

BAB II

TINJAUAN PUSTAKA

2.1 Pengertian Aplikasi

(Syani & Werstantia, 2019: 88) aplikasi adalah sebuah perangkat lunak yang berisi sebuah *coding* atau perintah yang dimana bisa diubah sesuai dengan keinginan.

(Sari, 2017: 83) aplikasi adalah sebuah perangkat lunak yang dimana tujuannya adalah agar bisa melayani setiap aktivitas komputerisasi yang dilakukan oleh pengguna.

(Dinata et al., 2015: 128) aplikasi adalah penerapan, menyimpan sesuatu baik berupa data, permasalahan, pekerjaan kedalam suatu sarana ataupun media yang bisa digunakan untuk diterapkan menjadi sebuah bentuk yang baru.

2.2 Pengertian Prediksi

Prediksi hampir sama dengan ramalan atau perkiraan. Menurut kamus besar bahasa Indonesia, prediksi merupakan hasil dari kegiatan memprediksi atau meramal atau memperkirakan. Prediksi dapat berdasarkan metode ilmiah maupun subjek belaka. Kesimpulan pengertian prediksi secara istilah akan tergantung pada konteks ataupun permasalahannya.

2.3 Metode Naïve Bayes

(Suntoro et al., 2018: 37) Algoritma Naïve Bayes adalah salah satu algoritma klasifikasi berdasarkan Native Bayesian pada statistika. Algoritma Naïve Bayes dapat digunakan untuk memprediksi probabilitas keanggotaan suatu kelas.

Native Bayesian menghitung nilai *posterior probability* $P(H|X)$ menggunakan probabilitas $P(H)$, $P(X)$, dan $P(X/H)$, dimana nilai X adalah data testing yang kelasnya belum diketahui. Nilai H adalah hipotesis data X yang merupakan satu kelas yang lebih spesifik. Nilai $P(X/H)$ atau disebut juga *likelihood*, adalah probabilitas hipotesis X berdasarkan kondisi H . Nilai $P(H)$ atau disebut juga dengan *prior probability* adalah probabilitas hipotesis H . Sedangkan nilai $P(X)$ yang disebut juga *predictor prior probability*, adalah probabilitas X .

$$P = \frac{P(X|H) \times P(H)}{P(X)}$$

“Persamaan Nominal”

Dimana:

P : probabilitas

X : data dengan *class* yang belum diketahui

H : hipotesis data merupakan suatu *class* spesifik

P(X|H) : probabilitas X berdasarkan kondisi pada hipotesis H

P(H) : probabilitas hipotesis H (prior probabilitas)

P(X) : probabilitas X

Algoritma Naïve Bayes sangat cocok untuk melakukan klasifikasi pada dataset bertipe nominal. Untuk data set bertipe nominal.

Langkah-langkah algoritma Naïve Bayes adalah sebagai berikut:

1. Siapkan dataset
2. Hitung jumlah kelas pada data training.
3. Hitung jumlah kasus yang sama dengan kelas yang sama.
4. Kalikan semua hasil sesuai dengan data testing yang akan dicari kelasnya.
5. Bandingkan hasil per kelas, nilai tertinggi ditetapkan sebagai kelas baru.

2.4 Bahasa Pemrograman Yang Digunakan

(Henry Februariyanti, 2012: 127) komputer membutuhkan software untuk beroperasi dan membutuhkan sistem operasi atau program-program untuk membuat komponen-komponen komputer bekerja secara baik. Software juga sering digunakan digunakan untuk menunjukkan semua program yang dapat dipakai dalam sistem komputer. Dalam pengertian yang sempit, istilah ini menunjukkan pada sebuah program yang dapat mempermudah pemakai dari berbagai jenis komputer untuk mendayagunakan hardware yang baik. Untuk merancang dan membangun aplikasi ini pembuat membutuhkan software- software penunjang untuk memaksimalkannya antara lain.

2.4.1 PHP (Hypertext Preprocessor)

(Henry Februariyanti, 2012: 128) PHP *hypertext pre processor* atau sering disebut PHP merupakan bahasa pemrograman berbasis server-side yang dapat melakukan parsing scrift php menjadi scrift web sehingga dari sisi client menghasilkan suatu tampilan yang menarik. PHP merupakan pengembangan dari FI atau form interface yang dibuat oleh Rasmus Lerdoff pada tahun 1995.

