

BAB II

TINJAUAN PUSTAKA

2.1. Teori

Teori didasarkan pada data empiris, penelitian, atau eksperimen yang dilakukan dalam bidang tertentu. Teori sering kali berkembang seiring dengan penemuan baru dan pengalaman faktual (Surahman et al., 2020).

2.1.1. Teori Umum

1. *Software Development*

Dengan *Software Development* membantu melakukan pendataan dan mengelola laporan pajak bumi dan sering terhambat ketika melakukan rekapan data dalam skala besar sehingga sebelumnya mempersulit admin, setelah dilakukan perancangan sistem pengelolaan pajak bumi, pengguna mempunyai 3 level fasilitas yaitu pendata, administrasi dan Kasubbid dan mampu membantu dan meringankan pekerjaan manusia (Eva et al., 2022).

Dalam konteks pembelian tiket secara digital atau *e-ticketing*, *software development* mempunyai peran yang krusial dalam membangun platform yang memfasilitasi penjualan dan pembelian tiket secara digital, yang memiliki kelebihan tingkat keamanan, efisiensi dan kenyamanan yang optimal. Pengembangan sistem penjualan dan pembelian tiket dilengkapi fitur-fitur yang *user friendly* sehingga proses pembelian tiket lebih mudah dan praktis (Mohsa et al., 2023)

Saat ini proses transaksi tidak lagi dalam bentuk *cash*, tetapi telah berkembang dengan menggunakan kartu kredit, dompet digital, transfer bank dan berbagai opsi digital lainnya, sehingga tingkat keamanan didalam transaksi online menjadi hal yang harus diperhatikan dengan baik (Wahyudi, 2020).

Tingkat keamanan transaksi berbasis *software development* cukup ketat dengan didukung enkripsi yang kuat sehingga privasi data pengguna serta informasi pembayaran tetap aman. Selain itu, manajemen acara yang komprehensif, termasuk registrasi, manajemen tiket, penjadwalan dan pembatalan acara, merupakan fitur penting yang harus tersedia (M. A. Nugroho & Fachrie, 2024).

Setiap tahapan dalam menciptakan, menguji, dan memelihara *software* bergantung pada *software development* dalam konteks siklus hidup *software development*. Pendekatan sistematis untuk *software development* yang terdiri dari serangkaian tahapan yang terorganisir, dimulai dari identifikasi kebutuhan *software* hingga penerapan, pemeliharaan, dan penarikan produk (Puspa & Wulandari, 2021).

Software development pada tahapan perumusan masalah melibatkan pemahaman terkait kebutuhan pengguna yaitu mengumpulkan ide-ide kemudian melanjutkan pada tahapan pengembangan *prototipe*, dimana membuat versi awal dari solusi yang diusulkan. Kemudian, *prototipe* diuji

dengan user potensial untuk mendapatkan umpan balik awal kemudian digunakan untuk mengiterasikan serta memperbaiki solusi yang dilakukan tahap pengujian dan iterasi melibatkan pengujian *prototipe* dengan user potensial secara berulang untuk menemukan kelebihan dan kekurangan dan melakukan perbaikan yang diperlukan (Arisa et al., 2023).

Setelah analisis, tahap desain dimulai, dimana *software* dirancang secara baik teknis dan fungsional berdasarkan arsitektur dan kebutuhan yang telah ditentukan sebelumnya. Spesifikasi *design* seperti pembuatan diagram aliran data dan diagram struktur program adalah bagian dari tahapan desain (Parhusip et al., 2023).

Pembuatan aplikasi dapat mempermudah pengelolaan data dalam jumlah yang sangat besar seperti data pembelian, data pemesanan tiket dan mempermudah admin dalam membuat laporan dan rekomendasi karena mudahnya pembaharuan fitur yang dapat menyesuaikan kebutuhan *user* serta perbaikan kerusakan dalam tahapan pemeliharaan sistem (Refandaru et al., 2022).

2. *E-Ticketing dan Volunteer*

Untuk membuat solusi yang efektif dan efisien untuk mendukung acara dan kegiatan, *Software Development Life Cycle* (SDLC) yang terdiri dari sistem manajemen *volunteer* dan pengembangan platform *e-ticketing*, bertanggung jawab untuk mengidentifikasi kebutuhan bisnis dan teknis untuk platform *e-ticketing*, termasuk fitur yang diperlukan untuk membuat pembelian tiket secara aman dan mudah secara online (Rahayu et al., 2024).

Selain itu, perencanaan harus mempertimbangkan peran yang dimainkan oleh *volunteer* dalam acara, seperti tugas yang diharapkan dan kebutuhan sistem manajemen *volunteer* (Makruf et al., 2023). Pengembangan *software* yang terdiri dari sistem manajemen *volunteer* dan pengembangan platform *e-ticketing*, bertanggung jawab untuk mengidentifikasi kebutuhan bisnis dan teknis untuk platform *e-ticketing*, termasuk fitur yang diperlukan untuk membuat pembelian tiket secara aman dan mudah secara online. Selain itu, perencanaan harus mempertimbangkan peran yang dimainkan oleh *volunteer* dalam menyediakan solusi yang efektif dan efisien untuk mendukung acara dan kegiatan.

Manajemen fesitaval ponorogo yang diadakan setiap satu tahun sekali mengadakan perbaikan sistem penjualan mulai dari penjualan tiket yang dilakukan secara online sehingga mempermudah masyarakat dalam pembelian tiket sehingga dapat meningkatkan hasil penjualan hal ini terbukti dengan adanya data yang didapat dari tahun 2017 yang menyatakan bahwa penjualan hanya berkisar 150 reservasi, setelah ditepakan penjualan secara online ditahun yang sama selama 8 hari kerja penjualan meningkat berkisar 850 kursi, hal ini juga diterapkan ditahun 2018 dengan rentan waktu kerja selama 4 hari dengan menghasilkan kisaran 950 kursi, dengan adanya hal ini membuktikan bahwa melakukan penjualan tiket secara online sangatlah efektif (Hilman & Nugroho, 2020).

BOOKING TIKET ONLINE
Festival Nasional Reyog Ponorogo XXV

PLATINUM 130
-1 Tiket
-1 Majalah
-1 Gantungan Kunci
-1 Kaos
-1 Soft Drink
-1 Paper bag

GOLD 90
-1 Tiket
-1 Gantungan Kunci
-1 Paper bag
-1 Kaos
-1 Soft Drink

SILVER 65
-1 Tiket
-1 Majalah
-1 Gantungan Kunci
-1 Soft Drink
-1 Paper bag

BRONZE Couple 45
-2 Tiket
-2 Gantungan Kunci
-2 Soft Drink

BRONZE Group 60
-4 Tiket
-4 Soft Drink

RESERVASI
ALFIN - 082234638866 | SHOFIA - 085231207475

Grebeg Suro Mobile Apps
GET IT ON Google Play

@grebegsuroponorogo | www.grebegsuroponorogo.com | Grebeg Suro Ponorogo

Gambar 2. 1 Contoh penerapan *E-Ticketing*

Sumber: (Hilman & Nugroho, 2020)

Begitupun dalam penyebaran perekrutan *volunteer* yang dilakukan pada sistem yang telah dirancang, metode ini sangat efektif hal ini terbukti dengan adanya penilaian responden dan hasil wawancara yang telah dilakukan. *Volunteer* menjadi bagian yang sangat penting dalam penyelenggaraan suatu *event*, dengan melakukan penyebaran informasi perekrutan *volunteer* secara online akan meningkatkan partisipasi masyarakat dalam mengikuti festival ini (Hilman & Nugroho, 2020).



