

BAB II

LANDASAN TEORI

2.1 Beasiswa

Beasiswa merupakan program kerja yang ada di setiap sekolah. Program beasiswa bertujuan untuk meringankan beban biaya siswa dalam menempuh masa studi sekolah. Pemberian beasiswa kepada siswa dilakukan secara selektif, SMKN 2 adalah salah satu sekolah menengah kejuruan yang menyediakan program beasiswa untuk meringankan beban biaya pendidikan apabila siswa memiliki prestasi karena beasiswa yang ada yaitu beasiswa prestasi. (Fiqih Satria, RZ. Abdul Aziz : 2016)

2.2 Sistem Pendukung Keputusan (SPK)

Sistem Pendukung Keputusan adalah sistem penghasil informasi spesifik yang ditujukan untuk memecahkan suatu masalah tertentu yang harus dipecahkan oleh manager pada berbagai tingkatan. Dengan kata lain, Sistem Pendukung Keputusan adalah suatu sistem informasi berbasis komputer yang menghasilkan berbagai alternatif keputusan untuk membantu manajemen dalam menangani berbagai permasalahan yang terstruktur dengan menggunakan data dan model. (Sukerti : 2015)

Karakteristik Sistem Pendukung Keputusan (Lulu, Sari, & Rachmawati : 2011).

- a) Mendukung pengambilan keputusan untuk membahas masalah-masalah terstruktur, semi struktur dan tidak terstruktur.
- b) Hasil ditujukan bagi personil organisasi dalam semua tingkatan.
- c) Mendukung di semua fase proses pengambilan keputusan yaitu intelegensi, desain, pilihan.
- d) Adanya *interface* manusia atau mesin, dimana manusia (*user*) tetap mengontrol proses pengambilan keputusan.
- e) Menggunakan model-model matematis dan statistik yang sesuai dengan pembahasan.

Menurut Simon, dalam pengambilan keputusan terdapat 3 tahap utama, yaitu Tahap *Intellegence*, Tahap *Design*, Tahap *Choice*, kemudian Simon menambahkan tahap keempat, yaitu Tahap *Implementation*. (Nofriansyah : 2013)

a) Tahap *Intellegence*

Tahap *Intellegence* adalah tahap pengenalan persoalan melalui penyelidikan lingkungan untuk mengetahui ada atau tidaknya permasalahan. Kesimpulan dari penelitian diperoleh dari pengolahan data dengan metode khusus.

b) Tahap *Design*

Pada tahap ini dilakukan pencarian, analisis serta perumusan alternatif tindakan yang akan diambil. Pada tahap ini pula, sistem informasi harus mampu membuat keputusan-keputusan.

c) Tahap *Choice*

Pada tahap ini seorang pengambil keputusan melakukan proses pemilihan diantara berbagai alternatif tindakan yang mungkin dijalankan. Hasil pemilihan tersebut kemudian diimplementasikan dalam proses pengambilan keputusan.

d) Tahap *Implementation*

Pada hakikatnya implementasi suatu solusi yang diusulkan untuk suatu masalah adalah inisiasi terhadap hal baru atau pengenalan terhadap perubahan. Dan perubahan harus dikelola. Harapan-harapan pengguna harus dikelola sebagai bagian dari manajemen perubahan.

2.3 *Naïve Bayes*

Naïve Bayes merupakan metode probabilistik pengklasifikasian sederhana berdasarkan *Teorema Bayes* dimana pengklasifikasian dilakukan melalui *training* 186 data secara efisien. *Naïve Bayes* mengasumsikan bahwa nilai dari sebuah *input* atribut pada kelas yang diberikan tidak tergantung dengan nilai atribut yang lain. *Teorema Bayes* sendiri dikemukakan oleh ilmuwan Inggris *Thomas Bayes*, yaitu memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai *Teorema Bayes*. (Diasrina Dahri, Fahrul Agus, Dyna Marisa Khairina: 2016).