Berbeda dengan HTML, kode PHP tidak diberikan secara langsung oleh server ketika ada permintaan atau request dari sisi client namun dengan cara pemrosesan dari sisi server. Kode PHP disisipkan pada kode HTML. Perbedaan dari kode (scrift) HTML dan PHP yaitu setiap kode PHP ditulis selalu diberi tag pembuka yaitu `<?php` dan pada ahir kode PHP diberi tag penutupnya yaitu `?>`.

Adapun kelebihan dan kekurangan bahasa PHP antara lain sebagai berikut:

Tabel 2.1 Kelebihan dan kekurangn PHP

| No. | Kelebihan | Kekurangan |
|-----|--|---|
| 1. | PHP menjadi populer karena kesederhanaannya dan kemampuannya dalam menghasilkan berbagai aplikasi web seperti <i>counter</i> , sistem artikel/ CMS, <i>e-commerce</i> , <i>bulletin board</i> , dll. | Tidak detail untuk pengembangan skala besar. |
| 2. | PHP adalah salah satu bahasa <i>server side</i> yang didesain khusus untuk aplikasi web. | Tidak detai untuk pengembangan skala besar. |
| 1. | PHP termasuk dalam <i>open souce Product</i> . | Tidak memiliki sistem pemrograman berorientasi objek yang sesungguhnya. |

| | | |
|----|---|---|
| 2. | Aplikasi PHP cukup cepat dibandingkan dengan aplikasi CGI dengan <i>perl</i> atau <i>python</i> bahkan lebih cepat dibandingkan dengan ASP maupun Java dalam berbagai aplikasi web. | Tidak memisahkan antara tampilan dengan <i>logic</i> dengan baik. |
|----|---|---|

2.4.2 MySQL

(Henry Februariyanti, 2012: 128) *MySQL* adalah sebuah program database *server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat, *multi user* serta menggunakan perintah standar SQL (*structured Query Language*) *MySQL* juga telah mendukung bahasa pemrograman berfitur API seperti java sehingga memudahkan para programmer java untuk berkoneksi dengan menggunakan *MySQL*.

Berikut beberapa keistimewaan yang dimiliki *MySQL* antara lain:

- a. *Portabilitas* yaitu *MySQL* dapat berjalan secara stabil pada berbagai macam sistem operasi.
- b. *Open Source* yaitu *MySQL* di distribusikan secara gratis.
- c. *Multi User* yaitu *MySQL* dapat digunakan dengan banyak user tanpa memiliki masalah.
- d. *Performance Turning* yaitu *MySQL* memiliki tipe kolom yang sangat kompleks.
- e. Perintah dan fungsi yaitu *MySQL* operator dan fungsi yang secara penuh perintah *Select* dan *Where* dalam *query*.
- f. Keamanan yaitu *MySQL* memiliki beberapa lapisan keamanan dengan sistem perizinan yang detail serta sandi yang terenkripsi.
- g. Skalabilitas dan pembatasan yaitu *MySQL* mampu menangani *database* berskala besar dengan jumlah *record* lebih dari 50 juta, table lebih dari 60 ribu dan baris lebih dari 5 milyar.
- h. *Konektivitas* yaitu *MySQL* dapat melakukan koneksi dengan klien dengan menggunakan *TCP/IP*, *UNIX* dan *Named Pipes*.

- i. Lokalisasi yaitu *MySQL* mendeteksi kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa meskipun bahasa Indonesia belum masuk didalamnya
- j. Antarmuka yaitu *MySQL* memiliki antarmuka yang mendukung berbagai bahasa pemrograman yang menggunakan fungsi API.
- k. Klien dan peralatan yaitu *MySQL* dilengkapi dengan berbagai *tool* yang dapat digunakan secara *online*.
- l. Struktur table yaitu *MySQL* memiliki struktur table yang *fleksibel* dalam menangani *alter table* jika dibandingkan dengan bahasa pemrograman lainnya.

2.4.3 XAMPP

(Henry Februariyanti, 2012: 129) XAMPP adalah sebuah software *web server* apache yang didalamnya sudah tersedia database *server* *MySQL* dan dapat mendukung pemrograman PHP. XAMPP merupakan *software* yang mudah digunakan, gratis dan mendukung instalasi di *Linux* dan *Windows*. Keuntungan lainnya adalah cuma menginstal satu kali sudah tersedia *Apache Web Server*, *MySQL Database Server*, *PHP Support* (PHP 4 dan PHP 5) dan beberapa *module* lainnya.