Gambar 2. 2 Contoh penerapan *E-Volunteer*
Sumber: (Hilman & Nugroho, 2020)

Dengan demikian, mengaitkan *e-ticketing* dan peran *e-volunteer*, memastikan bahwa kedua konsep tersebut terintegrasi dalam proses keberhasilan suatu *event*, dengan membangun *software* sebagai media penjualan tiket dan penyebarana informasi perkrutan *volunteer* dapat dipastikan bahwa implementasi dan pengelolaan yang efisien dan efektif dari platform *e-ticketing* dan sistem manajemen *volunteer* akan berjalan dengan baik dan dapat meningkatkan hasil penjualan secara optimal.

2.1.2. Teori Khusus

1. Metode *Design Thinking*

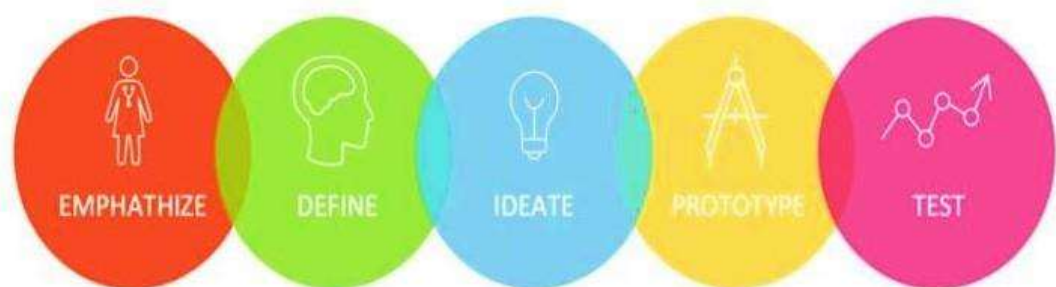
Metode *design thinking* sangat penting dalam suatu perancangan sistem, dengan menggunakan metode ini dapat mempermudah dalam memecahkan masalah dalam sistem yang mengakibatkan *website* tidak *user-friendly* sehingga tidak adanya interaksi antara *website* dan juga *user* (Ryan & Prahartiwi, 2023).

Design thinking melakukan pendekatan yang berguna untuk menyelesaikan sebuah masalah, pendekatan ini sangat bermanfaat ketika menghadapi masalah yang rumit atau tidak terdefiniskan dengan jelas. *Design thinking* dimanfaatkan untuk memahami masalah manusia secara kompleks. Dengan belum adanya penerapan media atau wadah yang biasa diakses oleh masyarakat untuk mencari informasi sekolah terkait pendaftaran, dan identitas sekolah oleh karena itu dirancanglah sebuah *website* sebagai tempat pencarian informasi sekolah SMKN 1 Tambang, pada *website* ini akan terdapat beberapa fitur unggulan yang sangat membantu masyarakat dalam mencari informasi yang diinginkan yaitu fitur *home*, tentang, jurusan, dan informasi (Ikhsandi et al., 2022).

Perusahaan Crowde yang bergerak dibidang keuangan, menghubungkan antara petani dengan perusahaan dengan tujuan pemodalana, dengan kemajuan dan keberhasilan perusahaan tentu saja Crowde berharap bisa menjangkau lebih banyak golongan bukan hanya dari golongan petani, *website* yang mereka miliki hanya mampu menjangkau para golongan petani dan mempunyai fitur terbatas yang tidak bisa diakses oleh golongan lain oleh karena itu dilakukan penelitian untuk menjawab masalah yang sedang dihadapi oleh *website* Crowde, maka dilakukanlah perancangan ulang pada *design thinking website* ini untuk mendapatkan hasil dan solusi dalam masalah ini untuk menciptakan *website* dengan fitur yang mendukung untuk kemajuan perusahaan, *design thinking* menjadi alur perencanaan dalam perancangan *website* ini (Arisa et al., 2023)

Dalam pengembangan *User Interface* (UI) dan *User Experience* (UX) *design thinking* mampu mengubah relevansi *design* serta kepuasan *user* secara efektif. Meskipun memiliki dampak dalam penerapana, tetapi *User Interface* (UI) dan *User Experience* (UX) mendapatkan peningkatan kualitas dan efektifitas setelah penerapan *design thinking* dalam proses pengembangannya hal ini terjadi karena adanya pendekatan mendalam tentang kebutuhan *user* yang diintegrasikan secara keseluruhan dalam proses pengembangan sistem (Rachman & Sutopo, 2023).

Berikut ini yang bisa dilakukan dalam tahapan perancangan dengan menggunakan metode *design thinking*:



Gambar 2. 3 Tahapan *Design Thinking*
Sumber: (Mohammad et al., 2024)

a. Memahami pengguna / *Empathize*

Design thinking mengutamakan pendekatan kepada *user* sehingga menghasilkan ide-ide dan solusi yang tepat dalam memecahkan masalah yang dialami oleh *user* salah satu tahapan dalam metode *design thinking* yaitu *emphatize* yang merupakan tahapan pengumpulan masalah yang sedang dialami dengan cara berempati terhadap *user*, proses ini diulang secara

berkala dengan tujuan untuk menemukan ide dan solusi yang tepat guna mengatasi masalah yang dihadapi *user* (Amin & Pasha, 2023).

b. Definisi Masalah / *Define*

Dalam definisi masalah *design thinking* menggunakan tahapan *define*, tahapan ini merupakan keberlanjutan dari tahapan *emphatize*, penjabaran hasil dari *emphatize* akan dilakukan sehingga akan mendapatkan hasil pemecahan masalah yang paling mendekati hal ini akan mempermudah dan lebih efektif dalam menemukan solusi yang inovatif serta relevan (Novita Kurnia Ningrum et al., 2022).

c. Pemikiran Kreatif / *Ideate*

Dengan mempertimbangkan berbagai ide yang sudah ada maka akan tercipta pemikiran yang lebih kreatif untuk menemukan solusi yang terbaik, *design thinking* mendorong pemikiran kreatif untuk mempertimbangkan berbagai ide dan pendekatan yang mungkin sulit bahkan tidak mungkin bisa dilakukan ditahapan pertama (Herfandi et al., 2022).

d. Prototyping / *Prototype*

Dalam *design thinking* memiliki tahapan *prototype*, tahapan ini sangat penting dalam perancangan awal suatu produk-produk aplikasi yang akan diciptakan, tahapan ini akan mengurangi kesalahan dalam perancangan karena akan dilakukan uji coba tahapan pertama kepada *user* yang bertujuan untuk memperoleh respon dari *user* dan berharap akan ada *feedback* untuk menyempurnakan rancangan agar sesuai dengan harapan *user* (Novita Kurnia Ningrum et al., 2022).

e. Pengujian dan Iterasi / *Test*

Pengujian dilakukan untuk mengumpulkan umpan balik *user* dari berbagai rancangan akhir yang telah dibuat selama proses *Prototipe* sebelumnya. Proses ini merupakan tahap akhir, tetapi juga merupakan proses hidup, yang memungkinkan kembali ke tahap perancangan sebelumnya jika terjadi kesalahan (Indah Pratiwi & Rani, 2023).

2. Metode *Software Development Live Cycle* (SDLC)

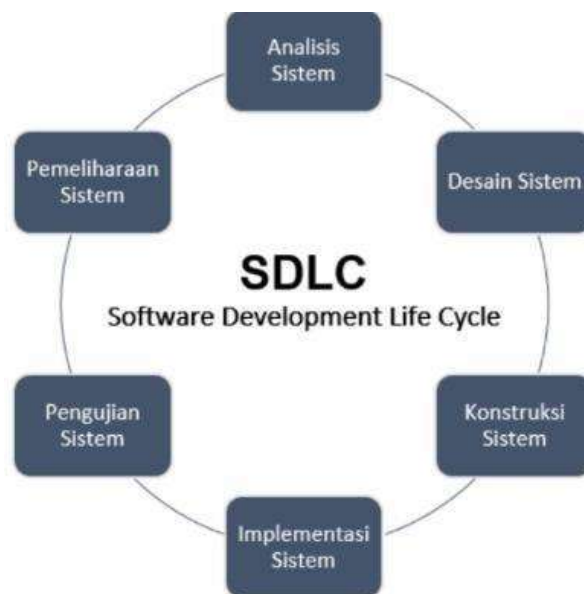
Software Development Live Cycle (SDLC) mampu memberika panduan dan langkah langkah dalam proses pengerjaan dalam proses perancangan serta pengembangan *software*, SDLC menggunakan tahapan yang harus diselesaikan secara bertahap, hal ini akan mengoptimalkan dan mengurangi kesalahan dari setiap tahapan artinya fokus kepada setiap masing-masing tahapan dapat dilakukan secara maksimal karena tidak adanya pekerjaan yang dikukan secara pararel (Puspa & Wulandari, 2021).