Teorema Bayes memiliki bentuk umum sebagai berikut.

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

Dengan :

X = Data dengan kelas yang belum diketahui

H = Hipotesis data X merupakan suatu label kelas tertentu

$P(H|X)$ = Probabilistik hipotesis H berdasarkan kondisi X (*posteriori probability*)

$P(H)$ = Probabilistik hipotesis H (*prior probability*)

$P(X|H)$ = Probabilistik X berdasarkan kondisi pada hipotesis H

$P(X)$ = Probabilistik X

Metode *Naïve Bayes* memiliki kelebihan sebagai berikut.

- a) Metode *Naïve Bayes* merupakan salah satu metode yang *simple* dan mudah dibuat.
- b) Metode *Naïve Bayes* mampu untuk mengukur kinerja relatif dalam perhitungan matematika sederhana.

Metode *Naïve Bayes* memiliki kekurangan sebagai berikut.

- a) Tidak berlaku jika probabilitas kondisionalnya adalah nol, apabila nol maka probabilitas prediksi akan bernilai nol juga.
- b) Satu probabilitas saja tidak bisa mengukur tingkat ke akuratan.

Tahapan proses metode *Naïve Bayes* sebagai berikut.

- a) Menghitung jumlah kelas / label.
- b) Menghitung jumlah kasus per kelas.
- c) Kalikan semua variable kelas.
- d) Bandingkan hasil per kelas.

2.4 Basis Data (*Database*)

Basis data adalah kumpulan data yang saling berhubungan secara logis dan didesain untuk mendapatkan data yang dibutuhkan oleh suatu organisasi. (Indrajani, 2015)

Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah dan membuat informasi tersedia saat dibutuhkan. (Rosa dan Shalahuddin: 2015)

2.5 Bahasa Pemrograman dan Perangkat Lunak Pendukung

Bahasa pemrograman dan perangkat lunak pendukung yang digunakan dalam penulisan ini adalah *Website*, *XAMPP*, *PHP*, *HTML*, *CSS*, *Java Script* dan *MySQL*.

2.5.1 *Website*

Website atau disingkat *web*, dapat diartikan sekumpulan halaman yang terdiri dari beberapa halaman yang berisi informasi dalam bentuk data digital baik berupa *text*, gambar, video, audio dan animasi lainnya yang disediakan melalui jalur koneksi internet. (Abdulloh: 2015)

2.5.2 *XAMPP*

Xampp adalah sebuah paket kumpulan *software* yang terdiri dari *Apache*, *MySQL*, *PhpMyAdmin*, *PHP*, *Perl*, *Filezilla* dan lainnya. Fungsi dari *Xampp* adalah untuk memudahkan instalasi lingkungan *PHP*, di mana biasanya lingkungan pengembangan *web* memerlukan *PHP*, *Apache*, *MySQL* dan *PhpMyAdmin*. (MADCOMS : 2016)

Xampp dikembangkan oleh perusahaan *apache friends* yang memiliki kelebihan bisa berperan sebagai *server web apache* untuk simulasi pengembangan *website*. *Tool* pengembangan *web* ini mendukung teknologi *web* populer seperti *PHP*, *MySQL* dan *Perl*.

2.5.3 PHP

PHP singkatan dari *Hypertext Preprocessor* yang merupakan *server-side programming*, yaitu bahasa pemrograman yang diproses di sisi *server*. Fungsi utama *PHP* dalam membangun *website* adalah untuk melakukan pengolahan data pada *database*. Data *website* akan dimasukkan ke *database*, diedit, dihapus dan ditampilkan pada *website* yang diatur oleh *PHP*. (Rohi Abdullah : 2015)

2.5.4 HTML

HTML singkatan dari *Hypertext Markup Language*, yaitu *script* yang berupa tag-tag untuk membuat dan mengatur struktur *website*. (Rohi Abdullah : 2015)

Beberapa tugas *HTML* dalam membangun *website* diantaranya sebagai berikut.

