

BAB II
TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Dalam penelitian ini menggunakan beberapa sumber pustaka sebagai pedoman dan perbandingan dalam penelitian yang akan dilakukan dan dapat dilihat perbandingan tinjauan pustaka pada tabel 2.1.

Tabel 2.1 Perbandingan Tinjauan Pustaka

No	Peneliti	Objek	Metode	Hasil Penelitian
1	Laila Septiana (2016)	Penyakit ISPA	<i>Certainty Factor</i>	Aplikasi yang dibangun dapat digunakan oleh pengguna untuk mendiagnosa penyakit Infeksi Saluran Pernafasan Atas (ISPA) sebelum melakukan pemeriksaan lebih lanjut ke dokter.
2	Yogi Permana, dkk (2017)	Penyakit Mata	<i>Certainty Factor</i>	Sistem pakar diagnosa penyakit mata menggunakan metode <i>certainty factor</i> berbasis Android yang telah dibuat memiliki tingkat keakuratan diagnosa sebesar 75% dengan rincian 15 penyakit dan 52 gejala

Tabel 2.1 Lanjutan

No	Peneliti	Objek	Metode	Hasil Penelitian
3	Muqorobin, dkk (2018)	Penyakit Ayam	<i>Certainty Factor</i>	Implementasi metode <i>certainty factor</i> telah mampu melakukan proses diagnosa penyakit ayam yang cukup dipercayai oleh <i>user</i> dengan didukung oleh nilai kepercayaan/CF yang dihasilkan yaitu 0,66 berdasarkan hasil pengujian sistem yang telah dilakukan
4	Yudi, dkk (2020)	Penyakit Gastritis	<i>Certainty Factor</i>	Hasil pengujian aplikasi sistem pakar diagnosis penyakit gastritis menggunakan olahan data kuesioner menunjukkan bahwa 86.8% lebih responden menjawab dengan jawaban positif atau dapat dikatakan aplikasi ini sudah berhasil dengan baik mengidentifikasi penyakit
5	Intan, dkk (2020)	Penyakit Hepatitis	<i>Certainty Factor</i>	Dengan adanya akurasi dengan pakar. Dari 150 sampel <i>rule</i> yang dipilih secara acak, 132 dinyatakan benar dan 18 <i>rule</i> dinyatakan tidak sesuai, sehingga dari pengujian tersebut didapatkan nilai akurasi sebesar 88%.

Tabel 2.1 Lanjutan

No	Peneliti	Objek	Metode	Hasil Penelitian
6	Usulan Penelitian (2017)	Penyakit pencernaan	Certainty Factor	Sistem Pakar Diagnosa Penyakit Pencernaan Pada Manusia Menggunakan Metode Certainty Factor

Penelitian tentang sistem pakar menggunakan metode *certainty factor* yang pernah dilakukan sebelumnya oleh Laila Septiana (2016) menyimpulkan bahwa aplikasi yang dibangun dapat digunakan oleh pengguna untuk mendiagnosa penyakit Infeksi Saluran Pernafasan Atas (ISPA) sebelum melakukan pemeriksaan lebih lanjut ke dokter. Aplikasi dibangun berbasis Android. Sistem pakar yang dibangun terdiri dari 10 penyakit dan 24 gejala.

Menurut Yogi Permana, dkk (2017) dalam penelitiannya menyimpulkan bahwa sistem pakar diagnosa penyakit mata menggunakan metode *certainty factor* berbasis Android yang telah dibuat memiliki tingkat keakuratan diagnosa sebesar 75% dengan rincian 15 penyakit dan 52 gejala. Aplikasi yang dibangun berbasis Android dapat digunakan untuk mendiagnosa penyakit mata dengan tingkat kepercayaan yang telah ditentukan oleh pakar terhadap gejala-gejala yang mempengaruhi probabilitas terjadinya suatu penyakit mata.

Menurut Muqorobin, dkk (2018) dalam penelitiannya menyimpulkan bahwa implementasi metode *certainty factor* telah mampu melakukan proses diagnosa penyakit ayam yang cukup dipercayai oleh *user* dengan didukung oleh nilai kepercayaan/CF yang dihasilkan yaitu 0,66 berdasarkan hasil pengujian sistem yang telah dilakukan. Alat yang digunakan dalam perancangan alur kerja data pada sistem menggunakan *Data Flow Diagram*. Metodologi penelitian menggunakan wawancara dan studi *literature* sehingga dapat menggumpulkan data yang lebih lengkap. Sistem Pakar ini akan diimplementasikan kedalam sebuah aplikasi *mobile*

berbasis Android dengan harapan memudahkan *user* dalam menggunakan karena dibuat kedalam aplikasi Android sehingga bisa di *install* dalam *handphone*. Hasil akhir berupa suatu informasi data penyakit dengan nilai *certainty factor*/kepastian.

