

BAB II

TINJAUAN PUSTAKA

2.1 Registrasi Tanah

Sistem registrasi tanah merupakan metode pendaftaran atau pencatatan kepemilikan dan hak atas tanah yang dilakukan oleh Lembaga pemerintah. registrasi tanah juga merupakan pendokumentasian dari hak milik tanah, perubahan kesesuaian dari kepemilikan tanah yang dapat menjaga hak-hak dari kepemilikan tanah menurut peraturan perundang-undangan yang tertera [25].

2.1.1 Registrasi Tanah di Indonesia

Sistem registrasi tanah di Indonesia sudah diatur oleh undang-undang. Berikut beberapa jenis kepemilikan tanah menurut pasal 16 UUD Nomor 5 Tahun 1960 mengenai peraturan Dasar Pokok-Pokok Agraria [26].

1. Hak Milik

Jenis hak tersebut adalah hak kepemilikan yang kuat atas tanah. Pihak yang dimana sudah memiliki hak kepemilikan tanah mendapatkan hak penuh untuk tanah tersebut, termasuk bangunan diatas tanah. Bentuk dari kepemilikan tersebut adanya Sertifikat Tanah.

2. Hak Guna Usaha

Jenis hak ini merupakan khusus yang sudah diberikan kepada pemerintah yang bertujuan untuk membawa kegiatan usaha yang meliputi pertanian, peternakan, serta perikanan. Hak ini Batasan jangka waktu maksimum 25 tahun. Namun perusahaan untuk membutuhkan jangka lama selama maksimum 35 tahun.

3. Hak Guna Bangunan

Jenis hak ini memberikan pemilik untuk membangun sebuah bangunan yang bukan miliknya. Hak tersebut memiliki jangka waktu selama 30 tahun dan dapat diperpanjang. Hak tersebut memiliki perizinan yang disetujui oleh negara.

4. Hak Penggunaan

Hak ini menggunakan atas pengambilan tanah dari kepemilikan

tanah negara atau kepemilikan tanah pribadi. Hak ini tidak dapat dialihkan kepada pihak lain terkecuali dengan izin dari pemilik tanah.

5. Hak Sewa

Hak ini merupakan hak yang diberikan seseorang untuk menyewa tanah atau bangunan tersebut dengan melakukan pembayaran sebelumnya digunakan atau pada waktu yang sudah ditentukan.

6. Hak untuk Membuka Lahan dan Hasil Hutan

Hak ini dapat dimiliki oleh kalangan warga Indonesia dan diatur dengan peraturan pemerintah di Indonesia, yang dimana tidak serta merta memberikan hak kepemilikan tanah terhadap orang lain.

2.2 Sertifikat Tanah

Sertifikat tanah adalah surat tanda bukti sebagai hak atas tanah, hak pengilaan, tanah wakaf, hak milik atas satuan rumah susun, serta hak tanggungan yang dimana masing-masing sudah dicatat kedalam buku tanah yang bersangkutan [26].

Halaman 1		NAMA PEMEGANG HAK											
a) HAK MILIK No. 808 Desa Ciporek	b) PENUNJUK Desa Desa Ciporek RT 2 - 15	15/00											
c) ANAK PERSIL 1. <u>1000</u> 2. <u>Pembelian</u> 3. <u>Perubahan dari R. 30/</u> <u>Ciporek.</u> 4. <u>Penggabungan</u>	d) PEMBUKUAN Bandung Tgl. <u>29-8-1988</u> A.a. BUPATI/WALIKOTA/BANDUNGA-KDH A.a. BUPATI/WALIKOTA/BANDUNGA KDH Tt. II Bandung Tt. II Bandung Kepala Kantor Agraria Kepala Kantor Agraria a.b. Kepala Seksi Pendaftaran Tanah a.b. Kepala Seksi Pendaftaran Tanah	Drs. U.S.M.A.S. S.H., N.S. Di. Sistem Kewilayatan No. 1 0001033 Di. 800 871 276											
e) SURAT KEPUTUSAN	f) SERTIFIKAT Bandung Tgl. <u>2-9-1988</u> A.a. BUPATI/WALIKOTA/BANDUNGA-KDH A.a. BUPATI/WALIKOTA/BANDUNGA KDH Tt. II Bandung Tt. II Bandung Kepala Kantor Agraria Kepala Kantor Agraria a.b. Kepala Seksi Pendaftaran Tanah a.b. Kepala Seksi Pendaftaran Tanah	Drs. U.S.M.A.S. S.H., N.S. Di. Sistem Kewilayatan No. 1 0001033 Di. 800 871 276											
g) Ganti rugi/wang wajib	k) CATATAN MENGENAI PAJAK												
h) Lama nya hak berikat	<table border="1"> <thead> <tr> <th>Tahun</th> <th>Besarnya</th> <th>Perambahan</th> <th>Pengurangan</th> <th>Catatan</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table>			Tahun	Besarnya	Perambahan	Pengurangan	Catatan					
Tahun	Besarnya	Perambahan	Pengurangan	Catatan									
i) Berakarnya hak Tgl.	Lusa : 273 M2 (Dua ratus tujuh puluh tiga meter persegi)												
j) SHRAY-DIKHR Camber-Situasi Tgl. 15 - 6 - 1988 No. 5347 / 1988													