- a) Menentukan *layout website*.
- b) Memformat *text* dasar seperti pengaturan *paragraph*, dan format *font*.
- c) Membuat *list* dan formulir.
- d) Membuat tabel, gambar, video, audio dan *link*.

2.5.5 CSS

Cascading Style Sheets yaitu untuk mengatur desain *website*. Walaupun *HTML* mempunyai kemampuan untuk mengatur tampilan *website*, namun kemampuannya sangat terbatas. Fungsi *CSS* adalah memberikan pengaturan yang lebih lengkap agar struktur *website* yang dibuat dengan *HTML* terlihat lebih indah. (Rohi Abdullah : 2015)

2.5.6 Java Script

JavaScript adalah suatu bahasa *scripting* yang digunakan sebagai fungsionalitas dalam membuat suatu *web*. (Hidayatullah dan Kawistara : 2015)

2.5.7 MySQL

MySQL (*My Structure Query Language*) merupakan suatu *database*. *MySQL* dapat juga dikatakan sebagai *database* yang sangat cocok bila dipadukan dengan

PHP. Secara umum *database* berfungsi sebagai tempat atau wadah untuk menyimpan, mengklasifikasikan data secara profesional. (Core & Fedora : 2014)

2.6 RUP (*Rational Unified Process*)

RUP (Rational Unified Process) merupakan suatu metode rekayasa perangkat lunak yang dikembangkan dengan mengumpulkan berbagai *best practises* yang terdapat dalam industri pengembangan perangkat lunak. Ciri utama metode ini adalah menggunakan *usecase driven* dan pendekatan iteratif untuk siklus pengembangan perangkat lunak. (Sutedi dan Melda Agarina : 2017)

2.7 UML (*Unified Modeling Language*)

UML (Unified Model Language) merupakan alat industri standar yang memungkinkan kita untuk mengkomunikasikan dengan jelas tentang kebutuhan, arsitektur dan desain. *UML* merupakan salah satu alat yang paling berguna dan paling populer di dalam dunia pengembangan sistem.

UML adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. (Dede Wira Trise Putra, Rahmi Andriani : 2019)

Tabel 2.1 Tipe Diagram UML (*Unified Modelling Language*)

No.	Diagram	Tujuan
1	<i>Class</i>	Memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi.
2	<i>Package</i>	Memperlihatkan kumpulan kelas-kelas, merupakan dari <i>diagram</i> komponen.
3	<i>Use Case</i>	<i>Diagram</i> ini memperlihatkan himpunan <i>use case</i> dan aktor-aktor (suatu jenis khusus dari kelas).
4	<i>Sequence</i>	<i>Diagram</i> interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.

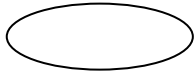
5	<i>Communication</i>	Sebagai pengganti diagram kolaborasi yang menekankan organisasi struktural dari objek-objek yang menerima serta mengirim pesan.
6	<i>Statechart</i>	<i>Diagram</i> status memperlihatkan keadaan-keadaan pada sistem, memuat status (<i>state</i>), transisi, kejadian serta aktivitas.
7	<i>Activity</i>	Tipe khusus dari <i>diagram</i> status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem.
8	<i>Component</i>	Memperlihatkan organisasi serta kebergantungan sistem atau perangkat lunak pada komponen komponen yang telah ada sebelumnya.

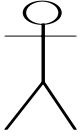

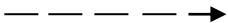
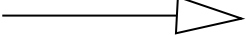
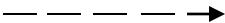
2.7.1 Jenis-Jenis *Diagram UML (Unified Modelling Language)*

2.7.1.1 *Use Case Diagram*

Use case diagram merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat. *Use case* bekerja dengan mendeskripsikan tipikal interaksi antara *user* sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sistem itu dipakai. (Dede Wira Trise Putra, Rahmi Andriani : 2019)

Table 2.2 Simbol *Use Case Diagram*


No.	Simbol	Deskripsi
1	<p><i>Use Case</i></p> 	Fungsional yang disediakan sistem sebagai unit-unit yang paling saling bertukar pesan antara unit atau <i>actor</i> , biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .

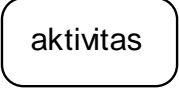
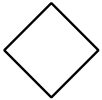

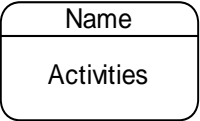

2	Aktor / <i>actor</i> 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan itu sendiri, sehingga walaupun simbol dari aktor adalah gambar orang biasanya dinyatakan menggunakan kata benda.
3	Asosiasi / <i>association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan actor.
4	<i>extend</i> > 	<i>Case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan.
5	<i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi satu adalah fungsi yang lebih umum dari lainnya.
6	< <i>include</i> > 	Fungsinya sebagai syarat dijalankannya <i>use case</i> .

2.7.1.2 Activity Diagram

Activity diagram merupakan *diagram* yang menggambarkan *workflow* atau aktivitas dari sebuah sistem yang ada pada perangkat lunak. (Dede Wira Trise Putra, Rahmi Andriani : 2019)

Table 2.3 Simbol *Activity Diagram*






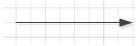
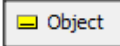



No.	Simbol	Deskripsi
1	Status awal 	Status awal yang dilakukan sistem, sebuah <i>diagram</i> aktivitas memiliki sebuah status awal.

2	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3	Percabangan 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4	Penggabungan 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5	<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.
6	Status akhir 	Status akhir yang dilakukan sistem, sebuah <i>diagram</i> aktivitas memiliki sebuah status akhir.

2.7.1.3 Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Semakin banyak *use case* yang didefinisikan, maka *sequence diagram* yang harus dibuat juga semakin banyak. (Dede Wira Trise Putra, Rahmi Andriani : 2019).

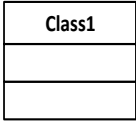
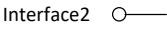


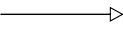
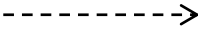

Table 2.4 Simbol *Sequence Diagram*

No.	Simbol	Deskripsi
1	<i>Actor</i> 	Menggambarkan orang yang berinteraksi dengan sistem.
2	<i>Entity class</i> 	Menggambarkan hubungan kegiatan yang akan dilakukan.
3	<i>Boundary class</i> 	Menggambarkan sebuah penggambaran dari <i>form</i> .
4	<i>Control class</i> 	Menggambarkan penghubung antara <i>boundary</i> dengan tabel.
5	<i>Lifeline</i> 	Menggambarkan tempat mulai dan berakhirnya sebuah pesan.
6	<i>Line message</i> 	Menggambarkan pengiriman pesan.
7	<i>Object</i> 	Menyatakan objek yang berinteraksi pesan.
8	Garis hidup 	Menyatakan kehidupan suatu objek.
9	Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi.
10	Stimulus 	Menyatakan suatu objek mengirimkan pesan untuk menjalankan operasi yang ada pada objek lain.

2.7.1.4 Class Diagram

Class diagram merupakan gambaran struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. *Class diagram* terdiri dari atribut dan operasi dengan tujuan pembuat program dapat membuat hubungan antara dokumentasi perancangan dan perangkat lunak. (Dede Wira Trise Putra, Rahmi Andriani : 2019)

Tabel 2.5 Simbol *Class Diagram*

No.	Simbol	Deskripsi
1	Kelas 	Kelas pada struktur sistem.
2	Antarmuka/ <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3	Asosiasi 	Relasi antar kelas dalam makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4	Asosiasi berarah 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
6	Kebergantungan 	Relasi antar kelas dengan makna kebergantungan antar kelas.
7	Agregasi 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>).