

BAB 2

TINJAUAN PUSTAKA

Dalam Bab ini berisi berbagai konsep dasar dan teori-teori yang berkaitan dengan pembangunan “Pengembangan Sistem Pencatatan Ternak Sapi Digital Berbasis QR Code Menggunakan Teknologi Blockchain”.

2.1 Pencatatan / *Recording* Ternak

Recording adalah suatu usaha yang dikerjakan oleh peternak untuk mencatat gagal atau berhasilnya suatu usaha peternakan. Di bidang usaha peternakan program ini diterapkan hampir pada semua sektor usaha ternak mulai ternak unggas (layer, broiler, penetasan), ternak potong (sapi perah, sapi potong, kambing dan domba), dan aneka ternak seperti kelinci dan lainnya [14].

Recording sangat bermanfaat dalam setiap kegiatan atau usaha apapun, untuk usaha ternak sapi, *Recording* bermanfaat untuk memudahkan pengenalan terhadap ternak dimana dengan mengetahui identitas dan ciri-ciri khusus ternak, serta mengetahui populasi ternak, memudahkan peternak mengingat kejadiankejadian penting pada ternaknya, memudahkan peternak mengambil keputusan ataupun tindakan nyata dalam penanganan, perawatan dan pengobatan pada ternak yang sakit berdasarkan catatan riwayat kesehatannya, memudahkan peternak melakukan seleksi ternak serta dapat mencegah terjadinya kawin sedarah atau inbreeding [15].

2.2 Basis Data

Menurut Indrajani “basis data adalah kumpulan data yang saling berhubungan secara logis dan didesain untuk mendapatkan data yang dibutuhkan oleh suatu organisasi”. Menurut Jayanti & Sumiari “Basis Data merupakan data yang terintegrasi, yang diorganisasi untuk memenuhi kebutuhan para pemakai di dalam suatu organisasi” [16].

2.3 Struktur Data

Struktur data adalah cara penyimpanan, pengorganisasian, dan pengaturan data di dalam media penyimpanan komputer sehingga data tersebut dapat digunakan secara efisien [17]. Sementara itu, menurut Hidayati struktur data adalah suatu koleksi atau kelompok data yang dapat dikarakteristikan oleh organisasi serta operasi yang didefinisikan terhadapnya. Artinya, struktur data adalah suatu koleksi data yang diorganisasikan sedemikian rupa agar dapat digunakan (dioperasikan) berdasarkan definisi yang diberikan pada struktur tersebut. Misalnya, kita dapat mendefinisikan struktur data bernama “informasi_mahasiswa” yang berisi kumpulan data-data personal mahasiswa meliputi nama, NIM, alamat, dsb [18].

2.4 Sistem Terdistribusi

Sistem terdistribusi merupakan suatu sistem yang terdiri dari beberapa komputer yang terhubung dan bekerja sama untuk menyelesaikan suatu tugas. Pemanfaatan sistem terdistribusi adalah dalam penyimpanan data dan file sharing. Dengan menggunakan sistem terdistribusi, data dapat disimpan dan diakses secara bersamaan oleh beberapa komputer yang terhubung dalam jaringan. Hal ini memudahkan bagi pengguna untuk mengakses dan memperbarui data secara bersamaan, tanpa harus tergantung pada satu komputer server utama. Dengan menggunakan banyak komputer yang terhubung dalam jaringan, sistem terdistribusi dapat mempercepat proses pemodelan dengan membagi beban komputasi ke berbagai komputer. Tujuan penting dari sistem terdistribusi adalah untuk memisahkan aplikasi dari platform, dengan menyediakan lapisan middleware. Tujuan pemisahan ini adalah untuk memastikan transparansi distribusi. Salah satu keuntungan dari model sistem terdistribusi adalah kemampuan untuk meningkatkan kecepatan pemrosesan data [19].