SDLC menjadi metode yang sangat penting dalam *software development*, SDLC mempunyai tujuan dalam merepresentasikan proses yaitu dilalui seperti perencanaan, membuat, menguji, serta penyebaran. SDLC menjadi metode dalam perencanaan ERP menu Hr-Training yang berbasis pada Odoo. Pengelolaan sumber daya manusia dalam perusahaan hal ini membutuhkan pengelolaan pelatihan pada karyawan, dengan banyaknya sumber daya manusia serta materi pelatihan membuat perusahaan membutuhkan rancangan manajemen ERP untuk mengelola bentuk pelatihan ini. Dengan menggunakan metode SDLC akan menentukan struktur dan

organisasi lebih jelas, perencanaan yang matang, kualitas serta pengujian yang akan tetap meningkat hal ini akan menciptakan sebuah sistem informasi HR-Training yang berkualitas (Jamal & Kusnadi, 2022)

Perancangan *website* menggunakan metode SDLC sangat membantu dalam perancangan *website*, perancangan sistem informasi pengelolaan data untuk mengelola informasi tentang kegiatan dan operasional yayasan yang berbasis *website*. Sistem informasi ini juga akan digunakan dalam pengelolaan data seperti data pasien, dan data keuangan. Dengan menggunakan metode SDLC sebagai tahapan awal sebelum perancangan akan mempermudah dalam menentukan kebutuhan secara detail serta fitur-fitur apa saja yang diperlukan *software* nantinya oleh karena itu SDLC sangat penting dalam perancangan sistem informasi (Widharma et al., 2023).

Software Development Live Cycle (SDLC) menjadi metodologi dalam perancangan sistem informasi afiliasi penjualan tiket seminar yang berbasis *website*, metodologi *Software Development Live Cycle* (SDLC) ini digunakan karna mampu memberikan struktur yang jelas, perancangan yang lebih baik, manajemen proyek yang efektif dan dapat mengendalikan kualitas dengan lebih baik. Hal ini terbukti dalam perancangan *website* penjualan yang bisa membantu para penyelenggara acara untuk mempermudah pekerjaan dalam penjualan, pengolahan data, serta pendaftaran peserta, *Software Development Live Cycle* (SDLC) menjadi panduan dalam perancangan sistem informasi serta menjadi wadah promosi yang efektif bagi penyelenggara acara (Pratama & Paramita, 2020).



Gambar 2. 4 Tahapan *software development live cycle* (SDLC)
Sumber: (Nazaruddin et al., 2022)

Membangun *software* dengan menggunakan metode SDLC sebagai perancangan dan mengembangkan akan sangat mudah, proses perancangan akan meliputi 6 tahapan yaitu:

a. *Requirement* / analisis kebutuhan

Analisis kebutuhan merupakan tahapan pertama yang akan dilakukan dalam proses perancangan *software*, pada tahapan ini sangat penting karena akan mengurangi resiko kegagalan, menghemat waktu dan biaya, memahami kebutuhan akan *user*, menentukan ruang lingkup *software* dan mendukung keputusan desain yang akan dibuat, dengan memperhatikan *Requirement* ditahapan pertama ini akan mempermudah untuk melangkah ke tahapan selanjutnya.

Pada tahapan *Requirement* dilakukan analisis terhadap masalah pemesanan tiket konser yang masih secara manual yang menyebabkan

pemesanan tiket dan pelayanan kurang baik karena *human error*. Dengan dilakukan tahapan analisis dirancang sebuah *website* pemesanan tiket konser musik yang akan mempermudah pembelian dan pelayanan (Supena et al., 2021).

b. *Design* / perancangan

Tahapan ini menjadi kunci dalam membangun *software*, dengan melakukan *Design* maka akan menciptakan struktur, arsitektur dan rincian implementasi suatu *software* yang akan dikembangkan dan ditetapkan.

Tahapan *Design* dengan menggambarkan *Unified Modelling Language* (UML) sebagai perancangan sistem *software*, *Entity Relationship Diagram* (ERD) dan *Logical Record Structure* (LRS) untuk perancangan *database*. Dengan melakukan perancangan ini maka akan mempermudah dalam melakukan perkembangan ke tahap selanjutnya (Erawati et al., 2023).

c. *Development* / pengkodean

Dalam tahapan pengkodean diawal ini akan dimulainya proses pengembangan atau implementasi dari *design* yang telah ditetapkan dan disepakati berdasarkan analisis dan kebutuhan dari *user*, dengan demikian melakukan pengkodean diawal ini akan mengubah kosep dari *design* menjadi *software* yang nyata dan menciptakan proses pengembangan secara menyeluruh hal ini juga akan memungkinkan untuk mendapatkan umpan balik dari *stakeholder* (Paksi et al., 2023).

d. *Testing* / percobaan

Tahapan *testing* memiliki banyak kelebihannya seperti melakukan peningkatan kualitas pada *software* dengan melakukan *testing* atau percobaan ini maka akan ditemukan kesalahan yang bisa diperbaiki sebelum *software* diliriskan untuk pengguna akhir (Rachman & Sutopo, 2023).

Dengan melakukan percobaan secara keseluruhan maka akan membantu untuk menemukan kekurangan atau kesalahan dalam beberapa kinerja atau fungsional pada *software* sebelum terjadi kesalahan yang besar dimasa yang akan datang. Dengan adanya tahapan *testing* memungkinkan untuk memastikan bahwa *software* memenuhi persyaratan dalam fungsional ataupun non fungsional yang telah ditetapkan sehingga akan tercipta kepuasan terhadap *user* dan mengurangi tingkat ketidak cocokan dalam menggunakan *software* ini sebagai kebutuhan *user*, dengan adanya tahapan ini juga akan mempersingkat waktu dan biaya di awal (Soedewi, 2022).

e. *Development* atau pengkodean

Pengkodean kedua meningkatkan kualitas *software* secara keseluruhan dengan memperbaiki dan menyempurnakan kode yang sudah ada. Dengan melakukan *refactoring*, pengembang dapat menghilangkan duplikasi kode, mengurangi kompleksitas, dan meningkatkan kejelasan kode, sehingga menjadi lebih mudah dipahami dan dikelola dimasa depan (Parhusip et al., 2023).

f. *Maintenance* atau perbaikan

Dalam tahapan ini akan dilakukan perbaikan dalam setiap permintaan *user*, dan seiring berjalannya waktu akan ada proses perbaikan mulai dari

keamanan, keberlangsungan kualitas *software* serta penambahan permintaan *user*. Dengan melakukan *Maintenance* secara teratur dan berkala maka akan mengoptimalkan kinerja pada *software* dan dapat diandalkan dalam jangka waktu yang cukup lama sehingga *software* akan tetap bisa digunakan dalam jangka waktu yang cukup panjang hal ini akan menciptakan kepuasan terhadap *user* (Cahya et al., 2021).

