

BAB II

LANDASAN TEORI

1.1 Tinjauan Pustaka

Tabel 1.1 *State Of The Art*

No	Peneliti	Masalah	Metode	Hasil
1	Sfenrianto, Indah Purnamasari dan Rizal Broer (2016)	Banyak penawaran produk yang tidak sesuai dengan pelanggan	<i>Naïve Bayes</i>	Akurasi yang dihasilkan algoritma <i>Naïve Bayes</i> yaitu 85.08%, dengan kesalahan klasifikasi 14,92%. Jika digabungkan dengan Metode <i>Particle Swarm Optimization</i> dapat meningkatkan akurasi sebesar 89.31%.
2	Ihsan A.Abu Amra dkk (2017)	Tidak tepatnya dalam menentukan proses kinerja pada siswa	<i>K-Nearest Neighbor</i> dan <i>Naïve Baiyes</i>	Akurasi yang dihasilkan pada prediksi kinerja siswa yaitu 93.17% dengan menggunakan data pada tahun 2015. Dengan mengambil atribut pekerjaan ayah, jenis kelamin, dan nilai rata-rata.
3	Puput Shinta Dewi (2016)	Pengunduran Diri Peserta Didik	<i>Naive Bayes</i>	Metode yang tepat untuk memprediksi pengunduran diri peserta didik dengan menggunakan 100 data testing, yang menghasilkan 77 data peserta didik atau 77% data terprediksi bertahan sedangkan 23 data peserta didik atau 23% data terprediksi mengundurkan diri
4	Muhammad Efendi (2014)	Pengunduran Diri Calon Mahasiswa Baru di Universitas Dian Nuswantoro Semarang	<i>Naive Bayes</i>	Metode <i>Naive Bayes</i> menghasilkan 78% akurasi prediksi dengan data yang digunakan 1147 pelajar

Tabel 1.2 *State Of The Art* (Lanjutan)

No	Peneliti	Masalah	Metode	Hasil
5	Hera Wasiati dan Dwi Wijayanti (2014)	Menentukan Kelayakan Calon Tenaga Kerja Indonesia	<i>Naive Bayes</i>	Hasil pengujian dengan data sebanyak 542 dengan 362 sebagai data training dan 180 sebagai data tes, sehingga menghasilkan pola sebesar 73,89% dan errornya 26,11% atau 133 layak diterima dan 47 tidak layak diterima
6	Diana Septiari (2016)	Mencari Kelayakan Calon Pendorong Darah	<i>Naive Bayes</i>	Hasil untuk menentukan Calon Pendorong Darah yaitu memiliki tingkat akurasi sebesar 81,6% dari 60 data yang digunakan.
7	Irma Apriliani (2017)	Untuk Memprediksi Perceraian di Pengadilan Agama Cimahi	<i>Naive Bayes, K-Nearest Neighbor</i>	Metode yang memiliki tingkat akurasi yang paling besar adalah <i>Naive Bayes</i> dengan nilai akurasi 72,5% sedangkan <i>K-Nearest Neighbor</i> memiliki nilai akurasi sebesar 57,5%. Dengan menggunakan 20 data testing dan 130 data training.

- a. Pada tahun 2016, Sfenrianto, Indah Purnamasari dan Rizal Broer pada jurnal yang dibuatnya dengan judul “*Naive Bayes classifier algorithm and Particle Swarm Optimization for classification of cross selling (Case study: PT TELKOM Jakarta)*” banyaknya tawaran pelanggan yang tidak sesuai, membuat pemasaran tidak optimal. Lalu dilakukan klasifikasi untuk menempatkan penawaran suatu produk pelanggan yang sesuai, dengan menggunakan *Naive Bayes* menghasilkan akurasi 85.08% dan Metode *Particle Swarm Optimasi* menghasilkan akurasi 89.31% [8].
- b. Pada tahun 2017, Ihsan A.Abu Amra dkk dengan judul jurnal “*Students Performance Prediction Using KNN and Naive Bayesian*” untuk membantu