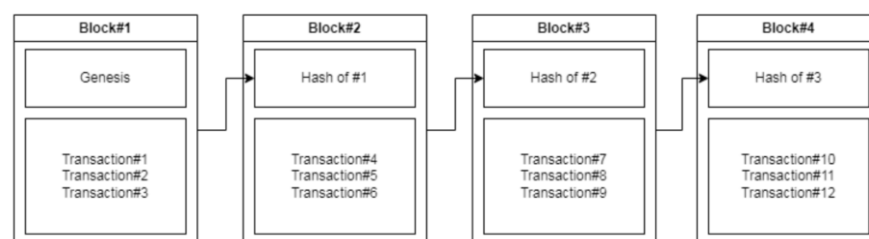
2.5 Web3

Web 3.0, yang pertama kali diperkenalkan pada tahun 2001 oleh Tim Berners-Lee, penemu World Wide Web, mengusung konsep di mana mesin memiliki kemampuan untuk membaca halaman-halaman Web seperti manusia. Meskipun definisi Web 3.0 masih diperdebatkan, beberapa ahli menggambarkan sebagai sebuah fase di mana browser berfungsi sebagai asisten pribadi yang memiliki pemahaman mendalam tentang pengguna dan dapat mencari informasi yang relevan di internet. Ini berarti bahwa internet digunakan untuk membuat koneksi dengan informasi, berbeda dengan Web 2.0 di mana internet digunakan untuk berkoneksi dengan teman. Teknologi Web 3.0 juga memungkinkan pencarian informasi yang lebih efisien, di mana browser dapat menganalisis data dan konteks dari keyword yang dimasukkan oleh pengguna untuk menampilkan hasil yang lebih relevan. Selain itu, profil unik setiap pengguna internet dimanfaatkan untuk menghasilkan hasil pencarian yang disesuaikan dengan kebiasaan browsing masing-masing pengguna. Teknologi yang mendukung implementasi Web 3.0 mencakup berbagai aspek. Pertama adalah Application Programming Interface (API) dan Mashup, di mana API memungkinkan interaksi antara perangkat lunak untuk menghasilkan aplikasi baru yang disebut Mashup. Contohnya adalah penggunaan API Facebook dalam permainan seperti Farmville. Selanjutnya, Web Semantik adalah konsep penting dalam Web 3.0, di mana informasi disajikan dengan metadata yang dapat dibaca oleh mesin. Web Semantik memungkinkan mesin untuk memahami konteks dari informasi yang disajikan. Teknologi lainnya termasuk tampilan web 3 dimensi yang berkaitan dengan dunia virtual seperti Second Life, web yang bersifat media-sentris di mana media seperti gambar dapat digunakan sebagai input untuk

mencari informasi, serta pengembangan web sosial yang memfasilitasi interaksi antar pengguna melalui platform web. Selain itu, konsep Ubiquitous Web menunjukkan bahwa teknologi Web 3.0 akan hadir di berbagai perangkat dan objek dalam kehidupan sehari-hari, seperti komputer, ponsel, atau bahkan furnitur, memungkinkan koneksi internet yang terus-menerus dan adanya integrasi dengan kebiasaan pengguna [20].

2.6 Blockchain

blockchain pertama adalah Bitcoin. Dalam blockchain Bitcoin, informasi yang mewakili uang elektronik terlampir ke alamat digital. Pengguna Bitcoin dapat menandatangani secara digital dan mentransfer hak atas informasi itu kepada pengguna lain, dan blockchain Bitcoin mencatat transfer ini secara publik, memungkinkan semua peserta jaringan untuk memverifikasi validitas transaksi secara independen. Blockchain Bitcoin disimpan, dipelihara, dan dikelola secara kolaboratif oleh sekelompok peserta yang terdistribusi. Hal ini, bersama dengan mekanisme kriptografi tertentu, membuat blockchain tahan terhadap upaya untuk mengubah buku besar nanti (memodifikasi blok atau memalsukan transaksi) [5]. Seperti sistem perusahaan yang lingkungannya memiliki banyak layanan, teknologi blockchain dapat diterapkan [21]. Blockchain dapat berfungsi sebagai blockchain private (*permissioned blockchain*) atau blockchain publik (*public blockchain*) [22].



Gambar 2. 1 Ilustrasi Blockchain

Gambar 2.5 adalah ilustrasi dari blockchain, yang mana blockchain merupakan struktur data yang tidak dapat dihapus ataupun diubah yang dibentuk

oleh serangkaian blok data yang terhubung secara linear dalam urutan waktu. Informasi disimpan di setiap blok dan dienkripsi dengan algoritma kriptografi asimetris untuk memastikan keamanan akses dan transmisi data. Ciri-ciri teknologi blockchain yang dapat diunggulkan adalah desentralisasi (*decentralized*): blockchain terdiri dari blok peer-to-peer (P2P), yang akan merekam dan menyimpan semua transaksi, kepercayaan terdistribusi (*distrusting*): karena teknologi blockchain diterapkan secara desentralisasi sistem, transfer data antar node dalam jaringan tidak memerlukan rasa saling

percaya di antara para peserta, transparansi (*transparency*): melalui blockchain, semua peserta berbagi catatan dan permintaan data dalam node pada struktur terdesentralisasi, dapat dilacak dan unforgeable: blockchain menggunakan stempel waktu untuk mengidentifikasi dan merekam setiap transaksi, sehingga meningkatkan dimensi waktu data, anonimitas: blockchain mengenkripsi data menggunakan teknik enkripsi asimetris. Enkripsi asimetris ini memiliki dua kegunaan dalam blockchain: enkripsi data dan tanda tangal digital (*digital signature*). Enkripsi data dalam blockchain memastikan keamanan data transaksi dan mengurangi risiko kehilangan atau pemalsuan data transaksi. Kredibilitas: pertukaran data blockchain sepenuhnya tergantung pada setiap node untuk membentuk perhitungan yang kuat untuk bertahan dari serangan eksternal tanpa campur tangan manusia [23].

2.6.1 Smart contract

Smart Contract adalah program komputer yang dirancang untuk mengeksekusi transaksi dan kontrak secara otomatis tanpa memerlukan perantara atau pihak ketiga. Fungsi utama dari Smart Contract adalah mengotomatisasi proses transaksi dan kontrak, memastikan kepatuhan terhadap aturan dan kondisi yang telah disepakati, serta mengurangi risiko kesalahan manusia dan meningkatkan efisiensi operasional [24]. *Smart Contract* adalah potongan kode atau perjanjian antara dua individu dalam jaringan blockchain yang mengatur syarat dan ketentuan tertentu. Transaksi hanya akan terjadi saat syarat dan ketentuan tersebut dipenuhi oleh kedua belah pihak, tanpa campur tangan pihak ketiga. Ini dimungkinkan karena semua proses kontrak pintar dilakukan secara otomatis dalam sistem blockchain [25].