3. *Database*

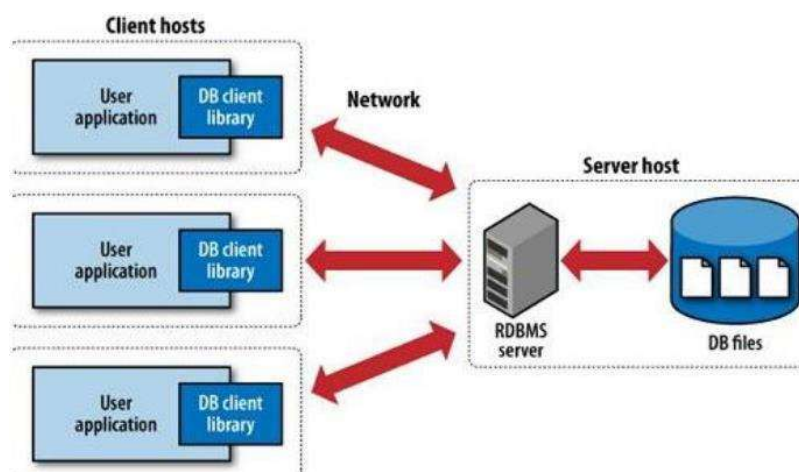
Pengolahan *database* diperlukan oleh organisasi perusahaan untuk mempercepat akses ke informasi dan peningkatan layanan. Penggunaan *database* sangat diperlukan untuk mempermudah melakukan aktifitas sehari-hari maupun untuk meringankan pekerjaan (Primadewi & Widyanto, 2022).

Database sebagai wadah pengelola data-data akademik pada universitas sebagai pengelolaan pendaftaran, pengelolaan sistem pembelajaran, pengelolaan data mahasiswa dan dosen. Pengelolaan data ini dilakukan pada aplikasi maupun *website* dikelola berdasarkan data pribadi yang langsung diinput saat mendaftar dan merancang pembelajaran yang telah ditetapkan (Primadewi & Widyanto, 2022).

Pada jasa keuangan *database* dapat menampung data para nasabah, data peminjaman, data penarikan, data tabungan dan data angsuran. Sehingga nasabah lebih mudah dalam mengakses sistem internet *banking*, ATM, dan *m-banking* (Primadewi & Widyanto, 2022).

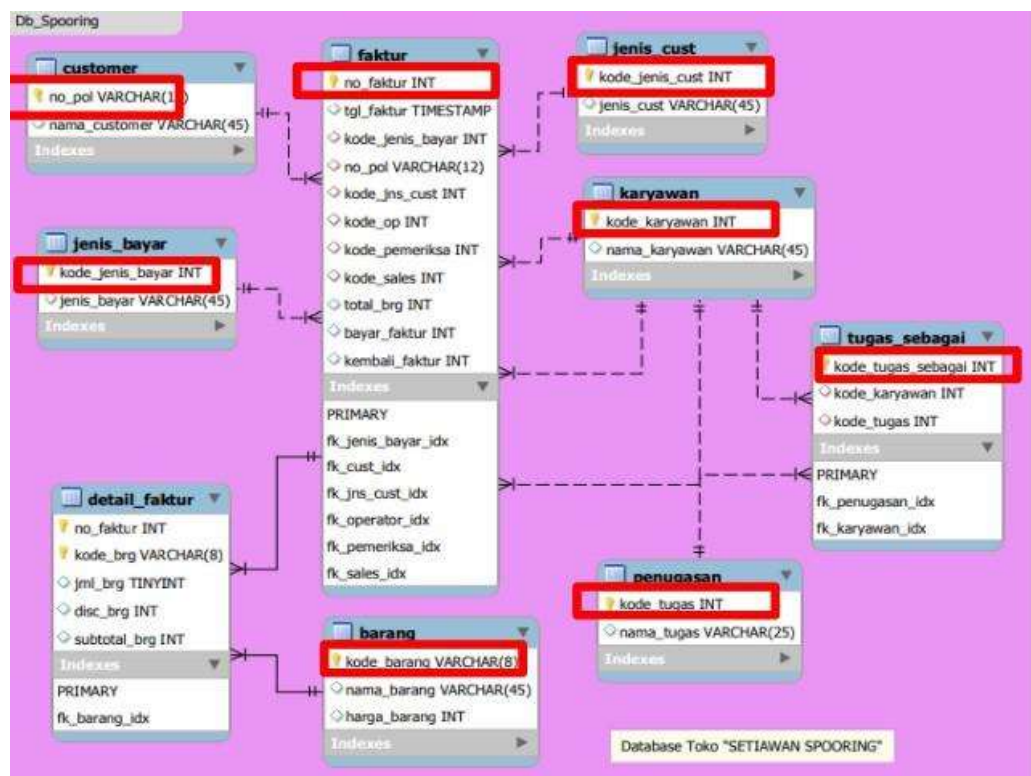
Dalam perancangan sistem penjualan tiket baik secara *mobile* maupun *website*, *database* menjadi platform penyimpanan data-data yang masuk seperti data nama, nomor NIK, tanggal lahir, nomor telepon, email dan masih banyak lagi (Primadewi & Widyanto, 2022).

Departemen *database* merupakan data yang dikumpulkan secara berkelompok. Sebagaimana pada gambar 2.5 yang menunjukkan bahwa data-data yang sudah ada akan dikelompokkan kedalam satu *database* file dan diberikan *key* sebelum diproses kedalam server. Untuk mempermudah pengaksesan oleh pihak manajemen maka *database* membutuhkan RDBM server sebagai tempat penyimpanan, pengolahan data, pengaturan data-data yang ada dan dengan adanya RDBMS server akan mempermudah dalam pengaksesan data oleh *client hosts* hanya dengan menggunakan jaringan, tentu saja *database* tidak hanya bisa diakses oleh *user* akan tetapi juga bisa diakses oleh berbagai *user* pengguna aplikasi (Primadewi & Widyanto, 2022).



Gambar 2. 5 Visualisasi departemen database
Sumber: (Primadewi & Widyanto, 2022)

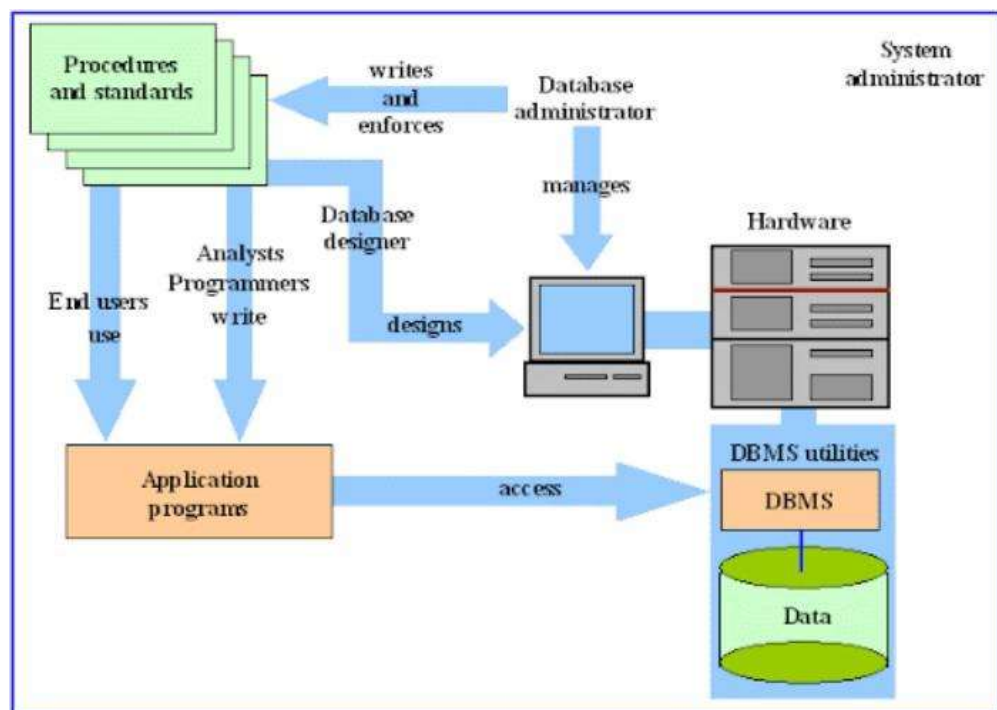
Dalam penyimpanan data pada database, setiap data yang ada akan diberikan primary key sebagai kunci utama yang berfungsi sebagai unik identifikasi sebagaimana contoh pada gambar 2.6 yang memiliki primary key pada setiap tabel hal ini akan mempermudah identifikasi pada setiap tabel, contohnya pada tabel faktur yang memiliki banyak identitas tapi hanya mempunyai satu primary key yaitu no_faktur int yang berarti dalam tabel faktur ini hanya memiliki satu no faktur, begitupun pada tabel tabel yang lain (Primadewi & Widyanto, 2022).