Gambar 2. 1 Bentuk Sertifikat Tanah

2.3 Sistem Informasi

Sistem Informasi merupakan gabungan dari sejumlah komponen (manusia, komputer, teknologi informasi, prosedur kerja) yang melakukan pemrosesan data menjadi informasi untuk mencapai suatu tujuan. Sistem informasi dapat disimpulkan sebagai kumpulan elemen yang saling berhubungan satu sama lain, sehingga dapat membentuk satu kesatuan yang berintegrasi pada data, memproses dan menyimpan hingga menjadi sebuah informasi [27].

2.4 Sistem Informasi Geografis (SIG)

Sistem Informasi Geografis adalah kumpulan-kumpulan terorganisir dari perangkat keras komputer, perangkat lunak, data geografis, serta personel yang didesain untuk memperoleh, menyimpan, memperbaiki, memanipulasi, menganalisis, dan menampilkan bentuk dari informasi yang bersifat geografis. Sistem Informasi Geografis adalah sebuah alat yang berfungsi sebagai pengumpulan, penyimpanan, pengambilan kembali terhadap data yang diinginkan, serta penayangan data yang berupa kenyaataan pada bumi [27].

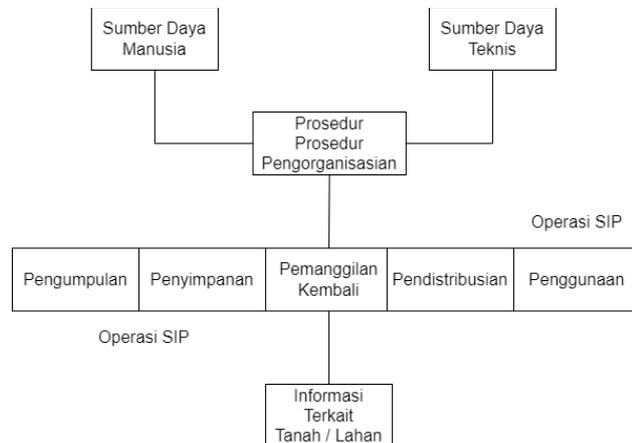
2.5 Sistem Informasi Pertanahan (SIP)

Sistem Informasi Pertanahan atau bisa dikenal dengan *Land-related information system* atau *land-record information system* (LRIS) merupakan turunan yang berspesifik dari Sistem Informasi Geografis (SIG). Sistem Informasi Pertanahan merupakan pergantian literatur terhadap SIG dalam merujuk kepada informasi mengenai kepemilikan tanah. Sistem Informasi Pertanahan dirancang sebagai detail informasi dari kepemilikan tanah. Informasi tersebut direkam pada peta yang berskala 1:1,000 hingga 1:10,000 dan disimpan dalam sistem berbasis komputer [28].

2.5.1 Konsep Sistem Informasi Pertanahan

Konsep Sistem Informasi Pertanahan (SIP) memberikan konsep yang terdiri dari komponen-komponen sumberdaya manusia, sumber-daya teknis, dan prosedur-prosedur pengorganisasian yang bekerja sama terhadap hasil informasi (terkait tanah) dalam menuduku kebutuhan manajerial. Untuk memberikan hasil

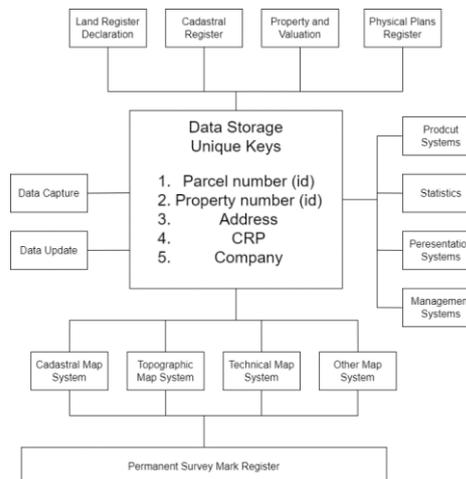
kepada informasi yang diperlukan yaitu, pengumpulan, penyimpanan, pemanggilan kembali, pemeliharaan, serta pemrosesan/analisis dari data, sehingga menghasilkan dan pendistribusian terhadap informasi [28]. Berikut contoh gambar pada Konsep Sistem Informasi Pertanahan (SIP) pada gambar 2.2.



Gambar 2. 2 Konsep Sistem Informasi Pertanahan (SIP)

Sumber: [Pramdhana Satya 2019]

Konsep dari SIP itu sendiri mempunyai konsep yang sangat detail terhadap pengelompokan informasi yang berbentuk data spasial dan beberapa modul (sub-sistem) yang khas dengan pertanahan. Berikut contoh gambar Konsep Sistem Informasi Pertanahan (SIP) pada gambar 2.3.