- kementrian pendidikan dalam meningkatkan kinerja siswa dengan menggunakan data pada tahun 2015. Lalu menghasilkan akurasi 93.6% [6].
- c. Pada tahun 2016, Puput Shinta Dewi dalam skripsi dengan judul “Sistem Prediksi Pengunduran Diri Peserta Didik Menggunakan Metode *Naive Bayes Classifier*” untuk mengklasifikasiannya menggunakan 100 data yang meliputi jarak tempuk, usia, pekerjaan, pendidikan terakhir, jenis kelamin dan program khusus sehingga menghasilkan 77 peserta didik atau 77% di nyatakan berhatan dan 23 data peserta didik atau sebesar 23% mengundurkan diri [7].
- d. Pada tahun 2014, Muhammad Efendi pada jurnalnya yang berjudul “Implementasi Data Mining Menggunakan Algoritma *Naive Bayes* Dalam Menentukan Pengunduran Diri Calon Mahasiswa Pada Universitas Dian Nuswantoro Semarang” membandingkan algoritma *Naive Bayes*, C4.5 dan *Nearest Neighbor* dengan mengambil data PMB pada tahun 2013 dengan tingkat akurasi sebesar 78% [5].
- e. Pada tahun 2014, Hera Wasiati dan Dwi Wijayanti dalam jurnalnya yang berjudul “Sistem Pendukung Keputusan Penentuan Kelayakan Calon Tenaga Kerja Indonesia Menggunakan Metode *Naive Bayes* (Studi Kasus : Di P.T. Karyatama Mitra Sejati Yogyakarta)” untuk memprediksi bahwa tenaga kerja ini layak diterima atau tidak membutuhkan kriteria sebagai berikut : pendidikan, usia, tinggi badan, berat badan, dan nilai tes. Dengan data yang digunakan sebanyak 542 dan menghasilkan 362 data training dan 180 sebagai data tes, yang menghasilkan pola sebesar 73,89% dan erornya 26,11% atau lebih tepatnya 133 diterima dan 47 tidak diterima [10].

- f. Pada tahun 2016, Diana Septiari di dalam jurnal yang berjudul “Implementasi Metode *Naive Bayes* Classification Dalam Klasifikasi Kelayakan Calon Pendoror Darah (Studi Kasus PMI Kab.Demak)” mencari calon pendonor darah dengan atribut nama, nomor id, jenis kelamin, usia, berat badan, kadar hemoglobin, tekanan darah atas, tekanan darah bawah, riwayat penyakit menular dan jangka waktu donor. Dengan pengujian menggunakan 60 data pendonor darah yang menghasilkan tingkat akurasi sebesar 81,6% [11].
- g. Pada tahun 2017, Irma Apriliani Dahlia menuliskan di dalam skripsi dengan judul “Perbandingan Metode *Naive Bayes* Dan K-Nearest Neighbor Untuk Prediksi Perceraian (Studi Kasus : Pengadilan Agama Cimahi)” membandingkan 2 algoritma *Naive Bayes* dan K-Nearest Neighbor memprediksi perceraian dengan 20 data testing dan 130 data training yang di mana *Naive Bayes* menghasilkan tingkat akurasi 72,5% sedangkan K-Nearest Neighbor menghasilkan tingkat akurasi perceraian sebanyak 57,5% [12].

1.2 Pengertian Pembayaran

Menurut pasal Undang – Undang No 23 Tahun 1999 Pasal 1 Ayat 6 dengan bunyi : “Pembayaran adalah suatu sistem yang mencakup seperangkat aturan, lembaga, dan mekanisme, yang digunakan untuk melaksanakan pemindahan dana guna memenuhi suatu kewajiban yang timbul dari suatu kegiatan ekonomi”. Pembayaran dapat melakukan mekanisme untuk memindahkan mata uang menjadi suatu barang, jasa atau informasi dari pembayar kepada penerima, baik secara langsung maupun melalui perbankan. Sebaliknya jika pembayar tidak melakukan pembayaran maka perpindahan mata uang menjadi satu barang, jasa atau informasi tidak akan mungkin terjadi.