Menurut Yudi, dkk (2020) dalam penelitiannya menyimpulkan bahwa hasil pengujian aplikasi sistem pakar diagnosis penyakit gastritis menggunakan olahan data kuesioner menunjukkan bahwa 86.8% lebih responden menjawab dengan jawaban positif atau dapat dikatakan aplikasi ini sudah berhasil dengan baik mengidentifikasi penyakit. Aplikasi sistem pakar berbasis Android ini terdiri dari 12 penyakit dan 12 gejala. Perancangan alur data pada sistem menggunakan pemodelan *use case* dan *activity diagram*.

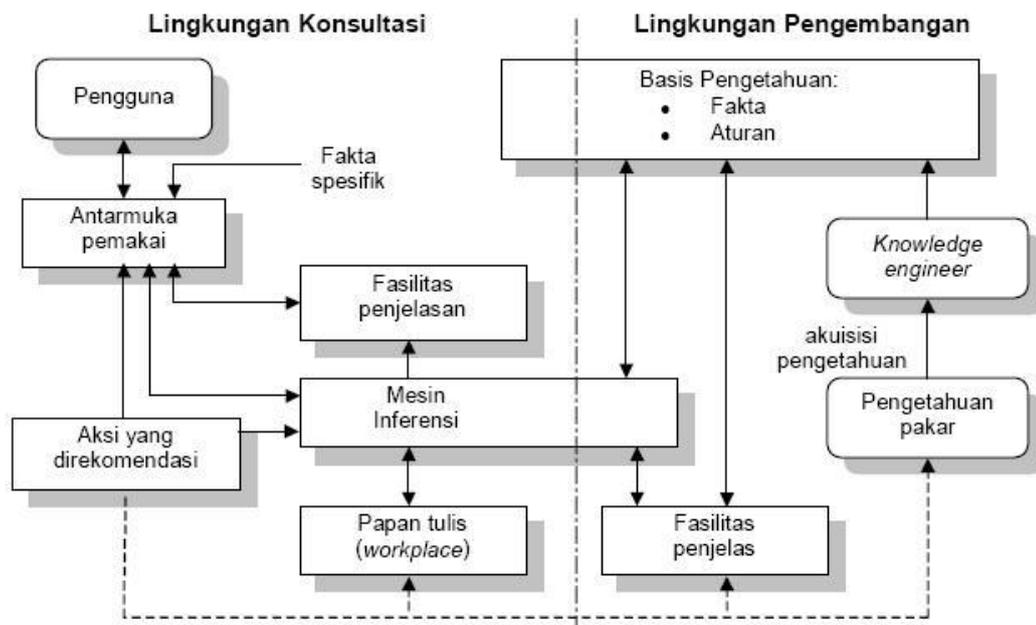
Menurut Intan, dkk (2020) dalam penelitiannya menyimpulkan bahwa penyebab penyakit hepatitis kian meningkat dikarenakan kurangnya kesadaran masyarakat akan penyakit hepatitis dan kurangnya inisiatif untuk berkonsultasi dengan dokter mengenai gejala-gejala yang dirasakan. Dalam penelitian ini dibuat sistem pakar berbasis Android untuk mendiagnosis penyakit hepatitis. Metode yang digunakan yaitu metode *certainty factor*. Berdasarkan hasil pengujian menggunakan *black box*, menunjukkan bahwa dengan adanya akurasi dengan pakar dari 150 sampel *rule* yang dipilih secara acak, 132 dinyatakan benar dan 18 *rule* dinyatakan tidak sesuai, sehingga dari pengujian tersebut didapatkan nilai akurasi sebesar 88%.