2.6.2 Peer Node

Peer adalah node-node yang terhubung ke klien dan bertanggung jawab atas komitmen transaksi ke dalam keadaan dunia. Setiap peer memiliki salinan transaksi mereka sendiri dalam basis data couchdb. Sebuah organisasi dapat memiliki lebih dari satu peer. Meskipun disarankan untuk memiliki beberapa peer dalam suatu orderer untuk menghindari kehilangan data, memiliki lebih dari 3 atau 4 peer mungkin akan mengakibatkan tingkat laten yang lebih tinggi [26]. Blockchain menggunakan jaringan terdistribusi yang biasa disebut sebagai *public ledger* (buku besar publik). Pada system ini setiap node/peserta memiliki salinan identik dari setiap transaksi yang terjadi dalam jaringan blockchain [25].

2.6.3 Ledger

Dalam *Hyperledger Fabric*, sebuah ledger terdiri dari dua komponen yang berbeda namun saling terkait, yaitu kondisi umum (*world state*) dan blockchain. Kedua komponen ini mencerminkan informasi yang signifikan tentang berbagai

entitas bisnis yang terdapat dalam sistem. Kondisi umum merujuk pada basis data yang menyimpan nilai-nilai terkini dari berbagai entitas dalam ledger. Melalui kondisi umum, program dapat dengan mudah mengakses nilai aktual dari entitas tersebut tanpa harus melakukan pencarian melalui log transaksi. Kondisi umum ini dapat berubah secara reguler sesuai dengan pembuatan, pembaruan, atau penghapusan entitas. Di sisi lain, blockchain berperan sebagai log transaksi yang merekam semua perubahan yang terjadi dalam kondisi umum. Transaksi-transaksi ini dikumpulkan dalam blok-blok yang kemudian ditambahkan ke dalam blockchain, memungkinkan pemahaman tentang sejarah perubahan yang telah membentuk kondisi umum saat ini. Penting untuk dicatat bahwa struktur data blockchain bersifat tidak dapat diubah setelah ditulis, sehingga memastikan keabsahan dan integritas data dalam sistem [27].

2.6.4 Channel

Untuk menciptakan dan mentransfer aset dalam jaringan *Hyperledger Fabric*, sebuah organisasi harus tergabung dalam sebuah *channel*. *Channel* merupakan lapisan komunikasi pribadi antara organisasi-organisasi tertentu yang tidak terlihat oleh anggota lain dalam jaringan. Setiap *channel* terdiri dari sebuah ledger terpisah yang hanya dapat diakses oleh anggota saluran, yang diperbolehkan untuk menghubungkan peer mereka ke *channel* dan menerima blok-blok transaksi baru dari layanan pemesanan. Sementara itu, peer, node pemesanan, dan Otoritas Sertifikat membentuk infrastruktur fisik jaringan, *channel* adalah proses di mana organisasi saling terhubung dan berinteraksi [28].

2.7 Hyperledger Fabric

Hyperledger Fabric adalah teknologi blockchain yang diatur oleh Linux Foundation, dibuat untuk mempromosikan pengguna teknologi blockchain ke perusahaan, terutama untuk aplikasi Perusahaan [29].

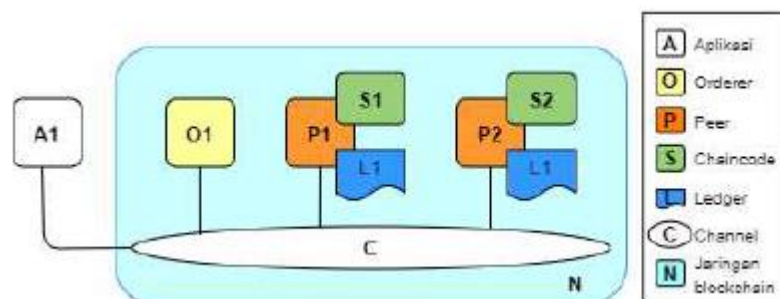


Gambar 2. 2 Logo Hyperledger Fabric

Hyperledger Fabric adalah *platform Distributed Ledger Technology (DLT) open source* yang diizinkan perusahaan, dirancang untuk digunakan dalam konteks perusahaan, yang memberikan beberapa kemampuan membedakan kunci dibandingkan *platform ledger* atau *blockchain* populer lainnya. *Fabric* adalah *platform ledger* terdistribusi pertama yang mendukung smart contract yang ditulis

dalam bahasa pemrograman java, go dan node.js. *Platform fabric* para peserta dalam jaringannya dapat diketahui satu sama lain karena bersifat privat (*permissioned*). Sedangkan yang bersifat publik yang mana para pesertanya adalah anonim oleh karena itu mungkin tidak percaya satu sama lain, misalnya menjadi pesaing dalam industri yang sama [30]. Berikut adalah beberapa tujuan dari Hyperledger [29]:

1. *Rich Queries*: Ini berarti pengguna dapat mengeksekusi transaksi pada platform blockchain. Dengan fitur ini, pengguna dapat melakukan kueri yang lebih kompleks dan mendapatkan hasil yang relevan.
2. *Arsitektur Modular*: Hyperledger Fabric memiliki arsitektur modular yang memungkinkan pengguna menggunakan berbagai modul secara bersamaan atau sesuai kebutuhan. Ini memberikan fleksibilitas dalam mengatur komponen-komponen sistem.
3. *Perlindungan Kunci Digital dan Data Sensitif*: Hyperledger Fabric bekerja dengan konsep distribusi buku besar umum dan melindungi kunci digital serta data dari manipulasi. Keamanan ini sangat penting dalam lingkungan blockchain.
4. *Data Berizin*: Hanya pihak-pihak tertentu yang diizinkan untuk melihat dan menggunakan data tertentu. Dengan mekanisme izin yang diterapkan, Hyperledger Fabric memastikan akses yang terkontrol terhadap informasi.