1.3 Data Mining

Data Mining adalah data yang diproses menjadi informasi penting karena data tersebut sangat menarik, yang didalamnya menghasilkan informasi yang sangat bernilai [13]. Proses terjadinya *data mining* yaitu dengan menggunakan teknik statistik, matematika, kecerdasan buatan, dan *machine learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terhubung dari berbagai database yang jumlahnya sangat besar [14]. *Data Mining* dibagi menjadi beberapa kelompok berdasarkan tugasnya yaitu [15] :

a. Klasifikasi

Proses membedakan kelas data dalam beberapa kategori.

b. Deskripsi

Teknik yang dipergunakan dalam menggambarkan pola dan kecenderungan yang dimiliki oleh data

c. Estimasi

Estimasi hampir sama dengan klasifikasi, namun variabel target estimasi lebih tertuju ke arah numerik dari pada ke arah kategori.

d. Prediksi

Prediksi memiliki kemiripan dengan klasifikasi dan estimasi, namun nilai dari hasil prediksi akan ada di masa yang akan datang.

e. Klaster

Klaster adalah pengelompokan sejumlah data yang memiliki kemiripan ke dalam kelompok-kelompok data.

f. Asosiasi

Teknik yang digunakan untuk mencari hubungan antar karakteristik tertentu dalam satu waktu.

Data mining juga merupakan sebuah sekumpulan data yang jumlahnya sangat besar sehingga membutuhkan proses pemilihan data yang akan memberikan informasi yang akurat.

Tahapan Proses *Data Mining* [16] :

- a. *Data Selection* (Seleksi data), memilih data yang akan dijadikan proses data *mining* dari banyaknya data.
- b. *Data Cleaning* (Membersihkan data) merupakan langkah pembersihan data yang memiliki atribut, nilai, dan konsiten data, dengan data yang diharapkan.
- c. *Data Transformation* (Transformasi data), data yang ditransformasikan atau digabungkan untuk dijadikan data yang sesuai dengan proses data *mining*.
- d. *Data Mining*, proses mencari pola atau informasi dari data yang telah dipilih dengan menggunakan teknik atau metode tertentu.
- e. *Pattern Evalution* (Pola evaluasi), pola atau informasi data yang dihasilkan dari proses *data mining* yang mencakup, apakah pola atau informasi yang telah diproses serta sesuai dengan hasil fakta atau hipotesis.
- f. *Knowledge presentation* (Presentasi pengetahuan), tahapan akhir proses *data mining* yang akan ditampilkan dalam bentuk yang dapat dimengerti.

1.4 Metode Klasifikasi

Klasifikasi merupakan proses untuk mencari fungsi dan model yang dapat membedakan atau menjelaskan konsep atau kelas data dengan maksud

memperkirakan kelas yang tidak diketahui dari suatu objek [11]. Klasifikasi merupakan bagian dari *data mining*, dimana *data mining* merupakan salah satu yang digunakan dalam menguraikan penemuan pengetahuan didalam *database* [17]. Contoh dari klasifikasi adalah menganalisa seseorang akan membeli *handphone* atau tidak.

1.5 Naïve Bayes

Naïve Bayes merupakan sebuah teknik prediksi yang berbassis pada probabilitas sederhana yang memiliki dasar pada penerapan *Teorema Bayes* (atau dengan aturan *bayes*) dan memiliki asumsi independensi (ketidak ketergantungan) yang kuat (naif) [10]. *Teorema Bayes* mempunyai kemampuan berupa klasifikasi serupa dengan *decision tree* dan *neural network* [17]. *Naïve Bayes* telah terbukti memiliki akurasi dan kecepatan dalam mengaplikasikan ke dalam database dengan data yang besar [17]. Secara umum, *teorema bayes* dinyatakan persamaan 1 :

$$P(H|X) = \frac{P(X|H).P(H)}{P(X)} \quad (1)$$

Dimana :