2.5 Website

2.5.1 Pengertian Website

(Arkhiansyah & Komputer, 2018: 58) *Website* adalah kumpulan halaman-halaman. Yang digunakan untuk menampilkan informasi teks gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masingdihubungkan dengan jaringan-jaringan halaman.

2.5.2 Web Server

Suatu aplikasi yang jalan pada suatu situs dan bertanggung jawab untuk merespon permintaan file dari *web browser*. Dalam melakukan permintaan suatu halaman pada suatu situs *web*, *browser* melakukan koneksi ke suatu *server* dengan protocol HTTP. *Server* akan menanggapi koneksi tersebut dengan mengirimkan isi file yang diminta dan memutuskan koneksi tersebut. Web Browser kemudian memformat informasi yang didapat dari *server*. Pada bagian *server*, *browser* yang berbeda dapat melakukan koneksi pada *server* yang sama untuk memperoleh informasi yang sama. Dalam memberikan halaman yang diminta *web server* dapat melakukan kerja sama dengan server lain seperti Aplikasi *server*.

2.5.3 Web Browser

Software yang dijalankan pada komputer pengguna (*client*) yang meminta informasi dari *server web* dan menampilkannya sesuai dengan *file* data itu sendiri. Tugas utama dari *web browser* adalah mendapatkan dokumen dari *web*.

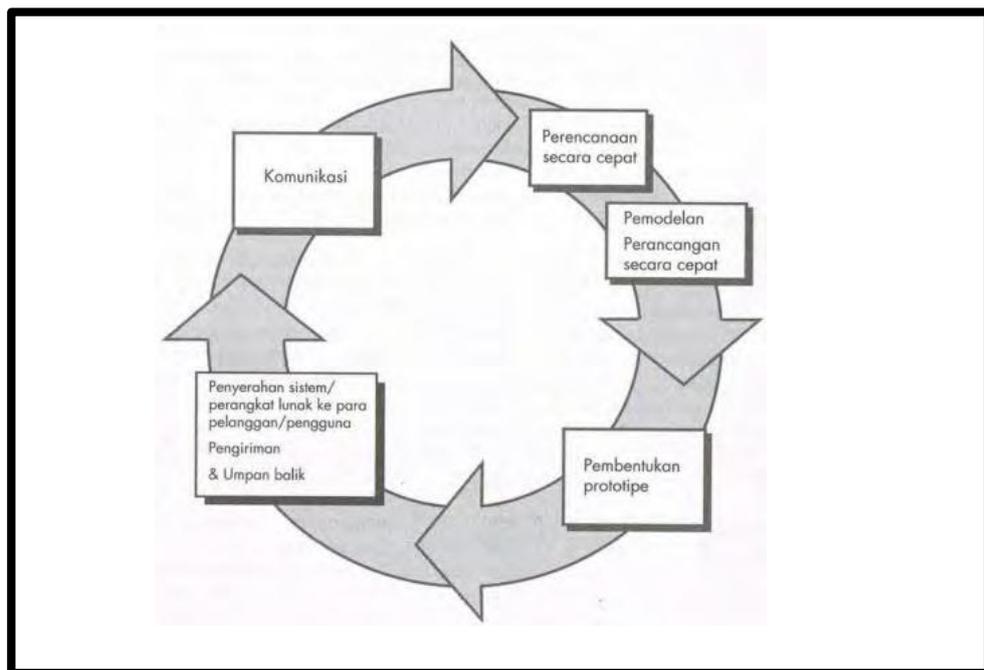
2.6 Metode Pengembangan Perangkat Lunak

2.6.1 Prototype

(Roger S. Pressman, 2012: 51) metode *prototype* adalah proses pembuatan model sederhana *software* yang mengizinkan pengguna memiliki gambaran dasar tentang program serta melakukan pengujian awal. *Prototype* memberikan fasilitas bagi pengembang dan pemakai untuk saling berinteraksi selama proses pembuatan, sehingga pengembang dapat dengan mudah memodelkan perangkat lunak yang akan dibuat. Metode ini cocok digunakan untuk mengembangkan sebuah perangkat lunak yang dikembangkan kembali. Metode ini dimulai dengan pengumpulan kebutuhan pengguna. Kemudian membuat sebuah rancangan kilat yang selanjutnya akan dievaluasi kembali sebelum di produksi secara benar. *Prototype* bukanlah merupakan sesuatu yang lengkap, tetapi tetapi sesuatu yang harus dievaluasi dan dimodifikasi kembali. Segala perubahan dapat terjadi pada saat *prototype* dibuat untuk memenuhi kebutuhan pengguna dan saat yang sama memungkinkan pengembangan untuk lebih memahami kebutuhan pengguna secara baik