Gambar 2. 3 Arsitektur *Hyperledger Fabric*

Hyperledger Fabric adalah salah satu kerangka kerja blockchain yang populer dan banyak diadopsi. Dalam Hyperledger Fabric, terdapat konsep channel yang memungkinkan organisasi-organisasi yang berpartisipasi untuk berkomunikasi secara rahasia. Setiap organisasi dapat bergabung dalam beberapa channel sekaligus. Komponen utama dalam Hyperledger Fabric meliputi Peer, Orderer, CA (Certificate Authority), dan Client. Fabric menggunakan konsensus berbasis voting yang mengasumsikan bahwa semua peserta dalam jaringan adalah sebagian terpercaya. Dalam arsitektur ini, setiap transaksi harus mendapatkan persetujuan dari endorsing peer sebelum dimasukkan ke dalam buku besar. Endorsing peer mengeksekusi transaksi terhadap basis data state saat ini dan menghasilkan hasil

transaksi. Hasil transaksi beserta tanda tangan endorsing peer dikirim kembali kepada klien. Orderer bertanggung jawab mengurutkan transaksi menjadi blok dan mengirimkannya ke semua peer. Setiap transaksi dalam blok ditandai sebagai valid atau invalid. Peer memvalidasi transaksi dan memasukkannya ke dalam buku besar. Dengan arsitektur ini, Hyperledger Fabric memastikan keamanan, kerahasiaan, dan konsistensi data. Dalam produksi, Hyperledger Fabric memperhatikan stabilitas dan operasi. Versi 1.4 merupakan versi pertama yang mendukung jangka panjang. Dengan fitur-fitur kunci dan popularitasnya, Hyperledger Fabric terus menarik perhatian para pengembang dan arsitek perangkat lunak. Selain itu, *Hyperledger Fabric* juga memperkenalkan konsep *private data collection*. Dalam hal ini, data sensitif hanya terlihat oleh pihak-pihak yang berwenang. Setiap peer memiliki ledger yang terdiri dari dua bagian: bagian data publik dan bagian data pribadi. Data publik dapat dilihat oleh semua peer, sedangkan data pribadi hanya dapat diakses oleh peer yang memiliki izin. Penting untuk memahami bahwa Hyperledger Fabric bukan hanya tentang teknologi blockchain, tetapi juga melibatkan aspek-aspek seperti manajemen identitas, keamanan, dan pengaturan akses. Dengan pemahaman yang mendalam tentang arsitektur ini, pengembang dapat merancang dan mengembangkan aplikasi yang aman dan efisien di dalam jaringan Hyperledger Fabric. Dalam praktiknya, Hyperledger Fabric telah digunakan dalam berbagai kasus penggunaan, termasuk pendidikan, perawatan kesehatan, Internet of Things (IoT), logistik, dan rantai pasok. Keberhasilan implementasi ini menunjukkan potensi besar dari kerangka kerja ini dalam menghadirkan solusi bisnis yang inovatif dan andal. Dengan semakin berkembangnya teknologi blockchain, Hyperledger Fabric tetap menjadi pilihan yang relevan bagi organisasi-organisasi yang membutuhkan infrastruktur blockchain yang kuat, aman, dan dapat disesuaikan [31].

2.8 Konsensus

Konsensus merujuk pada kesepakatan sukarela antara pihak-pihak yang terlibat dalam kontrak. Konsensus adalah prasyarat pertama agar sebuah kontrak menjadi sah secara hukum. Ini berarti bahwa semua pihak yang terlibat dalam kontrak harus setuju secara sukarela terhadap semua ketentuan dan kondisi yang tercantum dalam kontrak tersebut. Tidak boleh ada paksaan atau tekanan yang mempengaruhi keputusan para pihak untuk menyetujui kontrak tersebut. Konsensus mencerminkan adanya pemahaman bersama dan persetujuan penuh atas hak dan kewajiban masing-masing pihak sebagaimana yang diatur dalam kontrak [32].

2.8.1 Proof of Work (PoW)

Proof of Work (PoW) adalah mekanisme konsensus yang digunakan dalam teknologi blockchain untuk memvalidasi transaksi dan menambah blok

baru ke rantai blok. Mekanisme ini mengharuskan node atau penambang untuk memecahkan teka-teki kriptografi yang kompleks, yang memerlukan daya komputasi yang signifikan. Penambang yang pertama kali memecahkan teka-teki tersebut diberi hak untuk menambahkan blok baru ke blockchain dan menerima imbalan dalam bentuk mata uang kripto. Keunggulan utama PoW adalah keamanannya, karena memerlukan waktu dan energi yang besar untuk memecahkan teka-teki kriptografi, sehingga membuat serangan terhadap jaringan menjadi sangat mahal dan sulit dilakukan. Namun, kelemahan utama PoW adalah konsumsi energi yang sangat tinggi, karena setiap penambang berlomba-lomba memecahkan teka-teki yang sama, sehingga banyak energi terbuang percuma [33].