Gambar 2. 3 Konsep Sistem Informasi Pertanahan (SIP) Terhadap Sistem

Sumber: [Pramdhana Satya 2019]

Dari gambar berikut tool yang terkait kepada Sistem Informasi Petanahan (SIP) sebagai berikut;

1. Sebagai sistem legal (yuridis)/fiskal,
2. Sebagai sistem SIG

Sebagai modul sistem, Sistem Informasi Pertanian terdiri dari semua informasi yang saling berhubungan dengan data spasial. Meskipun demikian, sistem ini mengimplementasikannya kedalam modul *cadastral register*, *land register declaration*, dan *property valuation system*. Sementara relasi terhadap pengguna didasarkan kepada beberapa *field* yang bersifat *primary key*, seperti halnya *property-number*, *parcel-number*, *person identifier*, *address code* (kode-pos), dan *company-number* [28].

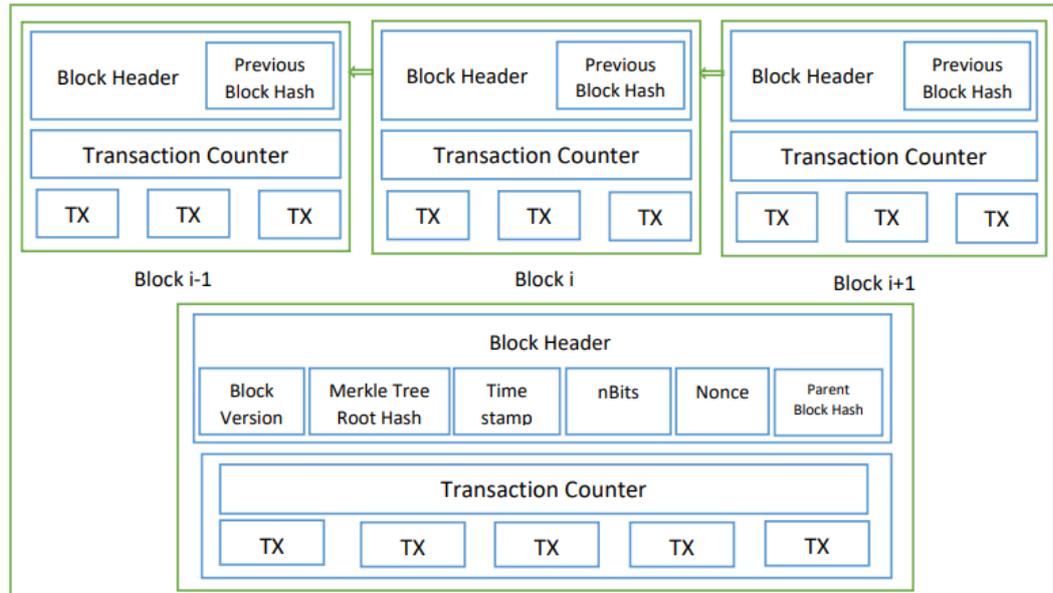
2.6 Blockchain

Blockchain adalah kecerdasan buatan yang telah berkembang pesat pada era revolusi industri 4.0. *Blockchain* memiliki peran yang berbeda namun saling melengkapi. Dengan salah satu fitur utamanya yaitu *immutability* yang bertugas sebagai pengamanan data agar tidak dapat diubah oleh seorang pengguna [29].

Awalnya terciptanya Blockchain untuk cryptocurrency bitcoin pada tahun 2008 blockchain dikembangkan oleh Satoshi Nakamoto dengan menerapkan jaringan *peer-to-peer*, dimana semua *peer* berfungsi sebagai semua node yang saling terhubung. Blockchain merupakan buku besar (*ledger*) yang tersebar secara publik [30].

Blockchain merupakan basisdata yang terdesentralisasi (*decentralized database*) menggunakan node independent sebagai penyimpanan serta pengambilan data. Teknologi blockchain melakukan penghubungan blok data secara berurutan kedalam buku besar yang sudah terdistribusi. Pada setiap blok menyimpan berbagai konten yang bisa disebut dengan “*Hash*”. Hash pada blockchain berfungsi sebagai pengelompokan dan penyimpanan pada setiap blok yang terhubung pada hash [31].

Blockchain mencatat transaksi pada transaksi tanpa ada pihak ketiga. Maka blockchain memiliki cara seperti pemantauan informasi transaksi, kontrak atau data yang sudah terdaftar dan diverifikasi secara independent. Serta menjamin hanya ada satu pemilik dan tidak ada duplikasi dari objek atau komponen dari data [31].