Gambar 2. 6 Visualisasi proses data menjadi informasi
Sumber: (Primadewi & Widyanto, 2022)

Sebagaimana contoh proses *database* pada gambar 2.7 yang menggambarkan alur dari penyimpanan data-data diawal sampai proses akses data oleh program aplikasi, dimana seorang administrasi *database* tidak

hanya memiliki kemampuan untuk melakukan pengolahan data tetapi mereka juga dapat mengubah standar dan prosedur sistem untuk memenuhi kebutuhan *user*. Sehingga *database* menjadi komponen penting dalam sebuah sistem yang harus di pahami oleh *user*. Dengan adanya primary key yang ditetapkan pada setiap data akan mempermudah dalam proses pengolahan data yang dilakukan oleh *user* (Primadewi & Widyanto, 2022).



Gambar 2. 7 Visualisasi Sistem Administrasi
Sumber: (Primadewi & Widyanto, 2022)

4. *Unified Modeling Language (UML)*

Untuk mengembangkan teknologi *software*, diperlukan pemodelan *software* untuk standarisasi sehingga orang diseluruh dunia dapat memahami model tersebut. Bahasa pemodelan *software* yang dapat dipahami oleh banyak orang telah dibuat oleh banyak pihak sesuai dengan teknologi

pemrograman saat ini. Misalnya *state transition diagram* (STD) digunakan untuk memodelkan sistem *real-time* dan *data flow diagram* (DFD) digunakan untuk memodelkan *software* yang menggunakan prosedur pemrograman atau struktural. Bahasa pemodelan *unified modeling language* (UML) merupakan standar untuk pengembangan *software* yang menggunakan teknik pemrograman berorientasi objek karena kebutuhan akan pemodelan visual untuk menspesifikasikan, membangun, menggambarkan, dan mendokumentasikan sistem *software*. UML merupakan bahasa visual yang dirancang untuk memodelkan dan berkomunikasi tentang sistem dengan menggunakan diagram dan teks pendukung. UML hanya digunakan untuk pemodelan, sehingga tidak terbatas pada metodologi tertentu. Namun biasanya UML digunakan untuk metodologi yang berorientasi pada objek (ROSA & SHALAHUDDIN, 2019).

Banyak sekali kendala dalam melakukan promosi sehingga banyak masyarakat yang tidak mendapatkan informasi seputar *event* yang akan diselenggarakan, dengan adanya sistem informasi yang membantu dalam melakukan promosi secara online dan masyarakat lebih mudah mendapatkan informasi, perancangan sistem informasi menggunakan beberapa diagram seperti *use case diagram*, *activity diagram*, *sequence diagram*, dan *class diagram* akan sangat membantu dalam pemodelan *software*, perancangan sistem informasi ini akan menghasilkan sebuah *website* yang bermanfaat bagi masyarakat (R. E. Nugroho, 2019).



Dalam konteks jasa memerlukan *Unified Modeling Language* (UML) sebagai pemodelan visual, sehingga mempermudah perancang dalam memahami kebutuhan *user* dan merancang solusi yang efektif (Sanjaya et al., 2022).




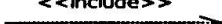

a. *Use Case Diagram*

Use case merupakan representasi suatu fungsionalitas sistem yang bisa memberikan gambaran pada pelanggan atau *user* tentang manfaat atau kegunaan dari sistem yang akan dibangun. *Use case diagram* sangat membantu dalam penyusunan analisis kebutuhan untuk membangun sistem, diagram ini digunakan untuk mengkomunikasikan suatu rancangan sistem dengan *user* bukan hanya itu diagram ini juga berguna untuk melakukan perancangan tes *case* pada semua fitur fitur yang nantinya akan ada pada *website* atau aplikasi (Hasanah, 2020).

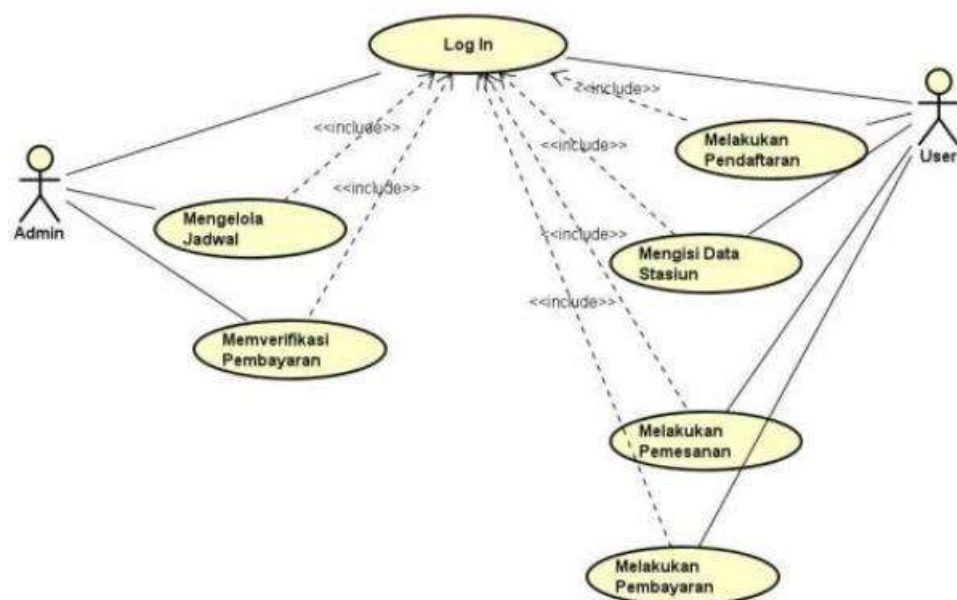
Dalam perancangan *use case diagram* ada beberapa simbol yang digunakan untuk menandai atau mewakili proses pada sebuah diagram *use case* dimana simbol yang digunakan sebagai berikut:

Tabel 2. 1 Komponen-komponen dasar *use case diagram*

Simbol	Keterangan
	Use case : Abstraksi dari interaksi antara sistem dan aktor
	Aktor : mewakili peran orang, sistem yang lain atau alat ketika berkomunikasi dengan use case

<p>Asosiasi / <i>association</i></p> 	<p><i>Association</i> adalah abstraksi dari penghubung antara aktor dan <i>use case</i></p>
<p><<extend>></p> 	<p>Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi</p>
<p>Generalisasi / <i>generalization</i></p> 	<p>Generalisasi : menunjukkan spesialisasi aktor untuk dapat berpartisipasi dalam <i>use case</i></p>
<p><<include>></p>  <p>«uses»</p> 	<p>Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya</p>

Sumber: (ROSA & SHALAHUDDIN, 2019)



Gambar 2. 8 Contoh *use case diagram* pada aplikasi pembelian tiket kereta api




Sumber: (Basri et al., 2022)



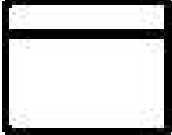
Terlihat pada gambar 2.8 contoh model *use case* pada perancangan aplikasi pembelian tiket kereta api (Basri et al., 2022).