Berikut adalah tahapan dalam metode *prototype*:

1. Komunikasi dan bertemu antara *user* dan *developer* untuk mengidentifikasi dan pengumpulan data awal untuk membangun sistem serta mengidentifikasi kebutuhan yang diinginkan oleh *user*.
2. Rancangan cepat pembuatan desain secara umum untuk dilakukan pengembangan.
3. Pembuatan *prototype* dilakukan untuk pengujian dan penyempurnaan.
4. Evaluasi *prototype*, melakukan evaluasi terhadap perkembangan *prototype* dan melakukan analisa yang sesuai dengan kebutuhan *user*.
5. Perbaiki *prototype*, pra pembuatan perangkat berdasarkan hasil dari evaluasi *prorotype*.
6. Produksi akhir, melakukan produksi perangkat yang sebenarnya untuk dapat digunakan oleh *user*.



Gambar 2.1 Diagram Prototype (Sumber: Pressman (2012 : 51))

Penjelasan dari gambar 2.1 diatas adalah sebagai berikut:

Tahap pertama adalah komunikasi dan pengumpulan data awal yaitu tahap suatu perencanaan yang dilakukan, mulai dari menciptakan dan melaksanakan proses untuk memastikan bahwa perencanaan tersebut berkualitas tinggi, terpercaya, efisiensi biaya. Tahap kedua adalah perancangan secara cepat yaitu analisis terhadap

kebutuhan pengguna. Tahap ketiga adalah pemodelan perancangan secara cepat yaitu pembuatan desain secara umum untuk selanjutnya dikembangkan kembali. Tahap empat adalah pembentukan *prototype* adalah pembuatan perangkat *prototype* termasuk pengujian dan penyempurnaan tahap kelima adalah penyerahan sistem dan umpak balik adalah tahap penyerahan sistem ke pengguna dan umpan balik.

2.7 Unified Modeling Language (UML)

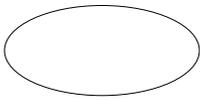
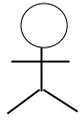
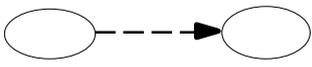
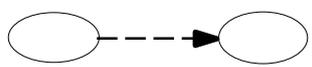
(A.S & Shalahuddin, 2015: 188) *unifed modelling language* (UML) adalah salah satu standar bahasa yang banyak digunakan didunia industry untuk mendefinisikan *requirement*, mmbuat aplikasi dan desain, serta menggambar arsitektur dalam pemrograman berorientasi objek.

Dengan demikian, penulis dapat mengutarakan bahwa metode *UML (Unified Modeling Language)* merupakan sebuah metode atau sebuah bahasa yang digunakan dalam menterjemahkan, menjelaskan, memodelkan, mendefinisikan, suatu sistem dengan bentuk symbol-simbol tertentu yang bertujuan untuk memberikan penjelasan-penjelasan detail dari sebuah sistem.

2.7.1 Use Case Diagram

(A.S & Shalahuddin, 2015: 195) menguraikan *use case* diagram merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem infomasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

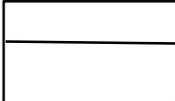
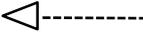
Table 2.2 Simbol *Use Case Diagram*

| Simbol | Keterangan |
|--|---|
| <p><i>Use Case</i></p>  | Menggambarkan bagaimana seseorang akan menggunakan atau memnfaatkan sistem. |
| <p>Actor</p>  | Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri. |
| <p>Asosiasi</p>  | Komunikasi antara <i>use case</i> dan aktor yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor. |
| <p>Generalisasi</p>  | Sebagai penghubung antara aktor- <i>use case</i> atau <i>use case-use case</i> . |
| <p><< Include >></p>  | <i>Include Relationship</i> (relasi cakupan) Memungkin suatu <i>use case</i> untuk menggunakan fungsionalitas yang disediakan oleh <i>use case</i> yang lainnya. |
| <p><< Extend >></p>  | <i>Extend Relationship</i> Memungkinkan relasi <i>use case</i> memiliki kemungkinan untuk memperluas fungsionalitas yang disediakan oleh <i>use case</i> yang lainnya. |

2.7.2 Class Diagram

Class diagram adalah sebuah klasifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan memanipulasi keadaan tersebut (metoda/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi dan lain-lain.