Gambar 2. 4 Struktur Blockchain

Sumber: [Sen. 2021]

2.6.1 Jenis Blockchain

Para peneliti telah memberikan jenis-jenis dari blockchain itu sendiri. Blockchain terbagi menjadi 3 jenis, yaitu public, private, permissioned. Berikut penjelasan dari ketiga jenis dari blockchain:

1. Public Blockchain

Public Blockchain merupakan jaringan yang terdistribusi. *Public blockchain* itu sendiri memberikan setiap orang yang bisa melakukan semua tingkatan fungsional, sehingga mendapatkan jaringan yang sama dalam pengolaan isi terhadap sistem yang menjalankan node, *cryptocurrency* merupakan perdangan token dengan secara menambang. *Source code* pada sistem tersebut bersifat *open source*, dengan contoh *Ethereum* serta *Bitcoin*. *Public blockchain* cenderung bersifat aman dan tidak dapat diubah dibandingkan dengan *blockchain* berjenis *permissioned* dan *private*, sehingga mahalnya dan lebih lambatya terhadap penggunaan dan penyimpanan yang terbatas.

2. Private Blockchain

Private Blockchain merupakan sistem yang bersifat kecil dan anggota pada sistem tersebut dapat terkontrol dan tertutup. Namun sistem ini dapat digunakan untuk kalangan pihak yang menggunakan bisnis yang tertentu. Sistem tersebut berjalan dengan cepat serta biaya yang cenderung lebih murah. Sistem tersebut dapat berjalan dikarenakan membutuhkan sistem *cryptocurrency*, namun untuk keamanan dari *private blockchain* tidak sama dengan *public*. Dikarenakan adanya validasi dari entitas pada node yang tidak sebanyak dari *public blockchain*. Tetapi, kapasistas dari penyimpanan pada *private blockchain* cenderung sangat besar.

3. Permissioned Blockchain

Permissioned Blockchain merupakan sistem yang bersifat besar dan tersebar seperti *public blockchain*, namun adanya aturan yang mengatur dari individu yang ingin mengakses terhadap sistem tersebut. Sehingga adanya entitas yang dapat mengontrolnya. Pada sistem ini dapat berjalan dengan adanya *cryptocurrency*, namun dengan biaya yang sangat murah. Adanya skala kebutuhan pada proyek serta volume terhadap transaksi dapat dibuktikan secara mudah. Sistem tersebut bersifat cepat dan rendah terhadap atensi, serta memiliki kapasitas penyimpanan yang sangat besar dibandingkan *Public Blockchain*. Sifat dari *source code* kepada sistem bersifat *open source* dan tidak *open source*.

2.6.2 Block

Blok adalah catatan transaksi yang dimasukkan. Pada setiap blok memiliki *header*, cap waktu, serta hash dari blok sebelumnya. Blok pertama, yang dapat disebut blok genesis, berisi informasi tambahan yang menggambarkan *blockchain* itu sendiri. *Hyperledger fabric* merupakan *blockchain* yang berjenis *permissioned*, blok genesis akan mencatat daftar anggota organisasi, sertifikat serta informasi yang dibutuhkan untuk mendukung transaksi. Selain itu identitas node pemesanan (*orderer node*) bertanggung jawab terhadap pemesanan transaksi serta pembuatan blok yang akan ditambahkan kedalam *blockchain*, blok juga memiliki penghubungan blok secara bersamaan dan memberikan kualitas integrasi yang baik, dan mencegah adanya potensi penyerangan yang dapat merusak data yang ada pada blok itu sendiri [33].

2.6.3 Desentralisasi

Blockchain merupakan teknologi yang terdesentralisasi. *Ledger* dapat terwujud dikarenakan adanya jaringan bersifat *peer-to-peer* yang dapat membentuk jaringan yang terdesentralisasi. Pengendalian pusat data yang hilang inilah yang muncul adanya konsep desentralisasi. Desentralisasi merupakan konsep memberikan untuk mengontrol kesemua node yang saling berkolaborasi untuk menjalankan fungsi pada jaringan. Dengan adanya desentralisasi dapat menghasilkan sebuah konsensus atau aturan terhadap otoritas terpusat [33].

2.6.4 Identitas

Smart contract atau *blockchain* membutuhkan adanya identitas agar dapat berfungsi terhadap sistem. Identitas adalah merepresentasikan entitas yang bisa berupa manusia ataupun komputer yang saling berpartisipasi pada proses transaksi pada *smart contract* atau administrasi *blockchain*. *Blockchain* memiliki kebutuhan *enterprise*, pelaku yang menjadi anggota yang dimasukan identitas untuk merepresentasikan sebagai pelaku melakukan transaksi. Identitas dilengkapi adanya *Private Key* serta *Public Key* untuk mendapatkan tanda tangan digital serta enkripsi data. Tanda tangan dapat mengotentifikasikan identitas dan mengizinkan pelaku untuk melakukan transaksi [33].