Pada penelitian ini membuat sebuah sistem pakar untuk diagnosa penyakit pencernaan pada manusia dengan menggunakan metode *certainty factor*. Dalam penelitian ini penyakit yang dapat dianalisa oleh sistem sebanyak lima penyakit. Dengan metode ini penghitungan yang dilakukan dengan cara memberikan nilai kepastian dan ketidakpastian terhadap gejala yang sudah ada. Dari nilai tersebut akan dikombinasikan dan dari hasil akhir nilai yang sudah di kombinasikan maka nilai kepastian akan dikurangi oleh nilai ketidak pastian, sehingga akan di dapat nilai keyakinan. *Output* dari sistem yang dibuat akan menampilkan gejala yang sudah dipilih dan menampilkan tiga penyakit dengan nilai tertinggi beserta nilai presentase.

2.2 Sistem Pakar

2.2.1 Pengertian Sistem Pakar

Menurut Kusrini (2008), sistem pakar adalah aplikasi berbasis komputer yang digunakan untuk menyelesaikan masalah sebagaimana yang dipikirkan oleh pakar. Pakar yang dimaksud di sini adalah orang yang mempunyai keahlian khusus yang dapat menyelesaikan masalah yang tidak dapat diselesaikan oleh orang awam. Menurut Aan (2019) dalam penelitiannya menyebutkan bahwa sistem pakar dirancang untuk menirukan keahlian seorang pakar dalam menjawab pertanyaan dan menyelesaikan suatu permasalahan baik di bidang kedokteran atau kesehatan, ekonomi, bisnis dan sebagainya. Sistem pakar sangat membantu untuk pengambilan keputusan, dimana sistem pakar dapat mengumpulkan dan menyimpan pengetahuan dari seseorang atau beberapa orang pakar dalam suatu basis pengetahuan (*knowledge base*) dan menggunakan sistem penalaran yang menyerupai seorang pakar dalam memecahkan masalah. Menurut Kusumadewi (2003), komponen-komponen yang terdapat dalam arsitektur sistem pakar dapat dilihat pada Gambar 2.1.



Gambar 2.1 Struktur Sistem Pakar

Menurut Aan (2019) secara garis besar, banyak manfaat yang dapat diambil dengan adanya sistem pakar, antara lain :

- a. Memungkinkan orang awam dapat mengerjakan pekerjaan para ahli.
- b. Dapat melakukan proses secara berulang dan otomatis.
- c. Menyimpan pengetahuan dan keahlian dari para pakar.
- d. Meningkatkan output dan produktivitas serta kualitas.
- e. Mampu mengambil dan melestarikan keahlian para pakar (terutama termasuk keahlian yang langka).
- f. Mampu beroperasi dalam lingkungan yang berbahaya.
- g. Memiliki kemampuan untuk mengakses pengetahuan dan bekerja dengan informasi yang kurang lengkap dan kurang pasti atau bahkan tidak lengkap dan tidak pasti.
- h. Meningkatkan kapabilitas sistem komputer.
- i. Sebagai media pelengkap dalam pelatihan.
- j. Meningkatkan kapabilitas dan efisiensi waktu yang dibutuhkan dalam memecahkan masalah.

Disamping memiliki beberapa manfaat, sistem pakar juga memiliki beberapa kelemahan, antara lain :

- a. Biaya yang diperlukan untuk pembuatan dan pemeliharaan aplikasi sistem pakar relatif mahal.
- b. Sulit dikembangkan, yang dikarenakan ketersediaan pakar di bidangnya yang terbatas.
- c. Sistem pakar tidak 100% bernilai benar.

2.2.2 Metode *Certainty Factor*

Menurut Saputra & Taman (2016) awal mula teori *certainty factor* (CF) diusulkan oleh Shortlife dan Buchanan pada 1975 untuk mengakomodasi ketidakpastian pemikiran seorang pakar, *certainty factor* menggunakan suatu nilai untuk mengasumsikan derajat keyakinan seorang pakar terhadap suatu data. Untuk mengakomodasi hal ini kita menggunakan *certainty factor* untuk menggambarkan tingkat keyakinan pakar terhadap masalah yang sedang dihadapi. *Certainty factor*

memperkenalkan konsep keyakinan dan ketidakyakinan yang kemudian diformulasikan kedalam rumus. Jika data yang diketahui adalah banyak hipotesa mempunyai banyak *evidence*, dan banyak CF *evidence*. Serta menggunakan rule KONJUNSI seperti if E1 AND E2 AND En, THEN H. Maka hasil yang dicari adalah CF Kombinasi terlebih dahulu CF kombinasi pada awalnya mencari 2 CF terlebih dahulu. Lalu hasil CF tersebut dihitung lagi dengan CF selanjutnya. Sampai semua CF selesai dihitung. Rumus CF kombinasi tergantung nilai CF, yaitu:

a. Jika kedua CF > 0, maka rumusnya adalah:

$$CF[H, E] = CF[lama] + CF[baru] (1 - CF[lama]) \dots\dots\dots(1)$$

b. Jika kedua CF < 0, maka rumusnya adalah:

$$CF[H, E] = CF[lama] + CF[baru] (1 + CF[lama]) \dots\dots\dots(2)$$

c. Jika kedua salah satu CF < 0, maka rumusnya adalah:

$$CF[H, E] = CF[lama] + CF[baru] / 1 - \min(CF[lama] | CF[lama]) \dots\dots\dots(3)$$

Dimana:

CF[H, E] = Cf dari hipotesis yang dipengaruhi *evidence*

CF[lama] = CF pertama atau CF hasil perhitungan sebelumnya

CF[baru] = CF kedua atau CF selanjutnya

Dalam implementasinya, CF(H) merupakan nilai kepastian yang diberikan oleh pakar dari pengetahuan yang dimilikinya terhadap suatu aturan, sedangkan CF(E) merupakan nilai kepercayaan yang diberikan oleh pengguna terhadap gejala yang dialaminya.

2.3 Penyakit Pencernaan Pada Manusia

2.3.1 Anatomi Sistem Pencernaan

Menurut Nuari (2015), saluran pencernaan manusia terdiri dari mulut, tenggorokan, kerongkongan, lambung, usus halus, usus besar, rektum, dan anus. Sistem pencernaan meliputi organ-organ yang terletak diluar saluran pencernaan, yaitu pankreas, hati dan kandung empedu).

2.3.2 Berbagai Penyakit Pencernaan Pada Manusia

Penyakit pencernaan merupakan suatu penyakit yang terjadi akibat tegangannya sistem pencernaan manusia. Penyebab utama dari penyakit gangguan pencernaan ini biasanya terjadi karena pola makan yang tidak teratur dan kurang sehat serta stres, infeksi bakteri, cacing dan bisa juga karena adanya gangguan pada lambung. Menurut Ashari dan Muniar (2016) banyak sekali penyakit yang berhubungan dengan gangguan pencernaan, diantaranya seperti penyakit diare, radang usus buntu, gastritis, tukak lambung, maag, dan lain sebagainya.

Pada penelitian ini, penyakit pencernaan pada manusia dibatasi pada penyakit sebagai berikut :

a. Diare

Diare adalah penyakit yang membuat penderitanya menjadi sering buang air besar, dengan kondisi tinja yang encer. Pada umumnya, diare terjadi akibat makanan dan minuman yang terpapar virus, bakteri, atau parasit. Penderita diare dapat meminum cairan elektrolit, guna mengganti cairan tubuh yang hilang akibat diare. Selama terjadi diare, konsumsi makanan yang lunak dan antibiotik atau obat anti diare.

b. Disentri

Disentri adalah infeksi pada usus yang menyebabkan diare yang disertai darah atau lendir. Kondisi ini umumnya berlangsung selama 3 hingga 7 hari, ditandai dengan kram perut, mual dan muntah, serta demam. Dalam penanganan disentri, pemberian antibiotik mungkin tidak diperlukan untuk mengatasi gejala ringan. Namun jika gejalanya bersifat sedang sampai berat, maka diperlukan antibiotik yang sesuai dengan bakteri penyebabnya. Selain itu bisa ditambah obat-obatan untuk mengurangi gejala diare. Hindari obat diare yang bersifat antimotilitas atau menurunkan pergerakan usus (seperti loperamide). Hal tersebut dapat

memperparah gejala yang ada. Jika sudah muncul tanda dehidrasi, maka diperlukan asupan cairan dengan oralit atau infus.