2.8.2 Proof of Stake (PoS)

Proof of Stake (PoS) adalah mekanisme konsensus yang dikembangkan sebagai alternatif hemat energi untuk PoW. Dalam PoS, pemimpin atau validator dipilih berdasarkan jumlah aset digital (stake) yang mereka miliki atau depositkan ke jaringan blockchain. Semakin besar stake yang dimiliki, semakin besar kemungkinan seseorang dipilih untuk memvalidasi transaksi dan menambah blok baru ke rantai blok. Berbeda dengan PoW yang memerlukan daya komputasi besar, PoS lebih efisien karena tidak memerlukan penambang untuk memecahkan teka-teki kriptografi yang kompleks. Validator dipilih secara acak berdasarkan stake mereka, menggunakan algoritma seperti Follow-the-Satoshi (FTS). Algoritma FTS bekerja dengan mengambil masukan berupa seed (misalnya header blok sebelumnya atau string acak) dan menghasilkan indeks token. Token tersebut kemudian digunakan untuk menemukan dan memilih pemilik token saat ini sebagai pemimpin. Keunggulan PoS adalah konsumsi energinya yang jauh lebih rendah dibandingkan PoW, serta kecepatan lebih tinggi dalam menghasilkan blok dan mengonfirmasi transaksi. Namun, PoS juga memiliki kelemahan, seperti potensi sentralisasi jika sebagian besar token dikuasai oleh segelintir pihak [33].

2.8.3 Practical Byzantine Fault Tolerant (PBFT)

PBFT adalah algoritma konsensus yang bertujuan untuk menyelesaikan masalah Byzantine Fault Tolerance (BFT) secara praktis. Dikembangkan oleh Miguel Castro dan Barbara Liskov pada tahun 1999, PBFT mengurangi kompleksitas waktu dari eksponensial menjadi polinomial, memungkinkan penyelesaian masalah Byzantine yang sebelumnya hanya teoretis menjadi lebih praktis. Konsep inti PBFT terdiri dari tiga bagian: view, replica, dan role. Dalam PBFT, setiap view merepresentasikan keadaan global dari sistem, dengan node-node yang berpartisipasi menjadi replicas. Dalam setiap view, satu replica berfungsi sebagai primary, sementara yang lainnya sebagai backups. Algoritma PBFT terdiri dari tiga protokol utama: konsensus, penggantian view, dan checkpoint. Protokol konsensus digunakan untuk memastikan konsistensi data di

seluruh jaringan, sementara protokol penggantian view dan checkpoint bertujuan untuk menggantikan node bermasalah dan membersihkan data yang kedaluwarsa. Proses eksekusi PBFT terdiri dari tiga langkah utama: pre-prepare, prepare, dan commit. Node utama menghasilkan pesan pre-prepare, yang kemudian diverifikasi oleh node cadangan sebelum diikuti dengan pesan prepare. Setelah menerima cukup banyak pesan commit, konsensus tercapai, dan node-node mengeksekusi permintaan dan menulis data [34].

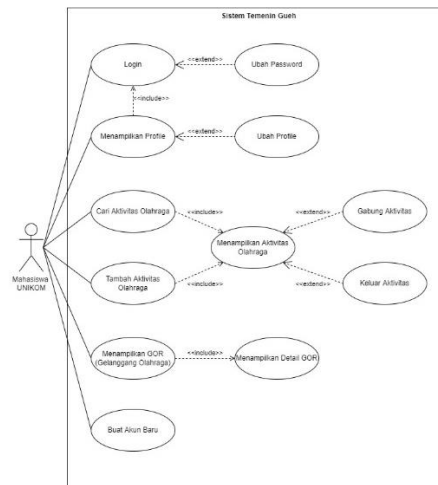
2.9 UML(*Unified Modelling Language*)

Unified Modelling Language (UML) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang, dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML, kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi, dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan class dan operation dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa berorientasi objek seperti C++, Java, C#, atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C [35].

Terdapat beberapa diagram yang biasanya digunakan untuk memodelkan analisis fungsional dalam rangka pengembangan perangkat lunak. Berikut diantaranya diagram yang umum digunakan :

a) *Use Case Diagram*

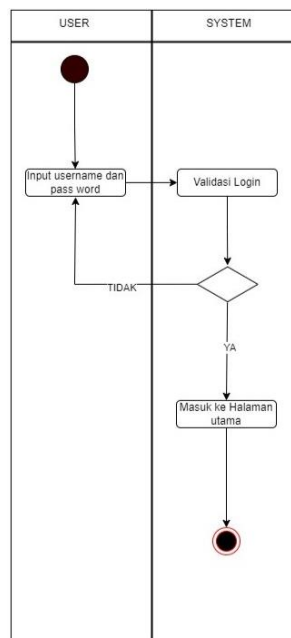
Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. Ilustrasi dari *use Case Diagram* dapat dilihat pada Gambar 2.1.