b. *Activity Diagram*

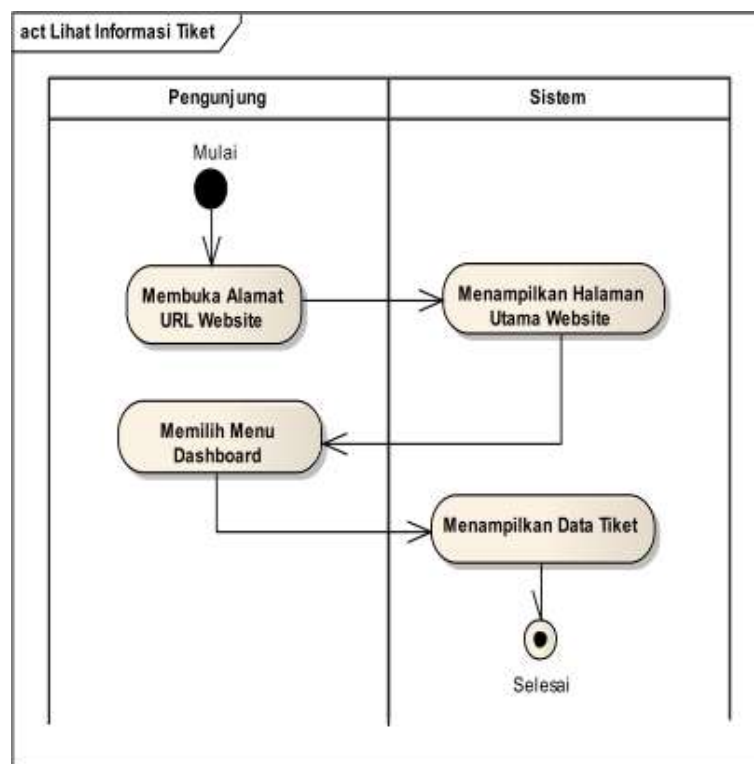
Activity diagram mengilustrasikan berbagai aliran aktivitas pada sistem yang sedang dirancang mulai dari aliran pertama. *Activity diagram* merupakan *state diagram* khusus karena bagian dari state ini merupakan *action* dan sebagian besar transisi dipicu oleh selesainya *internal processing*. *Activity diagram* tidak secara lengkap menggambarkan tentang perilaku internal sistem dan interaksi antar subsistem, melainkan lebih berfokus pada proses-proses dan jalur-jalur aktifitas secara umum dari level atas (Hasanah, 2020).

Tabel 2. 2 Komponen-komponen dasar *activity diagram*

Simbol	Keterangan
Status Awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja asosiasi
Percabangan / <i>Decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.

Penggabungan / Join 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status Akhir 	status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktifitas yang terjadi.

Sumber: (ROSA & SHALAHUDDIN, 2019)



Gambar 2. 9 Contoh rancangan *activity diagram* pada pembelian tiket wisata

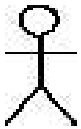

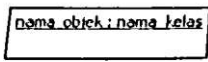
Sumber: (Hermanto, 2021)



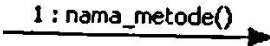
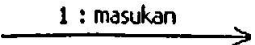

Terlihat pada gambar 2.9 contoh rancangan model *Activity Diagram* pada kasus aplikasi pembelian tiket wisata (Hermanto, 2021).

c. *Sequence Diagram*

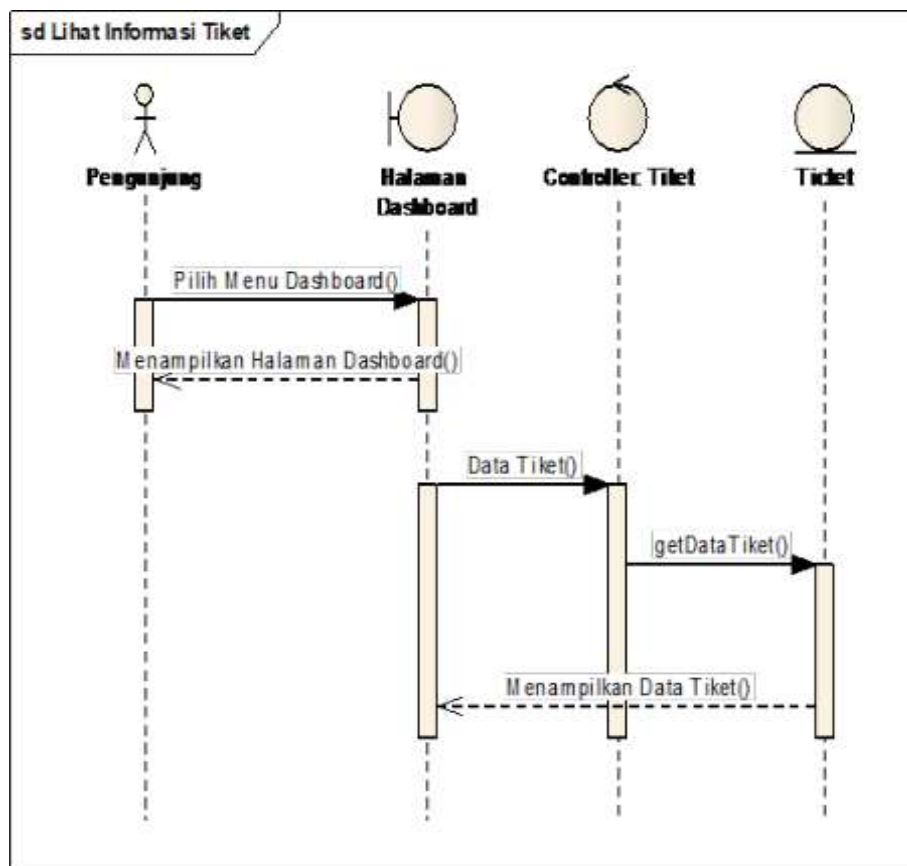
Sequence diagram menggambarkan perilaku objek dalam suatu *use case* dengan menggambarkan waktu hidup objek dan pesan-pesan yang dikirim dan diterima antar objek, untuk membuat *sequence diagram* penting untuk mengetahui objek-objek yang terlibat dalam suatu *use case* beserta metode-metode yang dimiliki oleh kelas yang diinstansiasi menjadi objek tersebut. Pembuatan *sequence diagram* juga diperlukan untuk melihat skenario-skenario yang ada dalam *use case* tersebut (ROSA & SHALAHUDDIN, 2019).

Tabel 2. 3 Komponen-komponen dasar *sequence diagram*

Simbol	Keterangan
 <p>Aktor</p>	Menggambarkan orang yang sedang berinteraksi dengan sistem.
<p><i>Life Line</i></p> 	Menyatakan kehidupan suatu objek.
<p>Objek</p> 	Menyatakan objek yang berinteraksi pesan.

<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.</p>
<p>Pesan Tipe Create</p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.</p>
<p>Pesan tipe call</p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri, Arah panah mengarah pada objek yang memiliki operasi / metode, karena ini memanggil operasi / metode maka operasi / metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.</p>
<p>Pesan tipe send</p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.</p>
<p>Pesan tipe return</p> 	<p>Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.</p>

Sumber: (ROSA & SHALAHUDDIN, 2019)



Gambar 2. 10 Contoh perancangan *sequence diagram* pada kasus pemesanan tiket event

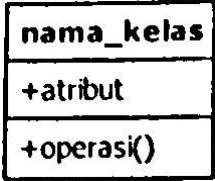



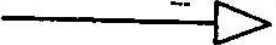
Sumber: (Hermanto, 2021)

Terlihat pada gambar 2.10 contoh rancangan model *Sequence Diagram* pada aplikasi pemesanan tiket event di kota Jakarta (Hermanto, 2021).