c. Gastritis

Gastritis merupakan penyakit pada lambung yang terjadi akibat peradangan dinding lambung. Pada dinding lambung atau lapisan mukosa lambung ini terdapat kelenjar yang menghasilkan asam lambung dan enzim pencernaan yang bernama pepsin. Untuk melindungi lapisan mukosa lambung dari kerusakan yang diakibatkan asam lambung, dinding lambung dilapisi oleh lendir (mukus) yang tebal. Apabila mukus tersebut rusak, dinding lambung rentan mengalami peradangan. Gastritis terjadi akibat peradangan pada dinding lambung. Dinding lambung tersusun dari jaringan yang mengandung kelenjar untuk menghasilkan enzim pencernaan dan asam lambung. Selain itu, dinding lambung juga dapat menghasilkan lendir (mukus) yang tebal untuk melindungi lapisan mukosa lambung dari kerusakan akibat enzim pencernaan dan asam lambung. Rusaknya mukus pelindung ini dapat menyebabkan peradangan pada mukosa lambung. Pengobatan yang diberikan kepada pasien oleh dokter, tergantung kepada penyebab dan kondisi yang memengaruhi terjadinya gastritis.

d. Tumor Gaster (Tumor Jinak Lambung)

Tumor jinak lambung adalah pertumbuhan yang bersifat nonkanker dalam lambung. Sifat jinak ini merujuk pada tumor yang tidak menyebar ke bagian tubuh lainnya. Karena jarang memicu gejala, tumor jinak lambung umumnya tidak membutuhkan penanganan medis. Terutama pada tumor yang berukuran kecil. Namun dokter tetap akan menganjurkan observasi secara berkala.

e. Apendisitis (usus buntu)

Penyakit usus buntu adalah peradangan yang terjadi pada usus buntu atau apendiks. Usus buntu merupakan organ berbentuk kantong kecil dan tipis, berukuran sepanjang 5 hingga 10 cm yang terhubung pada usus besar. Langkah pengobatan utama untuk penyakit usus buntu adalah melalui prosedur operasi pengangkatan usus buntu, atau yang dikenal dengan istilah apendektomi.

2.4 Aplikasi Android

2.4.1 Android

Penelitian ini membangun aplikasi berbasis Android. Pembangunan aplikasi android ini dibuat dengan menggunakan aplikasi Android Studio. Versi Android minimal yang digunakan dalam pembuatan aplikasi ini adalah versi Android 6.0 *Marsmallow*. Menurut *Google Developer Training Team* (2016), Android adalah sistem operasi dan *platform* pemrograman yang dikembangkan oleh Google untuk ponsel cerdas dan perangkat seluler lainnya (seperti tablet). Android bisa berjalan di beberapa macam perangkat dari banyak produsen yang berbeda. Android menyertakan kit *development* perangkat lunak untuk penulisan kode asli dan perakitan modul perangkat lunak untuk membuat aplikasi bagi pengguna Android. Android Juga menyediakan pasar untuk mendistribusikan aplikasi. secara keseluruhan, Android menyatakan ekosistem untuk aplikasi seluler.

a. Android 6.0 *Marsmallow*

OS Android ini mulai digunakan pada 5 Oktober 2015. OS Android ini memiliki fitur tambahan berupa support dari USB Type C. Android *Marsmallow* juga memiliki fitur sensor sidik jari serta peningkatan pada kinerja baterai.

b. Android 7.0 *Nougat*

Pada 22 Agustus 2016, Android *Nougat* resmi dirilis oleh Google. Pada Android *Nougat* terdapat beberapa fitur baru seperti *support* pada *Multi Window* yang cukup memuaskan para penggunanya.

c. Android 8.0 *Oreo*

Android *Oreo* dirilis pada 21 Agustus 2017. Pada Android *Oreo* tampaknya Google lebih terfokus pada peningkatan kecepatan dan juga efisiensi daya baterai. Bahkan kecepatan *Boot* pun dapat mencapai 2 kali lipat. Selain itu baterai pada sistem Android ini lebih tahan lama.

d. Android 9.0 *Pie*

Resmi pada 6 Agustus 2018, Google merilis Android *Pie* yang memiliki desain serta fitur yang sangat berbeda jika dibandingkan dengan versi sebelumnya. Fitur yang cukup menarik dari Android *Pie* ini adalah adanya *App Actions*. Fitur ini membuat Android dapat memprediksi tindakan yang akan dilakukan oleh penggunanya.

e. Android 10

Dimuali dari generasi ke 10 ini, Android memutuskan tidak lagi menggunakan nama makanan pada nama belakang OS-nya. Menurut pihak Android, ternyata nama dessert untuk Android tidak selalu dimengerti oleh masyarakat umum. Android 10 sendiri dirilis pada 3 September 2019. Android 10 saat ini masih menjadi OS Android yang banyak digunakan pengguna ponsel pintar berbasis Android di seluruh dunia.

f. Android 11

OS Android terbaru yang akan diluncurkan yaitu Android 11. Android 11 sendiri berdasar informasi dari techradar.com seharusnya dirilis pada 3 Juni 2020. Namun harus mundur akibat aksi protes yang sempat meluas di Amerika Serikat. Hingga kini, belum ada informasi resmi mengenai kapan akan dirilisnya Android 11 ini.