Gambar 2. 4 Contoh Use Case Diagram

b) *Activity Diagram*

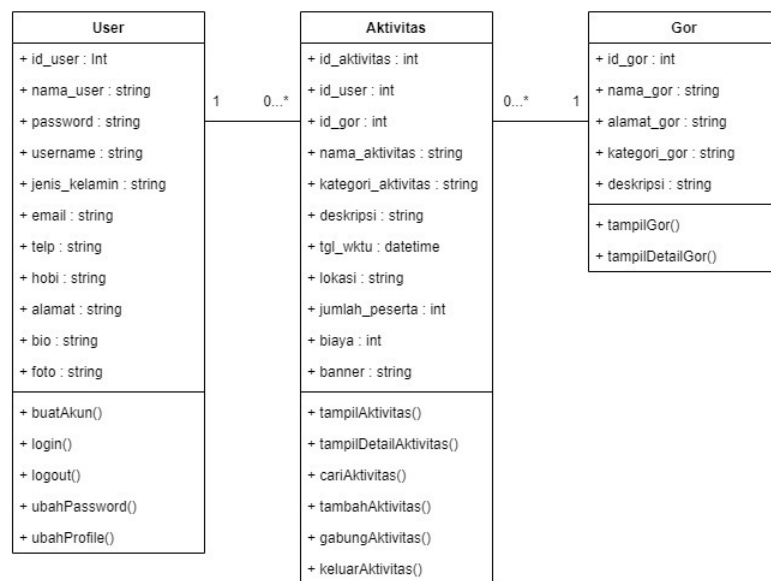
Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity Diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Berikut contoh dari *Activity Diagram* pada Gambar 2.2.



Gambar 2. 5 Contoh Activity Diagram

c) Class Diagram

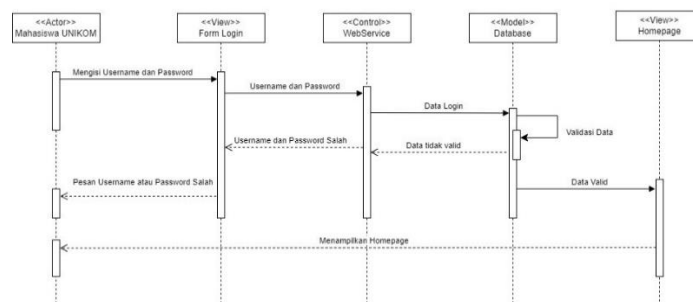
Class diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). Berikut contoh dari *class diagram* pada Gambar 2.3.



Gambar 2. 6 Contoh Class Diagram

d) Sequence Diagram

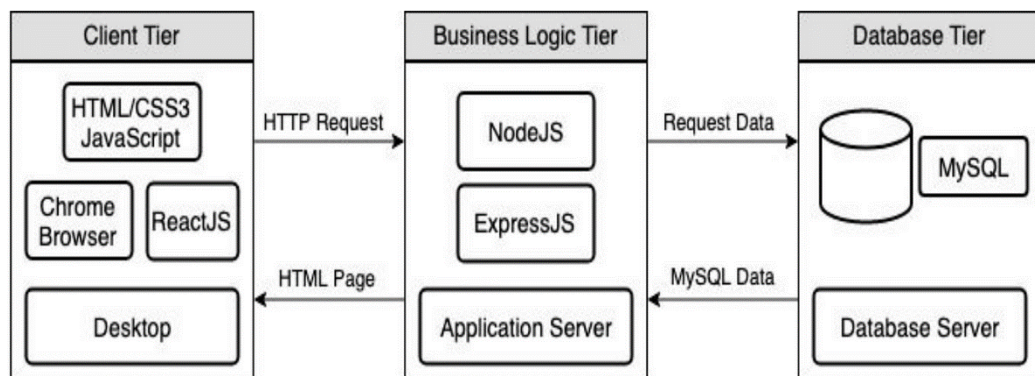
Sequence Diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence Diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). Berikut contoh dari *Sequence Diagram* pada Gambar 2.4.



Gambar 2. 7 Contoh Sequence Diagram

2.10 MERN Stack

MERN Stack merupakan rangkaian teknologi yang populer dalam pengembangan aplikasi web, terdiri dari empat komponen utama: MongoDB/MySQL, Express.js, React.js, dan Node.js. MERN Stack menawarkan sejumlah keunggulan bagi pengembang. Pertama-tama, semua teknologi dalam tumpukan menggunakan bahasa pemrograman JavaScript, sehingga memungkinkan pengembang untuk bekerja secara konsisten di seluruh stack. Ini juga memudahkan tim pengembangan untuk berkolaborasi dan mengelola kode. Selain itu, MERN Stack memiliki skalabilitas yang baik, yang memungkinkan aplikasi untuk ditingkatkan dengan mudah sesuai dengan kebutuhan bisnis dan pertumbuhan pengguna.



Gambar 2. 8 Stuktur MERN Stack

1. MySQL adalah sistem manajemen basis data (Database Management System atau DBMS) yang populer dan sering digunakan. Ini merupakan perangkat lunak open-source yang memungkinkan pengguna untuk menyimpan, mengelola, dan mengakses data dalam berbagai aplikasi.
2. Express.js adalah kerangka kerja web yang berjalan di sisi server menggunakan Node.js. Express.js menyediakan alat yang kuat untuk mengelola rute, permintaan HTTP, serta middleware untuk mengatur aliran data dalam aplikasi.
3. React.js adalah kerangka kerja JavaScript yang fokus pada pembangunan antarmuka pengguna (UI) yang responsif dan efisien. Dengan menggunakan konsep komponen, React memungkinkan pengembang untuk membangun aplikasi web dengan UI yang mudah dikelola dan dapat disesuaikan.

4. Node.js adalah runtime lingkungan JavaScript yang memungkinkan kode JavaScript dijalankan di sisi server. Dengan Node.js, pengembang dapat membuat server web dan mengelola permintaan HTTP dengan mudah.