- X adalah data dengan *class* belum diketahui
- H adalah hipotesis data X merupakan suatu *class* spesifik
- $P(H/X)$ adalah probabilitas hipotesis H berdasarkan kondisi X (posteriori probabilitas)
- $P(H)$ adalah probabilitas hipotesis H (probabilitas)
- $P(X/H)$ adalah probabilitas hipotesis X berdasarkan kondisi H
- $P(X)$ adalah probabilitas dari X

Keuntungan penggunaan *Naïve Bayes* adalah bahwa metode ini hanya memerlukan jumlah data latih (*data training*) yang kecil untuk memastikan estimasi parameter yang diperlukan dalam proses pengklasifikasian [4].

1.6 Confusion Matrix

Confusion matrix merupakan suatu metode yang digunakan untuk melakukan perhitungan akurasi pada konsep data mining [18] atau juga sebuah tabel yang menyatakan jumlah data uji yang benar diklasifikasikan dan jumlah data yang salah diklasifikasikan [19]. Recall adalah berapa persen data kategori positif yang terklasifikasikan dengan benar oleh system, Presisi yaitu nilai yang menggambarkan jumlah data kategori positif yang diklasifikasikan secara benar dibagi dengan total data yang diklasifikasi positif, dan Akurasi merupakan perbandingan kasus yang diidentifikasi benar dengan jumlah semua kasus [18].

Berikut rumus perhitungan recall, presisi, dan akurasi :

- a. Recall : $TP / (FN+TP) * 100\%$ (2)
- b. Presisi : $TP / (FP+TP) * 100\%$ (3)
- c. Akurasi : $(TN+TP) / (TP + TN + FP + FN) * 100\%$ (4)

Tabel 1.3 Tabel Confusion Matrix

Classification	Predicted Class	
	Class = Ya	Class = Tidak
Class = Ya	<i>True Positive (TP)</i>	<i>False Positive (FP)</i>
Class = Tidak	<i>False Negative (FN)</i>	<i>True Negative (TN)</i>

Dengan keterangan sebagai berikut :

- a. TP = Hasil yang terklasifikasi oleh naïve bayes YA dengan class asli YA
- b. FN = Hasil yang terklasifikasi oleh naïve bayes YA dengan class asli TIDAK
- c. FP = Hasil yang terklasifikasi oleh naïve bayes TIDAK dengan class asli YA
- d. TN = Hasil yang terklasifikasi oleh naïve bayes TIDAK dengan class asli TIDAK

1.7 *Unified Modeling Language (UML)*

Unified Modeling Language atau yang biasa disebut UML merupakan sebuah standar dalam membuat perangkat lunak yang terdiri dari visualisasi, merancang, dan dokumentasi sistem [20]. Penggunaan UML tidak memiliki batas pada metodologi tertentu, meskipun pada kenyataannya UML paling sering digunakan pada metodologi berorientasi objek [21]. UML memiliki standar penulisan yaitu berupa *blueprint*, yang dimana meliputi konsep proses bisnis, penulisan kelas-kelas dalam bahasa program yang lebih spesifik, memiliki skema *database*, dan beberapa komponen yang dibutuhkan oleh sistem.




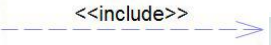
Berikut berupa alur diagram pada UML :

1.7.1 *Use Case Diagram*

Use Case Diagram merupakan suatu pemodelan untuk melakukan suatu interaksi antar aktor dengan sistem informasi yang akan dibuat, yang dimana interaksi usecase dapat dilakukan antar satu atau lebih aktor dengan sistem yang akan dibuat [21]. Syarat penamaan pada *use case* yaitu mendefinisikan nama yang dapat dipahami. Beberapa komponen utama yang ada di *use case diagram* yaitu :

- a. Aktor merupakan orang, proses atau sistem lain yang berhubungan dengan sistem secara langsung, tetapi aktor belum tentu orang.
- b. *Use Case* adalah fungsi dari suatu sistem yang dibangun, sehingga pengguna dapat memahami akan kegunaan sistem tersebut.