2.4.2 Android Studio

Android studio merupakan suatu aplikasi yang digunakan dalam pembuatan (*coding*) Android. Menurut Prabowo (2016) Android Studio merupakan sebuah *Integrated Development Environment (IDE)* untuk *platform* Android. Android Studio ini diumumkan pada tanggal 16 Mei 2013 pada Konferensi Google I/O oleh Produk Manajer Google, Ellie Powers. Android Studio bersifat *free* dibawah Apache License 2.0. Android studio awalnya dimulai dengan versi 0.1 pada bulan mei 2013, Kemudian dibuat versi beta 0.8 yang dirilis pada bulan juni 2014. Yang paling stabil dirilis pada bulan Desember 2014, dimulai dari versi 1.0. Berbasiskan JetBrains' IntelliJ IDEA, Studio didesain khusus untuk Android Development yang kini sudah bisa di *download* untuk Windows, Mac OS X, dan Linux.

2.4.3 Kotlin

Bahasa pemrograman yang digunakan dalam pembuatan aplikasi berbasis Android ini adalah Kotlin. Menurut artikel yang tertera pada *website* resmi Android yaitu developer.android.com, Kotlin adalah bahasa pemrograman modern melalui

pengetikan statis yang digunakan lebih dari 60% developer Android profesional untuk membantu meningkatkan produktivitas, kepuasan developer, dan keamanan kode. Sejak diluncurkan pada pagelaran Google I/O pada bulan Mei 2017, Kotlin menjadi bahasa kelas utama (*first class language*) dalam pembuatan aplikasi Android.

Menurut Azriel., dkk (2020), Kotlin adalah sebuah bahasa pemrograman dengan Statically typed (tipe statis) yang berjalan pada *platform Java Virtual Machine* (JVM). Kotlin menggunakan *compiler* LLVM yang artinya, dapat dikompilasi ke dalam kode JavaScript. Pengembang utamanya berasal dari tim programmer JetBrains yang bermarkas di Rusia. Bahasa pemrograman yang satu ini banyak diminati oleh para developer. Kotlin merupakan bahasa yang *powerfull* tentu cocok bagi developer dalam membuat aplikasi Android. Karena Kotlin adalah bahasa nomor satu untuk pengembangan aplikasi Android. Menurut Oktavijan (2017), kelebihan Kotlin adalah :

- a. *Concise* : Kotlin mampu mengurangi *boilerplate of code* atau tingkat kerumitan dari kode yang biasa kita tulis, ketika menggunakan bahasa Java.
- b. *Safe* : Kotlin mampu menjamin bahwa setiap *syntax* yang kita tulis secara proses kompilasi dapat mencegah kemungkinan terjadinya *error*, misalnya mampu mencegah terjadinya **NullPointerException** ketika kita *coding* menggunakan bahasa Java
- c. *Versatile* : Kotlin sejatinya sama seperti Java, karena memang kotlin itu sendiri di turunkan dari bahasa induknya, yaitu Java. Sehingga kotlin juga dapat di pakai dalam pengembangan aplikasi di *Web* maupun *Mobile*
- d. *Interoperable* : Kotlin tidak sama seperti bahasa Java turunan lainnya (misal; *Scala* ataupun *Clojure*) yang tidak dapat dijalankan bersamaan dengan kode yang kita tulis menggunakan Java. Kotlin mampu membaca kode lama atau library yang kita gunakan atau kita tulis dengan bahasa Java dan begitupun sebaliknya.