Selain itu, karena setiap komponen dalam MERN Stack dioptimalkan untuk kinerja, aplikasi yang dibangun dengan menggunakan teknologi ini cenderung responsif dan efisien. Dengan demikian, pengguna dapat merasakan pengalaman pengguna yang lebih baik tanpa harus menunggu lama untuk memuat halaman atau menanggapi Tindakan.

2.11 Javascript

JavaScript adalah bahasa yang berbentuk kumpulan skrip yang berjalan pada suatu dokumen HTML. Sebagai bahasa skrip pertama untuk web sepanjang sejarah internet, JavaScript memberikan kemampuan tambahan terhadap bahasa HTML dengan menjalankan perintah-perintah di sisi pengguna, yaitu di sisi browser, bukan di sisi server web. JavaScript bergantung kepada browser (navigator) yang memanggil halaman web yang berisi skrip-skrip JavaScript, yang disisipkan di dalam dokumen HTML. JavaScript juga tidak memerlukan kompilator atau penterjemah khusus untuk menjalankannya, karena kompilator JavaScript sendiri sudah termasuk di dalam browser tersebut. Ini berbeda dengan bahasa "Java", yang memerlukan kompilator khusus untuk menterjemahkannya di sisi pengguna/klien [36].

2.12 React.Js

React adalah library JavaScript open-source yang deklaratif, efisien, dan fleksibel untuk membangun antarmuka pengguna. React memungkinkan pembuatan antarmuka pengguna yang kompleks dengan menggunakan serangkaian kode kecil yang terisolasi yang disebut "komponen". React JS digunakan untuk mengelola lapisan tampilan dalam aplikasi satu halaman (SPA) dan pengembangan aplikasi mobile. React JS dikelola oleh Facebook, Instagram, komunitas pengembang, dan korporasi. Tujuan utama React adalah memberikan kecepatan, kesederhanaan, dan skalabilitas [37].

2.13 Node.Js

Node.js adalah mesin runtime JavaScript sumber terbuka yang digunakan untuk membangun aplikasi jaringan yang dapat diskalakan, didasarkan pada mesin V8 milik Chrome. Meskipun masih muda, Node.js sangat populer di kalangan komunitas sumber terbuka dan perusahaan besar seperti Microsoft, IBM, dan PayPal, karena arsitektur berbasis event non-blocking yang memudahkan skalabilitas. Node.js digunakan dalam pengembangan sistem-sistem kritis yang memerlukan keamanan tinggi. Pengembang mendistribusikan pustaka melalui NPM, manajer paket terbesar dengan lebih dari satu juta paket dan tingkat

pertumbuhan hampir 800 paket per hari. Namun, ketergantungan pada pustaka pihak ketiga membawa risiko keamanan, seperti kepercayaan pada keamanan dan pemeliharaan pustaka, serangan seperti typosquatting, dan risiko pembaruan atau penghapusan paket [38].

2.14 Couch DB

CouchDB adalah sistem manajemen basis data yang berbasis dokumen dan ditulis dalam bahasa pemrograman Erlang. Ia menyediakan kueri dinamis dan menggunakan format JSON untuk penyimpanan dan kueri data. Berbeda dengan basis data relasional yang menyimpan data dalam tabel, CouchDB mendukung kumpulan dokumen independen dalam setiap basis data. Ia juga menyediakan semantik ACID dan mengimplementasikan kontrol konkurensi multi-versi (MVCC) sehingga dapat digunakan oleh banyak pembaca dan penulis secara bersamaan tanpa konflik. Selain itu, CouchDB dapat melakukan replikasi ke perangkat yang dapat offline dan mendukung sinkronisasi data saat kembali online. Ia juga mendukung daftar terurut dan array asosiatif. Nilai-nilai bidang dalam dokumen dapat berupa string, angka, atau tanggal. CouchDB dapat menyimpan berbagai jenis data bersama-sama. Namun, perlu diperhatikan bahwa CouchDB memerlukan ruang tambahan untuk overhead dibandingkan dengan beberapa basis data lainnya [39].

2.15 Docker

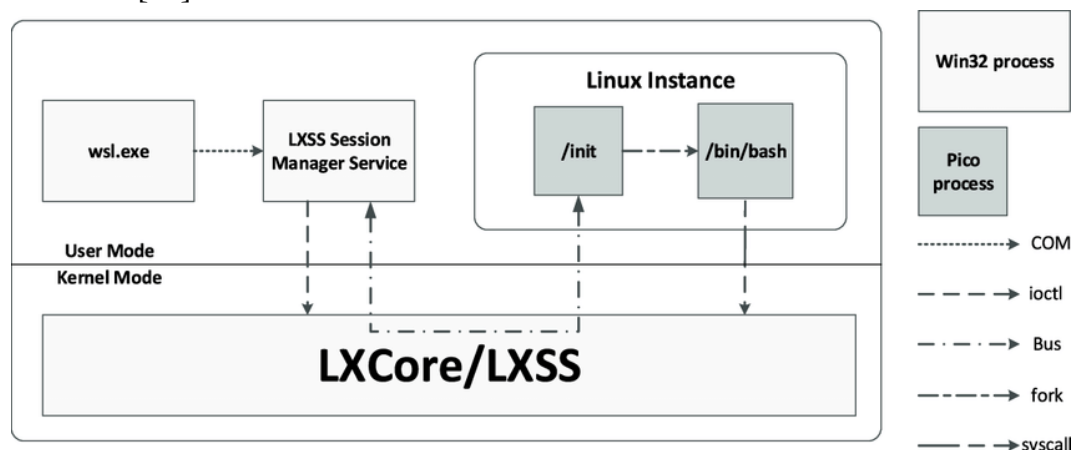
Docker adalah sebuah project open-source yang menyediakan platform terbuka untuk developer maupun sysadmin untuk dapat membangun, mengemas, dan menjalankan aplikasi dimanapun di dalam sebuah container. Docker menggunakan arsitektur client-server. Dimana client dan docker berkomunikasi dengan daemon docker, yang melakukan suatu tindakan untuk membangun, menjalankan, dan mendistribusikan container docker. Client docker dan daemon dapat berjalan pada sistem yang sama. Client docker dan daemon berkomunikasi menggunakan REST API, melalui soket UNIX atau antarmuka jaringan [40].