Tabel 1.4 Simbol *Use Diagram* [21].

Simbol	Nama	Deskripsi
 Actor	Actor	Actor adalah orang yang berinteraksi secara langsung dengan penggunaan sistem.
	Use Case	Use Case digambarkan berbentuk lingkaran <i>elips</i> , dimana fungsi <i>use case</i> disini sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.
	Association	Association digunakan untuk interaksi antar <i>use case</i> atau dengan aktor.
	Include	Include merupakan sebuah relasi tambahan antar <i>use case</i> dengan <i>use case</i> .

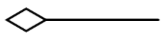


1.7.2 Class Diagram

Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem [21]. *Class Diagram* menjelaskan jenis-jenis objek dalam sistem dan berbagai hubungan statis yang terdapat diantara mereka. Berikut simbol-simbol yang biasa digunakan dalam membuat *class diagram*.

Tabel 1.5 Simbol *Class Diagram* [21]

Simbol	Nama	Deskripsi
<div style="border: 1px solid black; padding: 5px; width: fit-content;"> Nama Class + atribut + atribut + atribut + method + method </div>	Class	Kelas pada struktur sistem.
	Asosiasi / Association	Relasi antar kelas.






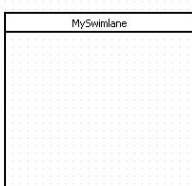
Tabel 1.6 Simbol *Class Diagram* [21] (Lanjutan)

Simbol	Nama	Deskripsi
	Agregasi / <i>Aggregation</i>	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>).
	Asosiasi berarah / <i>directed association</i>	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain.
	Generalisasi	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus)

1.7.3 Activity Diagram

Activity Diagram merupakan gambaran *workflow* (aliran kerja) atau alur dari suatu sistem atau proses bisnis atau menu yang ada pada perangkat lunak [21].






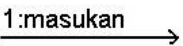
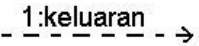
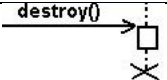
Tabel 1.7 Simbol *Class Diagram* [21]

Simbol	Nama	Deskripsi
	Status Awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah aktivitas awal.
	Aktivitas	Aktivitas yang dilakukan sistem.
	Percabangan/ <i>decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
 Penggabungan (Join)	Penggabungan / <i>join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
 End Point	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 yang terjadi

1.7.4 Sequence Diagram

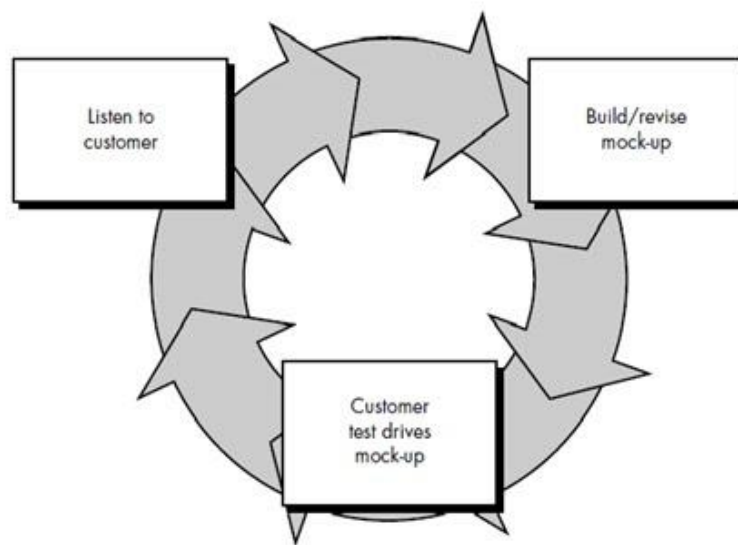
Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek [21] .

