaScript XML adalah sintaks yang digunakan di *library* React untuk menyederhanakan proses pembuatan antarmuka. JSX merupakan sintaks seperti XML atau HTML yang digunakan oleh React untuk memperluas ECMAScript sehingga teks seperti XML dan HTML dapat digunakan dengan kode JavaScript atau React dengan tujuan untuk memungkinkan React menampilkan pesan dan peringatan yang lebih berguna bagi pengembang (Ubaform & Iswari, 2021).

2.2.14 Inertia.js

Inertia.js adalah *library* JavaScript yang digunakan untuk menjembatani komunikasi antara *client-side* dan *server-side*. Inertia dikenal sebagai *modern monolith* yang memungkinkan pembuatan halaman web yang ditampilkan secara penuh di sisi *client* (Ahmad Fauzi dkk., 2023). Dalam penelitian ini, *library* Inertia akan digunakan untuk menghubungkan *framework back-end* Laravel dengan antarmuka *front-end* berbasis React.js sehingga pertukaran data dapat terjadi dengan lancar.

2.2.15 MySQL

MySQL adalah sebuah software RDBMS (*database server*) yang dapat mengelola basis data dengan sangat cepat dan menampung data dalam jumlah besar sehingga dapat diakses oleh banyak user dan dapat melakukan sinkronisasi (Lim & Silalahi, 2023). Dalam sebuah sistem, MySQL berfungsi sebagai tempat menyimpan informasi aset, pengguna, transaksi, dan data lain yang dibutuhkan.

2.2.16 Application Programming Interface (API)

Berdasarkan (Muri dkk., 2019) *Application Programming Interface* (API) adalah konsep fungsi antarmuka pemrograman aplikasi yang memungkinkan suatu aplikasi dapat diakses dan dimanfaatkan oleh aplikasi lain tanpa mengubah struktur kode maupun *database*. Salah satu

jenis API yang paling banyak digunakan adalah REST (*Representational State Transfer*), yaitu gaya arsitektur yang menggunakan protokol HTTP untuk mengatur komunikasi antara client dan server. REST bekerja dengan prinsip *stateless*, artinya setiap permintaan (*request*) dari *client* ke *server* harus berisi seluruh informasi yang diperlukan tanpa menyimpan status sesi sebelumnya. Komunikasi umumnya menggunakan format data seperti JSON (*JavaScript Object Notation*) atau XML untuk pertukaran informasi lintas *platform* (Muharom dkk., 2025). Dalam proyek ini, API digunakan sebagai penghubung antara aplikasi web berbasis Laravel sebagai *backend* dan aplikasi mobile berbasis Flutter sebagai *client* yang berkomunikasi melalui *endpoint* API.

2.2.17 QR Code

QR Code (*Quick Response Code*) adalah jenis kode matriks dua dimensi yang dirancang untuk dapat dibaca dengan cepat menggunakan alat pemindai (*scanner*) atau kamera ponsel. Dengan kemampuan untuk menyimpan data dalam jumlah yang lebih banyak dibandingkan dengan *barcode* satu dimensi, QR Code semakin banyak digunakan di berbagai aplikasi termasuk tiket digital, pembayaran, dan verifikasi identitas. Penggunaan QR Code dalam sistem tiket digital menawarkan kepraktisan dan kecepatan dalam proses validasi (Eka Judistira dkk., 2025).

2.2.18 Black Box Testing

Black Box Testing adalah metode pengujian perangkat lunak yang berfokus pada evaluasi fungsionalitas sistem tanpa mempertimbangkan struktur internal atau kode sumbernya. Menurut Pressman (2010), *Black Box Testing*, juga disebut pengujian perilaku, berfokus pada persyaratan fungsional perangkat lunak. Pengujian ini memerlukan perancangan *test case* yang disusun berdasarkan kebutuhan sistem. Metode ini memungkinkan pengembang untuk menguji serangkaian kondisi masukan yang nantinya akan melaksanakan semua persyaratan fungsional dari suatu program. Tujuan utama dari pengujian ini adalah memastikan perangkat lunak berfungsi sesuai dengan spesifikasi yang ditentukan dan memenuhi kebutuhan. Pada penelitian ini, teknik yang digunakan adalah *use case testing*, yaitu pengujian berdasarkan skenario interaksi antara pengguna dan sistem untuk memastikan setiap fungsi berjalan sesuai dengan kebutuhan pengguna.