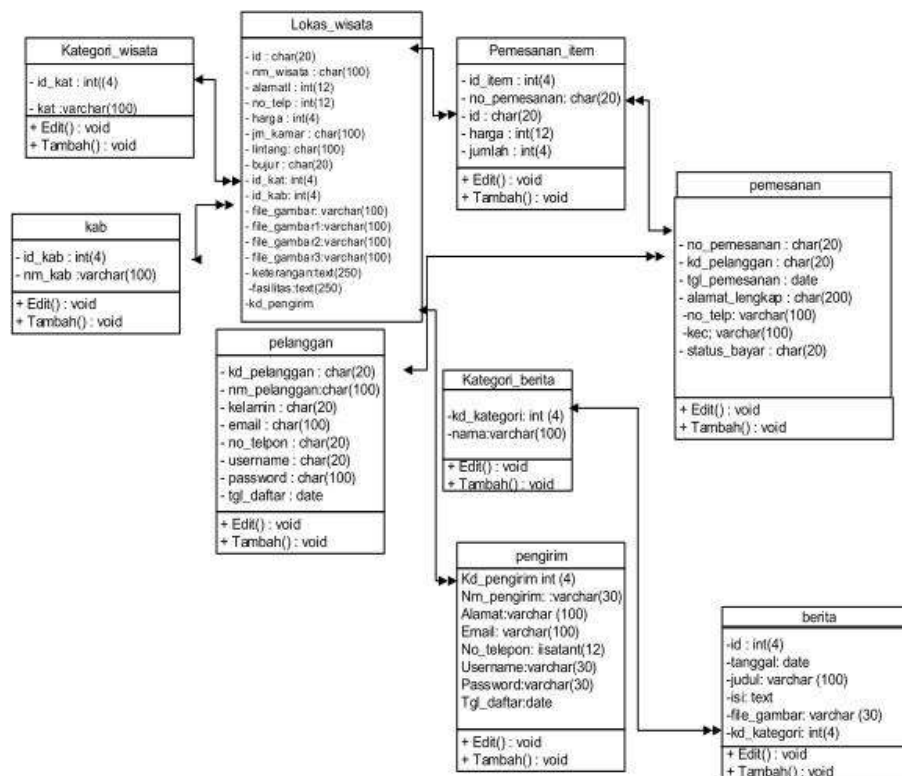
d. *Class Diagram*

Diagram kelas menggambarkan struktur sistem yang saling berhubungan dengan cara menjelaskan inti dari keterhubungan antar tabel hal ini berhubungan erat dengan data-data yang tersimpan pada *database* (Susianto, 2019).

Tabel 2. 4 Komponen-komponen dasar class diagram

Simbol	Keterangan
	Kelas pada struktur sistem
	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
<p>asosiasi / <i>association</i></p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
<p>asosiasi berarah / <i>directed association</i></p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
<p>generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)

Sumber: (ROSA & SHALAHUDDIN, 2019)



Gambar 2. 11 Contoh perancangan *class diagram* pada kasus pemesanan E-Tiket
Sumber: (Susianto, 2019)

Terlihat pada gambar 2.11 contoh perancangan model *Class Diagram* pada kasus pemesanan e-tiket di Lampung (Susianto, 2019).

2.2. Penelitian Terdahulu

Tabel 2. 5 Penelitian terdahulu

No	Judul Artikel	Penulis	Vol, No, dan Tahun	Nama Jurnal dan Akreditas	keterangan
1.	Rancang Bangun Sistem Informasi Afiliasi Penjualan Tiket Seminar Berbasis <i>Website Framework Laravel</i>	Dimas Kurnia Pratama, Adi Suryaputra Paramita	Vol 7, No1, Tahun 2020	Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK), Sinta 2	Dalam kegiatan penjualan tiket membutuhkan wadah sebagai media penjualan, promosi serta penghitung afiliasi penjualan tiket pada

					<p>perusahaan. jumlah pendaftaran, tingkat konversi perusahaan, kepuasan <i>user</i> menjadi hal utama dalam keberhasilan sistem yang akan dibangun. Metode yang digunakan <i>waterfall</i> serta (<i>Unified Modelling Language</i>) UML sebagai pemodelan <i>software</i>. MySQL dan framework laravel sebagai database dalam sistem ini sehingga akan menghasilkan sebuah <i>website</i> yang berfungsi sebagai wadah penyebaran informasi seminar, penjualan tiket seminar dan penghitungan afiliasi penjualan.</p>
2.	Implementasi Tiket Film Berbasis Web Yang Efisien Sistem Pembelian Dalam Konteks Bangladesh	Gazi Zahirul Islam, Isrut Jahan Zinnia, Md. Fokhray Hossain, Md. Riazur Rahman, Aman Ullah Juman, Al Nahian Bin Emran	Jil. 19, No.3, Tahun 2020	Jurnal Teknik Elektro dan Ilmu Komputer Indonesia, Sinta 1	Karena belum adanya sistem pembelian tiket yang memiliki fitur pembelian tiket dari berbagai bioskop di Banglades sehingga sulitnya untuk melakukan pembelian tiket

					<p>pada bioskop yang diinginkan. Banyaknya masyarakat yang menggunakan sistem ini akan mempengaruhi keberhasilan dari proyek ini. Pemodelan yang digunakan dalam proyek ini yaitu <i>Unified Modelling Language</i> (UML). <i>Javascript</i> dan HTML sebagai <i>tools</i> yang akan digunakan sehingga hasil dari proyek ini akan menghasilkan <i>website</i> penjualan tiket yang memiliki fitur yang lengkap dan sangat bermanfaat bagi masyarakat.</p>
3.	Aplikasi Penjualan Tiket Kelas Pelatihan Berbasis <i>Mobile</i> menggunakan <i>Flutter</i> .	Filian Enggar Krisnada, Radius Tanone	Vol 5, No 3, Tahun 2019	Jurnal Teknik Informatika dan Sistem Informasi, Sinta 3	<p>Pengembangan aplikasi penjualan tiket kelas dalam pelatihan yang berbasis <i>mobile</i> menggunakan fitur <i>flutter</i> yang bertujuan untuk mempermudah <i>user</i> dalam mencari dan mendaftar kelas pelatihan pada sistem ini mempunyai <i>collaborative filtering</i> yang</p>

					<p>berguna untuk membantu para pengguna mencari dan menemukan kelas yang menarik. Tiket kelas pelatihan menjadi indikator dari sistem ini. Sistem ini menggunakan MySQL sebagai <i>local database</i>. Adapun <i>tools</i> yang digunakan dalam perancangan yaitu <i>Framework Flutter</i> sebagai pengembangan aplikasi lintas platform.</p>
4.	Perancangan <i>Website Sistem Informasi Ticket Reporting</i>	Sudarsono Aritonang, Fajrin Nurhakim, Aziz Wisnu Wardana, Yogo Dwi Prasetyo	Vol 10, No 1, Tahun 2023	JURIKOM (Jurnal Riset Komputer), Sinta 4	<p>Objek wisata sangat sangat ramai dikunjungi terutama saat liburan, pengelolaan parawisata harus mempersiapkan berbagai kebutuhan dari para pengunjung begitupun pada pelayanan. Pelaporan hasil penjualan tiket menjadi masalah yang besar jika masih dilakukan secara manual oleh karena itu diperlukan sebuah sistem yang bias mengelola penjualan secara</p>

					digital dan efektif. Oleh karena itu akan dirancang sebuah sistem dengan menggunakan metode <i>waterfall</i> dan HTML serta MySQL sebagai <i>tools</i> yang sangat penting sehingga akan menghasilkan sebuah <i>website</i> yang berfungsi sebagai pengelola laporan hasil penjualan tiket yang efektif.
5.	Rancang Bangun Sistem Informasi Akademik Berbasis Website Dengan Metode SDLC	Wati Erawati1, Sujiliani Heristian, Rachmat Adi Purnama	Vol 3, No 2, Tahun 2023	Computer Science (CO-SCIENCE), Sinta 4	Proses penginputan data yang dilakukan secara manual akan mempersulit pekerjaan dosen, pegawai dan orang tua dengan membangun sistem informasi maka akan mempermudah proses dalam pelayanan informasi sekolah dan juga mempermudah pekerjaan, sistem yang akan dibangun menggunakan beberapa <i>tools</i> penting yaitu <i>software development life cycle</i> (SDLC), dengan metode