2.6.5 Transaksi

Transaksi merupakan salah satu proses kedalam sistem yang berintegrasi terhadap *blockchain*. Transaksi yang dilakukan oleh client diproses menggunakan *node* atau *peer*, jika diterima maka transaksi dapat ditambahkan pada rantai *blockchain*. Transaksi bisa seperti, pembaruan data, pembayaran layanan, pembelian produk. Transaksi pada *blockchain* selalu diberikan *time stamp* dan *ordered*, dan tidak dapat diubah. Transaksi yang ditandatangani dengan *Private Key* yang dikirimkan yang akan diproses serta dapat di autentifikasikan atas perizinan dari *blockchain* untuk pengiriman transaksi [33].

2.6.6 Node

Node berperan sebagai menciptakan jaringan yang terdistribusi. *Node* ini bisa berupa mesin virtual di *cloud* yang dapat dijadikan *host* yang berbentuk server fisik. *Node* dapat berupa perangkat keras yang berpartisipasi kedalam distribusi jaringan. *Node* dapat berupa *Internet of Things* (IoT) yang beroperasi sebagai jaringan yang desentralisasi kedalam industri. *Node* merupakan istilah umum yang mewakili titik akhir dalam jaringan. Jaringan yang dibentuk kedalam kabel atau nirkabel. Nama lain dari *node* itu sendiri adalah *peer*, sebagai contoh adanya jaringan yang bersifat *peer-to-peer* yang dimana akan dirancang sebagai aplikasi terdesentralisasi. Dalam *Hyperledger Fabric*, istilah kata *peer* dapat mewakili *node* koputasi yang dapat mengeksekusi pengesahan dan komitmen kedalam transaksi serta dapat *hosting* pada *ledger* [33]. *Hyperledger Fabric* pada *node* memiliki identitas yang dapat memungkinkan untuk *node* dapat beroperasi terhadap sistem. Berikut beberapa klasifikasi dari *node* kedalam jaringan *Hyperledger Fabric*:

1. *Client*, permulaan siklus hidup terhadap transaksi pada *Fabric*.
2. *Endorser*, bertugas untuk bertanggung jawab atas proses pelaksanaan logika bisnis yang dapat dilapis kedalam program yang disebut *chaincode*.
3. *Orderer*, berpartisipasi kedalam algoritma konsensus untuk memberikan keputusan pesanan transaksi.
4. *Committer*, memberikan hasil *ledger* dari *orderer* serta menerapkan logika validasi sebagai penetapan validasi terhadap transaksi yang akan dipesan.

2.6.7 Chaincode

Smart contract yang bias akita kenal, namun pada *Hyperledger Fabric* dapat disebut sebagai *Chaincode*. *Chaincode* merupakan suatu bagian dari kode pengimplementasian dari logika aplikasi dan berjalannya selama fase eksekusi. *Chaincode* yang terpasang pada *peer* dari sebuah *channel* yang membutuhkan proses eksekusi komputasi yang tentunya melalui anggota yang terotorisasi. Setiap *chaincode* menentukan keputusan bijak terhadap *endorsement* yang dievaluasi oleh comiter pada face validasi. Kebijakan khusus tersebut dapat memungkinkan *chaincode* dapat melakukan pengesahan terhadap transaksi dalam pentuk serangkaian *peer* yang diperlukan sebagai pengesahan. *Chaincode* juga memiliki 2 fungsi yaitu *INIT* dan *INVOKE*, *Init* dapat digunakan untuk menginisialisai terhadap struktur data yang mungkin dapat diperlukan *chaincode* serta dapat dipanggil hanya sekali. *Invoke* merupakan titik yang dimana logika proses dapat disediakan oleh *chaincode*. *Chaincode* dapat menerima argumen, serta *chaincode* dapat memutuskan operasi apa yang akan dijalankan [33].

2.6.8 Ledger

Ledger merupakan bagian yang mendasar kedalam fungsi dari *blockchain* serta *Hyperledger Fabric*. *Ledger* dapat disimpan sebagai salinan identik terhadap *node* yang terbentuk jaringan. *Channel Fabric*, memberikan keterikatan terhadap *ledger*. *Ledger* memiliki 2 konsep yaitu *Blockchain* serta *World State*. *Blockchain* merupakan cakupan semua catatan yang dapat disimpan kedalam jaringan. *World State* adalah bagian dari *blockchain* yang mengacu kepada pasangan nilai kunci yang berisi data-data yang terbaru dan direkam kedalamnya. *Ledger Fabric* juga memiliki model *execute-order-validate*, yang dapat dibedakan dengan *ledger* pada umumnya sebagai *raw ledger* serta *validated ledger*. *Raw Ledger* dapat mewakili urutan pada transaksi yang dipesan melalui *ordering service*. *Validated Ledger* dapat mewakili urutan transaksi yang sudah divalidasi. Dari *ledger* yang sudah divalidasikan dapat diartikan sebagai *World State* [33].

2.6.9 Channel

Channel pada *blockchain Hyperledger Fabric* adalah konsorium yang terorganisasi kedalam kolaborasi untuk melakukan transaksi yang bertujuan untuk menentukan hasil yang diinginkan [33].