2.4.4 SQLite

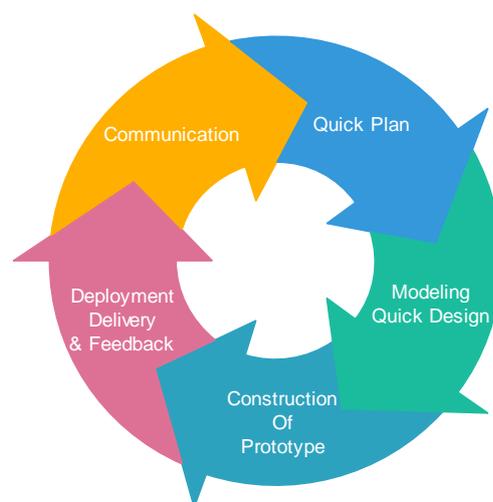
Basisdata yang digunakan dalam penyimpanan atau pengelolaan data adalah SQLite. Menurut Yudhanto dan Wijayanto (2019), SQLite adalah sebuah sistem

manajemen basis data relasional yang bersifat ACID-compliant, memiliki ukuran pustaka kode yang relatif kecil, ditulis dengan bahasa C. SQLite merupakan proyek yang bersifat public domain yang dikerjakan oleh D. Richard Hipp.

SQLite ini merupakan mesin database SQL yang tertanam pada sistem yang kita gunakan. Tidak seperti paradigma *client-server* umumnya, inti SQLite bukanlah sebuah sistem yang mandiri yang berkomunikasi dengan sebuah program secara keseluruhan. Oleh karena itu, protokol komunikasi utama yang digunakan adalah melalui panggilan API secara langsung melalui bahasa pemrograman. Mekanisme seperti ini tentunya membawa keuntungan karena dapat mereduksi *overhead*, *latency times*, dan secara keseluruhan lebih sederhana. Seluruh elemen basisdata (definisi data, tabel, indeks, dan data) disimpan sebagai sebuah *file*.

2.5 Metode Prototipe

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah prototipe. Menurut Pressman (2012), metode *prototype* suatu proses pembuatan *software* yang bersifat berulang dan dengan perencanaan yang cepat yang dimana terdapat umpan balik yang memungkinkan terjadinya perulangan dan perbaikan *software* sampai dengan *software* tersebut memenuhi kebutuhan dari sisi pengguna. Siklus atau ilustrasi dari metode prototipe dapat dilihat pada Gambar 2.2.



Gambar 2.2 Ilustrasi Model Prototipe (Sumber : Pressman, 2012)

Pembuatan prototipe dimulai dengan dilakukannya komunikasi antara tim pengembang perangkat lunak dengan para pelanggan. Tim pengembang perangkat lunak akan melakukan pertemuan-pertemuan dengan para stakeholder untuk mendefinisikan sasaran keseluruhan untuk perangkat lunak yang akan dikembangkan, mengidentifikasi spesifikasi kebutuhan apa pun yang saat ini diketahui, dan menggambarkan area-area dimana definisi lebih jauh pada iterasi selanjutnya merupakan keharusan. Iterasi pembuatan prototipe direncanakan dengan cepat dan pemodelan (dalam bentuk "rancangan cepat") dilakukan. Suatu rancangan cepat berfokus pada representasi semua aspek perangkat lunak yang akan dilihat oleh para pengguna akhir (misalnya rancangan antar muka pengguna (*user interface*) atau format tampilan). Rancangan cepat (*quick design*) akan memulai konstruksi pembuatan prototipe. Prototipe kemudian akan diserahkan kepada para stakeholder dan kemudian mereka akan melakukan evaluasi-evaluasi tertentu terhadap prototipe yang telah dilakukan sebelumnya, kemudian akhirnya akan memberikan umpan-balik yang akan digunakan untuk memperhalus spesifikasi kebutuhan. Iterasi akan terjadi saat prototipe diperbaiki untuk memenuhi kebutuhan dari para stakeholder, sementara pada saat yang sama memungkinkan kita untuk lebih memahami kebutuhan apa yang akan dikerjakan pada saat iterasi selanjutnya.

2.6 Metode Black Box Testing

Metode pengujian yang dipakai sebagai uji coba aplikasi yang dibangun adalah *black box testing*. Menurut Rosa dan Shalahuddin (2018), *black box testing* (pengujian kotak hitam) yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan pengeluaran perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Menurut Sidi (2015), *black box testing* berfokus pada spesifikasi fungsional dari perangkat lunak. Tester dapat mendefinisikan kumpulan kondisi *input* dan melakukan pengetesan pada spesifikasi fungsional program. Menurut Fa'atulo., dkk (2020) pengujian data sangat penting dilakukan untuk mengetahui kelemahan teknologi tersebut.

Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi

yang dibutuhkan. Kasus uji dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah, misalkan untuk kasus *login* maka kasus uji yang dibuat adalah:

- a. Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang benar.
- b. Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang salah.