Tabel 1.8 Simbol *Sequence Diagram* [21]

Simbol	Nama	Deskripsi
	Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
	Garis hidup/ <i>lifeline</i>	Menyatakan kehidupan suatu objek
	Objek	Menyatakan objek yang berinteraksi pesan
	Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini.
	Pesan tipe <i>create</i>	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
	Pesan tipe <i>end</i>	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
	Pesan tipe <i>return</i>	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
	Pesan tipe <i>destroy</i>	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>

1.8 Pengertian *Prototype*

Prototype merupakan sistem informasi yang menggambarkan hal-hal penting dari sistem informasi yang akan datang [17]. Metode *Prototype* dibutuhkan dalam menyambungkan ketidak pahaman pelanggan mengenai hal teknik dan untuk memperjelas spesifikasi kebutuhan yang dibutuhkan pelanggan kepada pengembang perangkat lunak [21].



Gambar 1.1 Alur *Prototype*

Keunggulan *Prototype* :

- a. User dapat berpartisipasi aktif.
- b. Menentukan kebutuhan lebih mudah tercipta.
- c. Mempersingkat waktu pengembangan Sistem Informasi kelemahan.

Kekurangan *Prototype* :

- a. Proses analisis dan perancang terlalu singkat.
- b. Kurang fleksibel dalam menghadapi perubahan
- c. Prototype yang dihasilkan tidak selamanya mudah dirubah

- d. Prototype terlalu cepat selesai.

1.9 Bahasa Pemrograman

1.9.1 HTML

HyperText Markup Language atau HTML merupakan bahasa yang digunakan pada dokumen web sebagai bahan untuk pertukaran dokumen web [22] atau sebuah bahasa pemrograman yang dipakai untuk membuat suatu halaman web [23]. HTML memiliki beberapa element tersusun dari tag-tag yang memiliki fungsinya sendiri, seperti tag *heading*, *paragraf*, pembuatan form, tombol, *list*, membuat *hyperlink* atau *link* yang menghubungkan antar halaman website dan masih banyak element-element lainnya [23]. Suatu halaman website selalu diisi oleh kode tag `<html> </html>`. File HTML selalu menyimpan file dengan tipe *.htm atau *.html. Jadi jika mengetik sebuah kode program dan menyimpannya dengan ekstensi .html itu disebut file yang berformat HTML [24].

1.9.2 Cascading Style Sheets (CSS)

Cascading Style Sheets atau yang biasa disebut CSS merupakan bahasa formatting yang digunakan untuk men-design sebuah halaman website [23]. CSS sendiri dapat digunakan untuk mengubah ukuran *font*, warna dan format form, mengatur halaman website yang responsive dan masih banyak lagi yang dapat dilakukan CSS [23]. CSS memakai *selector* (*id* dan *class*) untuk menentukan element yang akan di modifikasi atau yang akan diberi sentuhan css. Jika diibaratkan HTML sebagai tiang pada sebuah bangunan rumah, maka CSS berfungsi sebagai dekorai pada bangunan rumah tersebut [23].

1.9.3 PHP

PHP merupakan interpreter yaitu proses penerjemahan baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan. PHP disebut sebagai pemrograman *Server Side Programing*, hal ini karena seluruh prosesnya dijalankan pada *server* [22]. Kode pada pemrograman PHP dapat ditulis dalam dua bentuk yaitu penulisan kode baris PHP file tunggal dan penulisan kode pada halaman html (*embedded*). Kedua cara tersebut tidak memiliki perbedaan, hanya saja menjadi kebiasaan gaya penulisan dari programmer [22].

Penulisan pada baris kode PHP terdiri dari dua bentuk yaitu penulisan baris kode menggunakan format PHP maupun penulisan baris kode dengan format campuran antara HTML dan PHP [22]. PHP sendiri diciptakan pada tahun 1994 oleh seorang programmer bernama Rasmus Lerdorf yang awalnya hanya membuat sebuah halaman website pribadi, tujuan Rasmus Lerdorf membuat halaman website pribadi adalah untuk mempertahankan halaman website pribadi tersebut sekaligus membangun halaman web yang dinamis [22].