2.6.10 Hyperledger Fabric

Hyperledger Fabric merupakan salah satu proyek yang dinaungi oleh *Linux Foundation*. *Fabric* sendiri memperkenalkan arsitektur baru dalam *blockchain* itu sendiri yang bertujuan untuk ketahanan, fleksibilitas, aksibilitas, serta kerahaisaan. *Hyperledger Fabric* merupakan sistem *blockchain* yang pertama kali yang mendukung penerapan aplikasi terdistribusi dan dapat mencatat dalam bahasa pemograman standar sehingga dapat dieksekusi secara konsisten dengan banyaknya *node* yang ada.



Gambar 2. 5 Logo Hyperledger Fabric

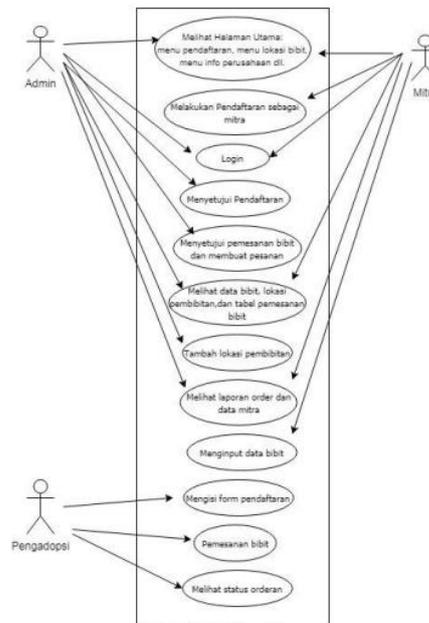
Sumber: <https://cn.hyperledger.org/fabric-logo-2>

2.7 Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah sebuah “Bahasa” yang sudah menjadi standarisasi untuk merancang dan mendokumentasi dari sistem perangkat lunak, UML juga digunakan untuk membuat suatu model perangkat lunak, diperkenalkannya UML sejak tahun 1997, dengan saat ini sudah dikembangkan menjadi Bahasa pemodelan yang baku (*de facto*) pada sebuah pengembangan perangkat lunak [34]. UML juga digambarkan kedalam beberapa diagram seperti *Use Case*, *Activity Diagram*, *Clas Diagram*, *Sequence Diagram*. Berikut adalah penjelasan dari UML itu sendiri:

2.7.1 Use Case Diagram

Use Case Diagram merupakan gambaran bagaimana *actor* yang sedang berinteraksi kepada sistem, serta membuat alur bisnis yang sudah didefinisikan pada Analisa sistem yang akan dijalankan. *Use Case Diagram* dapat digambarkan dengan *actor* dan *use case*. Aktor dapat digambarkan menjadi pelaku yang terlibat dalam penggunaan sistem, sementara itu *use case* merupakan gambaran dari sistem perangkat lunak yang akan dibangun. Berikut contoh dari Use Case Diagram pada gambar 2.6.

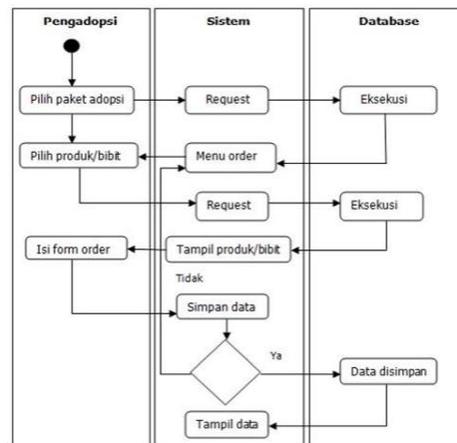


Gambar 2. 6 Use Case Diagram

Sumber: [Purba. 2020]

2.7.2 Activity Diagram

Activity Diagram Merupakan langkah-langkah cara kerja dalam sistem yang dirancang, sehingga terdapat masing-masing dari alur kerja sistem dimulai dari awal, kondisi yang akan terjadi pada saat eksekusi sistem samapai akhir sistem tersebut yang berjalan. Berikut contoh dari *Activity Diagram* pada gambar 2.7.

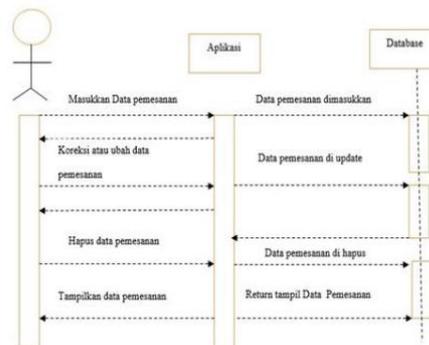


Gambar 2. 7 Activity Diagram

Sumber: [Purba. 2020]

2.7.3 Sequence Diagram

Sequence Diagram gambaran dari objek yang berinteraksi dan disusun dalam urutan waktu / kejadian pada sebuah proses. Berikut contoh dari *Sequence Diagram* pada gambar 2.8.