2.2.19 Usability Testing

Usability Testing adalah metode evaluasi yang digunakan untuk mengukur sejauh mana sistem dapat digunakan secara efektif, efisien, dan memuaskan oleh pengguna. Metode ini melibatkan pengamatan langsung terhadap pengguna saat berinteraksi dengan sistem dan melalui pengumpulan umpan balik untuk mengidentifikasi perbaikan yang diperlukan. Salah satu alat pengujian *usability* yang sering digunakan adalah *System Usability Scale* (SUS). Dikembangkan oleh John Brooke pada tahun 1986, SUS terdiri dari 10 pertanyaan yang dinilai menggunakan skala Likert untuk memberikan penilaian cepat terhadap sistem. Adapun 10 pertanyaan yang digunakan akan berdasarkan adaptasi Indonesia berdasarkan penelitian yang dilakukan oleh (Sharfina & Santoso, 2017) SUS pada dasarnya dirancang untuk menghasilkan skor *usability*. Namun, penelitian oleh Lewis & Sauro (2009) menunjukkan bahwa SUS dapat dikelompokkan ke dalam dua aspek, yaitu *usability* dan *learnability*. Dalam pembagiannya, pertanyaan 4 dan 10 termasuk aspek *learnability* dan pertanyaan 1, 2, 3, 5, 6, 7, 8, dan 9 aspek *usability*.

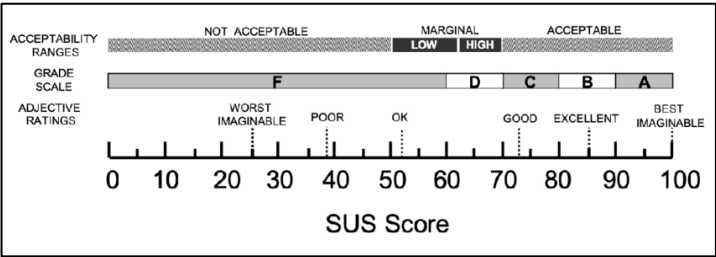
Skala Likert adalah alat pengukuran yang digunakan untuk mengukur sikap atau persepsi seseorang. Skala ini terdiri dari serangkaian pertanyaan yang diikuti oleh pilihan dengan tingkat persetujuan tertentu. Setiap pilihan jawaban diberikan skor numerik yang kemudian dapat dianalisis untuk menentukan persepsi responden terhadap pernyataan yang diberikan. Berikut adalah tabel yang berisi contoh 5 butir penilaian dengan skala Likert beserta dengan masing-masing nilainya.

Tabel 2.1 Skala Likert

Skor	Keterangan Skala
1	Sangat Tidak Setuju (STS)
2	Tidak Setuju (TS)
3	Netral (N)
4	Setuju (S)
5	Sangat Setuju (SS)

Perhitungan skor SUS dilakukan dengan mengurangi 1 pada setiap pertanyaan bernomor ganjil dan mengurangi skor dari 5 pada setiap pertanyaan bernomor genap untuk memperoleh nilai kontribusi masing-masing pertanyaan. Seluruh nilai kemudian dijumlahkan dan dikalikan 2,5 sehingga menghasilkan skor akhir dalam rentang 0–100. Skor tersebut selanjutnya diinterpretasikan menggunakan kategori penilaian yang

dikembangkan oleh Bangor dkk. (2009), yang meliputi *adjective ratings*, *grade scale*, dan *acceptability ranges* sebagaimana ditunjukkan pada Gambar 2.2 dan Tabel 2.3.



Gambar 2.2 Kategori Skor SUS
 Sumber: (Bangor dkk., 2009)

Tabel 2.2 Keterangan Penilaian Akhir Skor SUS

Skor	<i>Acceptability Ranges</i>	<i>Grade</i>	Skor SUS	Keterangan
0-50	<i>Not Acceptable</i>	F	0-25	<i>Worst Imaginable</i>
50-60	<i>Marginal (Low)</i>		26-39	<i>Poor</i>
60-70	<i>Marginal (High)</i>	D	40-51	<i>OK</i>
70-80	<i>Acceptable</i>	C	52-73	<i>Good</i>
80-90		B	74-84	<i>Excellent</i>
90-100		A	85-100	<i>Best Imaginable</i>

2.2.20 Populasi dan Sampel

Populasi adalah wilayah generalisasi yang terdiri atas obyek atau subyek yang mempunyai kualitas dan karakteristik tertentu yang ditetapkan oleh peneliti untuk dipelajari dan kemudian ditarik kesimpulannya (Sugiyono, 2013). Dalam penelitian ini, populasi difokuskan pada masyarakat usia produktif dengan rentang usia 15-64 tahun di Pekanbaru sebagai kelompok utama pengguna teknologi digital.

Sampel adalah bagian dari jumlah dan karakteristik yang dimiliki oleh populasi tersebut (Sugiyono, 2013). Penentuan ukuran sampel dalam penelitian ini menggunakan Rumus Slovin. Menurut Sugiyono (2013), rumus Slovin adalah rumus yang digunakan untuk menentukan jumlah sampel minimal yang dinilai mampu mewakili keseluruhan populasi. Adapun rumus perhitungannya adalah sebagai berikut:

$$n = \frac{N}{1 + N(e)^2}$$

Keterangan:

n = Ukuran sampel

N = Ukuran populasi

E = *Margin of error* (persentase batas toleransi kesalahan).