1.9.4 Framework

1.9.4.1 Bootstrap

Bootstrap adalah *library* (pustaka/kumpulan fungsi-fungsi) dari sebuah framework CSS yang diciptakan untuk sebuah pengembangan *frontend* website. Didalam bootstrap sendiri terdapat *library* dengan bermacam-macam jenis HTML, CSS, dan java script [25]. *Bootstrap* telah memberikan kumpulan aturan dan komponen sebuah *class interface* dasar sebagai modal untuk pembuatan web yang

telah dirancang sangat baik, menarik, bersih, ringan dan mempermudah penggunaanya [25].

1.9.4.2 Code Igniter

Code Igniter (CI) adalah sebuah *framework* pengembangan aplikasi (*Application Development Framework*) dengan menggunakan bahasa pemrograman PHP, yang dimana kerangka untuk bekerja menggunakan PHP yang lebih sistematis. Pemrogram tidak perlu repot lagi membuat program dari awal, karena CI disini menyediakan sekumpulan *library* yang banyak yang diperlukan untuk menyelesaikan pekerjaan yang umum, dengan menggunakan antarmuka dan struktur logika yang sederhana untuk mengakses librarinya [26]. *Framework* CI merupakan framework yang mempunyai dokumentasi yang jelas dan lengkap, yang memudahkan pengembangan agar mempelajari dengan mudah. Pendekatan dari CI sangatlah mudah, dari sekedar membuat tulisan hingga dengan yang kompleks dapat didekati dengan mudah[26].

1.9.4.3 Basis Data MySQL

MySQL merupakan suatu RDBMS (*Relational Database Management Sytem*) yaitu suatu aplikasi yang menjalankan fungsi pengolahan pada data. *MySQL* diciptakan pertama kali oleh *MySQL AB* yang kemudian diakuisisi oleh Sun Microsystems dan diakusisi kembali oleh Oracle Coorportaion [22]. Singkatnya *MySQL* mempermudah pekerjaan pada *SQL* (*Structured Query Language*), yang dimana *SQL* merupakan sebuah sintak perintah-perintah tertentu atau bahasa (pemrograman) yang digunakan untuk mengelola suatu database.

1.10 Teknik Pengujian

Pengujian merupakan salah satu set kegiatan yang direncanakan dan sistematis untuk menguji atau mengevaluasi kebenaran yang diinginkan. Aktivitas pengujian terdiri dari satu set atau sekumpulan langkah dimana dapat menempatkan desain kasus uji yang spesifik dan metode pengujian [21]. Berikut pola pengujian pada perangkat lunak sebagai berikut :

- a. Pengujian diawali dari level komponen hingga integrasi antar komponen menjadi sebuah sistem.
- b. Teknik pengujian berbeda-beda sesuai dengan berbagai sisi atau unit uji dalam waktu yang berbeda-beda pula bergantung pada pengujian pada bagian mana yang dibutuhkan.
- c. Pengujian dilakukan oleh pengembang perangkat lunak, dan jika untuk proyek besar, pengujian bias dilakukan oleh tim uji yang tidak terkait dengan tim pengembang perangkat lunak.
- d. Pengujian dilakukan untuk mencari kesalahan (*error*) baik dari sudut pandang orang secara umum maupun dari sudut pandang pengembang tanpa harus menemukan lokasi kesalahan pada kode program.

a. Pengujian *Black-Box* (Kotak Hitam)

Black-Box adalah pengujian perangkat lunak dengan melihat spesifikasi fungsional tanpa melakukan pengujian terhadap desain dan kode program. Pengujian ini dimaksud untuk mencari tahu apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan [21]. Pengujian *Black-Box* dilakukan untuk menguji apakah perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Untuk melakukan pengujian *Black-Box* ini

harus dibuat dengan kasus benar dan kasus salah, contoh pengujian *black-box* tentang proses login [21]:

- a. Jika user memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang benar.
- b. Jika user memasukan nama pemakai (*username*) dan kata sandi salah (*password*) yang salah, misalnya nama pemakai benar tapi kata sandi salah, atau sebaliknya, atau keduanya salah.