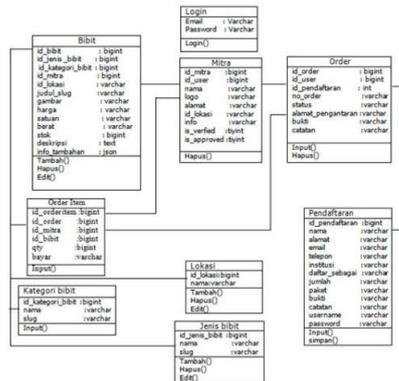


Gambar 2. 8 Sequence Diagram

Sumber: [Purba. 2020]

2.7.4 Class Diagram

Class Diagram merupakan sebuah spesifikasi yang menghasilkan sebuah objek berupa inti dari sebuah pengembangan dan perancangan berorientasi objek. Kelas (*class*) digambarkan sebuah keadaan (atribut/properti) suatu sistem, serta memberikan layanan untuk memanipulasikan keadaan tersebut (metode/fungsi). Berikut contoh dari Class Diagram pada gambar 2.9.



Gambar 2. 9 Class Diagram

Sumber: [Purba. 2020]

2.8 Mern Stack

Mern Stack adalah gaya mengoding untuk membuat atau membangun aplikasi website. *Mern stack* juga mempunyai 2 versi, yaitu versi yang memakai *Mysql* atau *MongoDB* sebagai database utama yang digunakan dalam membangun aplikasi bersifat website. Penulis menggunakan penelitian tersebut menggunakan *Mern stack* versi *MongoDB*, jadi karena itu singkatan *Mern Stack* merujuk pada *MongoDB*, *React*, *Express*, serta *Node*.



Gambar 2. 10 Mern Stack

Sumber: <https://www.rlogical.com/blog/for-what-reason-is-mern-stack-considered-the-best-for-developing-web-apps/>

2.9 Javascript

Javascript dipublikasikan pada tahun 1995 oleh Netscape. Awalnya *Javascript* dikenal dengan nama “*LiveScript*” yang berugas sebagai penyederhanaan terhadap pembuatan *browser*. *Javascript* adalah bahasa dengan model kumpulan skrip yang didalamnya memiliki arsip yang bertugas sebagai pembuatan manajemen serta pembuatan lain-lainya yang bertentang terhadap kejadian-kejadian internal dan skternal yang sangat diperlukan pada suatu asas informasi yang dapat menarik terhadap keputusan [35].

Javascript merupakan bahasa yang dipakai untuk membuat suatu program yang dapat digunakan sebagai dokumen *HTML*, yang dapat ditampilkan dalam *browser* menjadi interaktif. *Javascript* memberikan fungsional kedalam suatu web. Sehingga dapat membuat sebuah program yang sudah disajikan dengan menggunakan antarmuka [36].



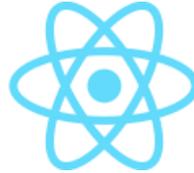
Gambar 2. 11 Logo Javascript

Sumber: <https://commons.wikimedia.org/wiki/File:JavaScript-logo.png>

2.10 React.Js

React.Js merupakan *library* bahasa pemrograman *javascript* yang diciptakan oleh *facebook* untuk membangun *user interface* terhadap sebuah *web application* yang bersifat *open source*, serta dapat membuat beberapa komponen *user interface* pada aplikasi yang akan dibangun secara kompleks [37].

React.js memiliki beberapa keunggulan yang diantaranya memberikan kecepatan, *simplicity*, dan *scalability*. *React.js* dapat memungkinkan pengembang dapat membangun sebuah komponen *UI* yang interaktif, *stateful*, & *reusable*. Pada *Model View Control* (MVC), *React.js* bertanggung jawab sebagai bagian *view* [38].



Gambar 2. 12 Logo React.Js

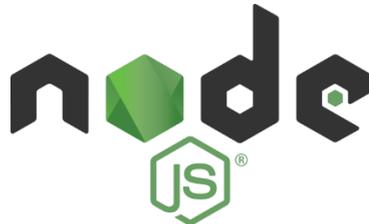
Sumber: [https://en.wikipedia.org/wiki/React_\(JavaScript_library\)](https://en.wikipedia.org/wiki/React_(JavaScript_library))

2.11 Express.Js

Express.js merupakan kerangka kerja atau *framework* yang terdapat pada *Node.js* yang mudah dikembangkan untuk mengembangkan aplikasi, web, *service API*, *routing*, serta *security*. *Express.js* juga berguna untuk penggunaan *design pattern* yang dapat disesuaikan dengan *framework* yang sangat *fleksible* serta *powerfull* [39].

2.12 Node.Js

Node.js merupakan *platform* yang pengembangan *open source* untuk mengesekusi dari sisi *server* kode pada *javascript*. *Node.js* berguna untuk mengembangkan aplikasi secara memerlukan koneksi yang terus-menerus dari *browser* kepada *server* serta sering digunakan untuk pembuatan aplikasi secara waktu yang nyata seperti *chat*, *news feed* dan web *push notification* [39].