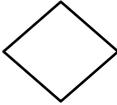
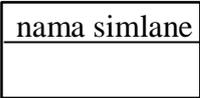
Tabel 2.3 Simbol *Class Diagram*

| SIMBOL | NAMA | KETERANGAN |
|---|-------------------------|---|
|  | <i>Generalization</i> | Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>) |
|  | <i>Nary Association</i> | Upaya untuk menghindari asosiasi dengan lebih dari 2 objek. |
|  | <i>Class</i> | Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama. |
|  | <i>Collaboration</i> | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor. |
|  | <i>Realization</i> | Operasi yang benar-benar dilakukan oleh objek. |
|  | <i>Dependency</i> | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri. |
|  | <i>Association</i> | Apa yang menghubungkan antara objek satu dengan objek lainnya. |

2.7.3 Activity Diagram

Activity Diagram menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh system.

Table 2.4 Simbol *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. |
| percabangan  | Asosiasi percabangan dimana ada pilihan aktivitas lebih dari satu. |
| Penggabungan  | Asosiasi penggungan dimana lebih dari satu aktivitas digabungkan menjadi satu. |
| Status Akhir  | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |
| <i>Swimlane</i>  | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas. |

2.8 Pengujian untuk menguji perhitungan pada data *training* dengan menggunakan data *testing*. Pada tahap pengujian ini dilakukan perhitungan tingkat *precision*, *recall*, *accuracy* untuk menguji tingkat akurasi aplikasi yang dibangun.

a. *Precision* adalah perhitungan gambaran proporsi kasus positif yang benar dan dirumuskan dalam persamaan 2: (Vafeiadis et al., 2015) $precision = \frac{TP}{TP+FP}$

b. *Recall* adalah perhitungan gambaran proporsi kasus positif yang diidentifikasi benar dan dirumuskan pada persamaan 3 :

$$Recall = \frac{TP}{TP + FN}$$

c. *Accuracy* adalah perhitungan akan proporsi dan jumlah total prediksi yang benar dan dirumuskan pada persamaan 4:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Keterangan :

TP : *true positive*

TN : *true negative*

FP : *false positive*

FN : *false negative*

(Badan et al., 2013) menguraikan bahwa prediksi kelahiran pada ibu hamil untuk mengurangi Angka Kematian Ibu (AKI) dan Angka Kematian Bayi (AKB) menggunakan metode Naïve Bayes. Penelitian ini menggunakan beberapa variable yaitu karakteristik yang dialami ibu hamil diantara lain, usia pada ibu, tinggi badan, jumlah Hb, tekanan darah, riwayat kehamilan dan penyakit bawaan. Hasil dari penelitian ini adalah system klasifikasi dengan bahasa R dengan *output* prediksi ibu hamil mengalami kelahiran yang beresiko dan kelahiran normal.

(Marlina et al., 2017) menguraikan bahwa dengan penggunaan Naïve Bayes Classifier dalam system pakar yang dibangun. Penelitian ini bertujuan untuk mempermudah dalam mendiagnosa penyakit ISPA dengan menggunakan konversi hasil suara pasien.

(Lutfiansyah, 2016) menguraikan bahwa dengan menggunakan metode naïve bayes dalam sistem pendukung keputusan pemilihan jurusan berdasarkan kuisisioner minat bakat. Penelitian ini tingkat akurasi yang digunakan masih kurang karena sedikitnya data yang dipakai.

Berdasarkan pustaka dari hasil penelitian diatas, alasan penulis untuk menggunakan metode naïve bayes yaitu naïve bayes merupakan metode klasifikasi dengan rumus yang sederhana serta metode naïve bayes memiliki tingkat akurasi yang cukup tinggi dibandingkn dengan metode lainnya. Dengan memprediksi usia kelahiran bayi ini diharapkan dapat membantu mengurangi bahaya kematian pada bayi akibat usia kelahiran yang sangat muda maupun sangat tua serta mengantisipasi aslinya sejak dini.