					<i>waterfall</i> dan juga UML sebagai pemodelan data, sehingga menghasilkan <i>website</i> sebagai wadah informasi akademik sekolah.
6.	The Effectiveness Of Online Ticket Management By Volunteer In Grebeg Suro 2018 And Reyog Ponorogo National Festival xxv	Yusuf Adam Hilman, Irfan Nugroho	Vol 6, No 1, Tahun 2020	CosmoGov: Jurnal Ilmu Pemerintahan, Sinta 2	Dengan semakin meningkatnya parawisata yang diselenggarakan di Kabupaten Ponorogo, penyelenggara melakukan analisis untuk memperbaiki sistem penjualan agar pengelola lebih mudah dalam melakukan rekapan data penjualan dan perekrutan sukarelawan/ <i>volunteer</i> untuk membantu parawisata yang ingin berlibur di Kabupaten Ponorogo. Proses analisis pada sistem ini menggunakan metode deskriptif kualitatif serta pengujian dengan cara <i>cross check</i> sehingga nantinya akan menghasilkan sebuah analisis yang menguatkan penyelidik untuk

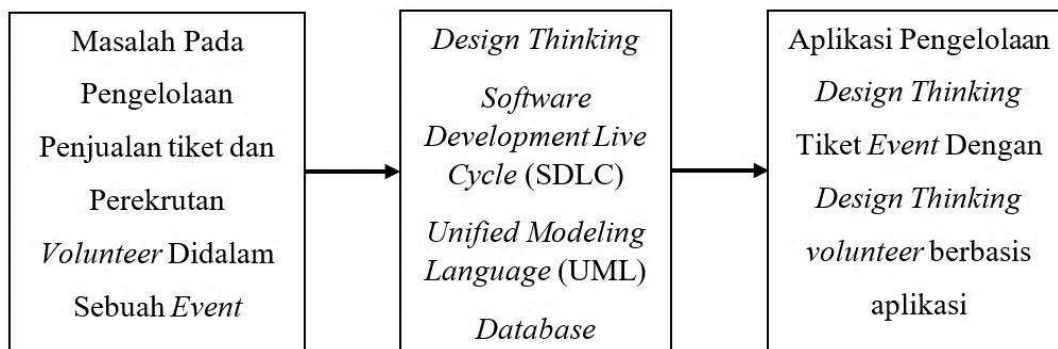
					melakukan perbaikan terhadap sistem yang ada.
7.	Penerapan Metode <i>Design Thinking</i> Dalam Pengembangan UI/UX: Tinjauan Literatur	Annisa Rachman, Joko Sutopo	Vol 9, No 2, Tahun 2023	SemanTIK, Sinta 4	Penelitian ini menyelidiki bagaimana konsep <i>design thinking</i> dapat diterima dalam UI/UX dengan memberikan penekanan khusus pada analisis fase-fase, metode pelaksanaan serta manfaat dan masalah yang harus dipecahkan, metode yang digunakan yaitu peninjauan literatur penelitian yang relevan dan membuat <i>research questions</i> sesuai dengan kebutuhan dalam penelitian serta <i>design thinking</i> sebagai pendekatan yang berpusat kepada <i>user</i> untuk mendapatkan solusi yang tepat. Indikator penelitian ini yaitu keberhasilan dalam sebuah <i>desing thinking</i> dan ouputnya peningkatan kinerja UI/IX semakin baik.

8.	Penerapan Metode Design Thinking dan Agile dalam Rancang Bangun Aplikasi Penjualanku	Zaid Amin, Nabila Pasha Fakultas	Vol 4, No 3, Tahun 2023	Journal of Information System Research (JOSH) , Sinta 4	Proses yang dilakukan dalam penelitian ini yaitu melakukan analisis permasalahan penjualan UMKM menggunakan <i>design thinking</i> dengan tujuan mendapatkan ide dan solusi dari permasalahan penjualan UMKM, dengan menggunakan beberapa <i>tools</i> pendukung seperti <i>design thinking, agile</i> dan <i>javaskrip</i> sebagai bahasa pemrograman sehingga menghasilkan aplikasi penjualan yang sangat bermanfaat bagi UMKM.
9.	Sistem Informasi Pemesanan Tiket Konser Musik <i>Theater</i> Berbasis <i>Website</i>	Khuzaipei, Melan Susanti, Mari Rahmawati	Vol 9, No 2, Tahun 2021	Bianglala Informatika, Sinta 5	Dengan kemajuan teknologi yang semakin pesat, pemasaran serta penyebaran informasi telah dilakukan secara digital begitupun dalam pemesanan tiket yang bisa dilakukan dimanapun membahas dengan membangun sistem pemesanan

					<p>tiket secara online akan membantu masyarakat dan mempermudah <i>user</i> dalam melakukan pembelian dan mendapatkan informasi tentang konser yang akan diselenggarakan. Oleh karena itu akan dibangun sebuah sistem dengan menggunakan <i>Unified Modelling Language</i> (UML) sebagai bahasa pemodelan standar, <i>software</i> dirancang dengan menggunakan metode <i>System Development Life Cycle</i> (SDLC), <i>tools</i> sistem menggunakan <i>unified modeling language</i> (UML) sehingga akan menghasilkan sebuah <i>website</i> sebagai wadah penyebaran informasi konser serta wadah pembelian tiket secara digital.</p>
10.	Analisis dan Pembuatan Aplikasi <i>E-commerce</i> Terintegrasi Pada Sistem B2B	Mufida Dewi Puspa, Lily Wulandari	Vol 9, No 1, Tahun 2021	JUTIS (Jurnal Teknik Informatika Unis) Vol., Sinta 5	Tujuan dari penelitian ini sebagai analisis terhadap produk penjualan <i>e-commerce</i> dan

	Dengan Pendekatan <i>System Development Life Cycle</i> (SDLC) Untuk Penjualan Produk Transformator			melakukan perancangan aplikasi <i>e-commerce</i> sebagai media penjualan yang unggul dan kompetitif. Dalam perancangan ini dibangun menggunakan <i>tools System Development Life Cycle</i> (SDLC) sehingga akan menghasilkan sebuah sistem aplikasi berbasis <i>website</i> .
--	--	--	--	---

2.3. Kerangka Pemikiran



Gambar 2. 12 Proses kerangka pemikiran
Sumber: (Data Penelitian, 2024)

Permasalahan pada penelitian ini dengan teridentifikasinya permasalahan pada penjualan tiket *event* yang berjalan secara manual, sehingga pihak pengelolaan penjualan tiket dan penyelenggaraan event

mengalami masalah, yang juga ditambah dengan sulitnya melakukan perekrutan volunteer pada event yang diselenggarakan.

Untuk memecahkan permasalahan tersebut penelitian ini akan menerapkan prinsip kerja metode *design thinking* yang berfokus bagaimana mengolah data terkait penjualan tiket dan perekrutan volunteer dan dengan memanfaatkan metode *software development live cycle* (SDLC) untuk perancangan aplikasi berbasis website, dengan dibantu permodelan arsitektur menggunakan *unified modeling language* (UML), serta database untuk menyimpan data yang dibutuhkan.

Penelitian ini nantinya akan menghasilkan aplikasi berbasis *website* dengan menerapkan konsep metode desain *thingking* untuk membantu penjualan tiket *event* dan perekrutan *voluntee*.