Gambar 2. 13 Logo Node.Js

Sumber: https://commons.wikimedia.org/wiki/File:Node.js_logo.svg

2.13 Database

Database merupakan kumpulan informasi yang disusun serta kesatuan yang dapat disimpan kedalam perangkat keras secara sistematis, sehingga dapat terintegrasi dengan perangkat lunak yang akan dibangun. Dengan adanya sistem yang terhimpun dalam suatu database dapat menghasilkan informasi yang berguna [39].

2.13.1 MongoDB

MongoDB merupakan database *noSQL* yang bersifat *document based*, yang tidak memiliki tabel, kolom, ataupun baris. *MongoDB* hanya memiliki dokumen serta koleksi. Data yang akan disimpan kedalam database berupa file *JSON*. Sistem database pada *MongoDB* menggunakan *key-value*, dengan kata lain setiap dokumen dalam *MongoDB* dapat dipastikan memiliki *key* [39].



Gambar 2. 14 Logo MongoDB

Sumber: https://commons.wikimedia.org/wiki/File:Node.js_logo.svg

2.14 Leaflet.Js

Leaflet.Js adalah *library JavaScript* untuk berbasis web atau *mobile* dengan peta yang mudah untuk diintegrasikan terhadap peta yang bersifat interaktif, *Leaflet.Js* pun mempunyai beberapa fitur dalam pembuatan peta. *Leaflet.Js* juga membangun suatu sistem sebagai *plugin* yang dapat memperluas fungsional seperti penanda, *popup*, garis *overlay*, serta bentuk, *zoom*, *pan* [40].

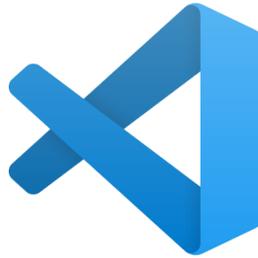


Gambar 2. 15 Logo Leaflet.Js

Sumber: https://wikimedia.org/wiki/File:Leaflet_logo.svg

2.15 Microsoft Visual Studio Code

Visual Studio Code merupakan editor kode yang ringan namun kuat untuk berjalan pada *desktop* serta tersedia pada *windows*, *macOS*, serta *Linux*, *Visual Studio Code* memberikan dukungan bawaan seperti *Javascript*, *TypeScript* serta *Node.js* dan mempunyai ekstensi yang kaya terhadap bahasa lain seperti *C++*, *C#*, *Java*, *Python*, *PHP*, *GO* serta dengan runtime seperti *.NET* dan *Unity* [39].



Gambar 2. 16 Logo Microsoft Visual Studio

Sumber: https://commons.wikimedia.org/wiki/File:Visual_Studio_Code_1.35_icon.svg

2.16 XAMPP

Pemograman aplikasi *XAMPP* yang berfungsi menjadi *server local* yang mengampu berbagai jenis data yang bersifat *website* dan dapat diproses pengembangan. *XAMPP* dapat menjadi kompilasi dari beberapa program. Fungsi pada *XAMPP* merupakan sebagai *server localhost*.



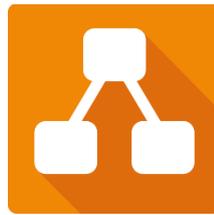
Gambar 2. 17 Logo XAMPP

Sumber: https://id.m.wikipedia.org/wiki/Berkas:Xampp_logo.svg

2.17 Draw.io

Draw.io merupakan sebuah *website* yang dapat mendisain khusus untuk menggambar suatu diagram secara *online*. Situs tersebut bisa diakses melalui *browser* yang didukung oleh *HTML*.

Draw.io ini digunakan untuk membuat diagram terkait dengan sistem yang akan dibangun seperti. *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, serta *Class Diagram*.

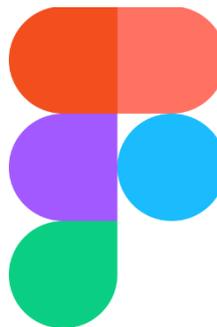


Gambar 2. 18 Logo Draw.io

Sumber: https://commons.wikimedia.org/wiki/File:Diagrams.net_Logo.svg

2.18 Figma

Figma merupakan alat desain *UI/UX* untuk membangun tampilan antarmuka aplikasi berbasis *website* atau *mobile apps*. Figma memudahkan beberapa designer untuk mengkolaborasikan pekerjaan tim secara bersamaan dalam dokumen yang sama dan dapat memberikan komentar, saran, serta mengubah rancangan desain dalam waktu yang bersamaan. Figma bersifat *real time* yang dimana setiap perubahan dalam tampilan akan tersimpan secara otomatis [41].



Gambar 2. 19 Logo Figma

Sumber: <https://commons.wikimedia.org/wiki/File:Figma-logo.svg>