2.2.21 International Organization for Standardization (ISO)

ISO (International Organization for Standardization) adalah organisasi internasional non-pemerintah yang terdiri dari badan standardisasi nasional dari lebih dari 160 negara dan bertugas mengembangkan standar internasional untuk memastikan kualitas dan kesesuaian produk, proses, serta layanan. Salah satu standar yang dikembangkan adalah seri ISO/IEC 25000 SQuaRE (System and Software Quality Requirements and Evaluation), yang menyediakan kerangka kerja untuk mengevaluasi kualitas sistem dan perangkat lunak.

2.2.21.1 ISO 25010

Berdasarkan ISO/IEC 25010 (2022), model kualitas dalam ISO 25010 adalah kerangka yang digunakan untuk menilai kualitas suatu perangkat lunak. Model ini menentukan aspek-aspek apa saja yang harus diperhatikan saat mengevaluasi sistem. ISO 25010 mengelompokkan kualitas produk perangkat lunak ke dalam sembilan kategori utama yang disebut sebagai karakteristik kualitas, masing-masing dengan sub-kategori yang lebih spesifik. Karakteristik-karakteristik tersebut dijabarkan seperti pada Gambar 2.3.

SOFTWARE PRODUCT QUALITY								
FUNCTIONAL SUITABILITY	PERFORMANCE EFFICIENCY	COMPATIBILITY	INTERACTION CAPABILITY	RELIABILITY	SECURITY	MAINTAINABILITY	FLEXIBILITY	SAFETY
FUNCTIONAL COMPLETENESS	TIME BEHAVIOUR	CO-EXISTENCE	APPROPRIATENESS	FAULTLESSNESS	CONFIDENTIALITY	MODULARITY	ADAPTABILITY	OPERATIONAL CONSTRAINT
FUNCTIONAL CORRECTNESS	RESOURCE UTILIZATION	INTEROPERABILITY	RECOGNIZABILITY	AVAILABILITY	INTEGRITY	REUSABILITY	SCALABILITY	RISK IDENTIFICATION
FUNCTIONAL APPROPRIATENESS	CAPACITY		LEARNABILITY	FAULT TOLERANCE	NON-REPUDIATION	ANALYSABILITY	INSTALLABILITY	FAIL SAFE
			OPERABILITY	RECOVERABILITY	ACCOUNTABILITY	MODIFIABILITY	REPLACEABILITY	HAZARD WARNING
			USER ERROR PROTECTION		AUTHENTICITY	TESTABILITY		SAFE INTEGRATION
			USER ENGAGEMENT		RESISTANCE			
			INCLUSIVITY					
			USER ASSISTANCE					
			SELF-DESCRIPTIVENESS					

Gambar 2.3 9 Karakteristik Kualitas Perangkat Lunak ISO 25010
Sumber: ISO/IEC 25010 (2022)

2.2.21.2 ISO 25023

ISO 25023 adalah standar pengukuran kualitas perangkat lunak berdasarkan berbagai karakteristik yang telah ditentukan ISO 25010 (*ISO 25023:2016 - Systems and Software Engineering*, 2016). Standar ini termasuk dalam seri *Quality Measurement* (Pengukuran Kualitas) dari ISO/IEC 2502n dan menyediakan metode serta fungsi yang digunakan untuk menilai aspek kualitas perangkat lunak secara kuantitatif. Setiap karakteristik dan sub-karakteristik yang ditetapkan dalam ISO 25010 memiliki metode pengukuran yang berbeda. Untuk mengukurnya, digunakan model kualitas yang membagi kualitas ke dalam karakteristik dan sub-karakteristik dengan metrik dan rumus yang berbeda.

2.2.22 Performance Testing

Performance testing adalah pengujian perangkat lunak yang digunakan untuk menguji kecepatan, waktu respons, stabilitas, kehandalan, skalabilitas, dan penggunaan *resource* perangkat lunak di bawah beban kerja tertentu (Ismail dkk., 2023). Pengujian ini dapat bertujuan untuk mengidentifikasi dan menghilangkan hambatan teknis yang dapat memperlambat akses pengguna terhadap sistem.

2.2.23 Maze

Maze adalah sebuah *continuous product discovery platform* yang digunakan untuk melakukan *Usability Testing* secara jarak jauh dan tanpa moderasi (*unmoderated remote testing*). Alat ini memungkinkan peneliti UI/UX untuk mengumpulkan data kuantitatif dan kualitatif dari pengguna tanpa harus hadir secara fisik di samping partisipan (Maze, 2026).

2.2.24 Useberry

Useberry adalah sebuah *platform user experience (UX) research* dan pengujian *usability* yang dirancang untuk memungkinkan desainer dan peneliti produk melakukan pengujian jarak jauh tanpa moderasi (*unmoderated remote testing*) secara cepat dan efisien. Platform ini menjembatani kesenjangan antara desain prototipe dan wawasan pengguna nyata dengan memungkinkan integrasi langsung dari alat desain populer seperti Figma dan lainnya. (Useberry, 2026).