2.16 Windows Subsystem for Linux (WSL)

WSL merupakan aplikasi paling prominent dari proses pico di Windows, dirilis pada tahun 2017 bersamaan dengan versi 64-bit dari Windows 10 Fall Creators Update setelah lebih dari satu tahun pengujian beta. Ini memungkinkan pengguna untuk menjalankan program Linux di lingkungan pengguna Windows 10 dengan mengaitkan setiap aplikasi Linux yang berjalan dengan proses pico. Ini memungkinkan pengguna untuk menjalankan berkas ELF tanpa perlu mesin virtual, modifikasi kode sumber, atau aplikasi perantara. Selain itu, pengguna dapat mengunduh aplikasi untuk masing-masing dari lima distribusi Linux yang saat ini didukung dari Microsoft Store: Ubuntu, Debian GNU/Linux, openSUSE

Leap 42, SUSE Linux Enterprise Server 12, dan Kali Linux. Berikut adalah proses-proses yang merupakan komponen dari implementasi WSL:

1. `wsl.exe` atau `bash.exe`: Proses baris perintah lingkungan pengguna melalui mana pengguna berinteraksi dengan WSL. Program ini dapat dijalankan lebih dari satu kali.
2. `LxssManager`: Layanan Windows yang memfasilitasi komunikasi antara proses `wsl.exe`/`bash.exe` dan penyedia pico WSL.
3. `lxss`: Layanan sistem Windows yang berfungsi sebagai penyedia pico WSL.
4. `/init`: Proses pico Linux yang memfasilitasi komunikasi antara proses Windows dan keturunannya. `lxss` membuat satu proses `/init` untuk setiap distribusi Linux yang dijalankan.
5. `/bin/bash`: Proses pico Linux yang mendukung program shell WSL. Setiap proses `wsl.exe` dan `bash.exe` dipasangkan dengan proses `/bin/bash` yang sesuai[41].



Gambar 2. 9 Ilustrasi Pemrosesan Implementasi WSL

2.17 Interplanetary File System (IPFS)

IPFS (Interplanetary File System) dijelaskan sebagai sistem penyimpanan file terdesentralisasi yang menggunakan jaringan distribusi untuk menyimpan dan mengambil file. IPFS beroperasi pada jaringan terdistribusi, memastikan file tidak disimpan di satu lokasi saja, yang meningkatkan redundansi dan toleransi terhadap kegagalan. Alih-alih merujuk file berdasarkan lokasinya, IPFS menggunakan pengalamatan konten. Setiap file diberi pengenal konten unik (CID) berdasarkan isinya, sehingga memudahkan pengambilan dan verifikasi file. Dalam skenario yang disajikan, file yang dienkripsi dan diunggah oleh admin disimpan di IPFS, memastikan file tetap aman dan dapat diambil dengan mudah menggunakan CID-nya. IPFS menyediakan nilai hash untuk konten yang disimpan, yang dapat

disimpan di blockchain untuk memastikan integritas dan keaslian konten. Hash ini berfungsi sebagai referensi untuk konten tanpa menyimpan data sebenarnya di blockchain. Sifat terdistribusi IPFS memastikan bahwa file disimpan secara redundan di seluruh jaringan, meningkatkan toleransi terhadap kegagalan dan memastikan file tetap dapat diakses meskipun beberapa node gagal. Secara keseluruhan, IPFS menyediakan cara yang kuat, terdesentralisasi, dan aman untuk menyimpan dan mengambil file, yang melengkapi kemampuan blockchain dengan menangani penyimpanan file besar secara efisien [7].

2.18 QR Code

QR Code adalah image berupa matriks dua dimensi yang memiliki kemampuan untuk menyimpan data di dalamnya. *QR Code* merupakan evolusi dari kode batang (*barcode*). Barcode merupakan sebuah simbol penandaan objek nyata yang terbuat dari pola batang-batang berwarna hitam dan putih agar mudah untuk dikenali oleh komputer.



Gambar 2. 10 Contoh Qr Code

QR Code merupakan singkatan dari *Quick Response Code*, atau dapat diterjemahkan menjadi kode respon cepat. *QR Code* dikembangkan oleh Denso Corporation, sebuah perusahaan Jepang yang banyak bergerak di bidang otomotif. *QR Code* ini dipublikasikan pada tahun 1994 dengan tujuan untuk pelacakan kendaraan di bagian manufaktur dengan cepat dan mendapatkan respon dengan cepat pula [42].