

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Pada penelitian (Rachmawati et al., 2019), dilakukan penelitian pembangunan Helpdesk Ticketing System berbasis web. Dengan adanya Helpdesk Ticketing System berbasis web ini dapat mempermudah dalam penyelesaian masalah dan mengimbangi kebutuhan akan penyajian informasi yang cepat, tepat, akurat dan lebih efektif. Pembangunan Helpdesk Ticketing System berbasis web ini akan memberi dampak yang lebih baik dan efisien dibandingkan dengan monitoring secara manual. Pengujian dilakukan menggunakan metode waterfall dalam pengembangannya.

Dalam penelitian yang dilakukan oleh (Alfian et al. 2020), dilakukan perancangan sistem informasi ticketing helpdesk pada DPMPTS (Dinas Penanggulangan Kebakaran dan Penyelamatan) Pemprov DKI Jakarta. Aplikasi dalam penelitian ini bertujuan untuk memastikan proses persetujuan dan pengerjaan berjalan tanpa kesalahan komunikasi di DPMPTS Pemprov DKI Jakarta. Dengan adanya sistem informasi Ticketing Helpdesk ini, diharapkan dapat membantu meningkatkan efisiensi kegiatan operasional, sehingga proses bisnis di bidang PSTIK (Pencegahan dan Penanggulangan Kebakaran) dapat berjalan lebih baik dibandingkan dengan sistem sebelumnya. Pengembangan aplikasi ini menggunakan metode waterfall dalam proses pengujian dan implementasinya.

Pada penelitian yang dilakukan oleh (Adam et al. 2020), dilakukan pengembangan sistem IT Helpdesk berbasis web di Universitas Klabat.

Departemen yang bertugas melaporkan gangguan dan kerusakan sering menghadapi kendala dalam mengetahui perkiraan waktu penanganan masalah karena proses antrian pekerjaan tidak terlihat jelas, sehingga terkesan kurang efisien. Tujuan dari penelitian ini adalah untuk mengembangkan sistem IT Helpdesk yang mampu mengatasi pelaporan gangguan dan memantau pekerjaan perbaikan secara lebih efektif. Metode yang digunakan dalam penelitian ini adalah Sistem Development Lifecycle. Hasil dari penelitian ini adalah sebuah sistem yang dapat digunakan oleh departemen IT untuk membantu memonitor tugas dan pekerjaan yang masuk dengan lebih baik.

2.2 Teori Terkait

2.2.1. Sistem Informasi

Menurut Galuh dalam (Putu Wuri Handayani, 2020), Sistem informasi merupakan kumpulan komponen yang terkait dan memiliki tujuan untuk mengumpulkan, memproses, menyimpan, dan mendistribusikan informasi dalam suatu organisasi. Tujuan utamanya adalah untuk mendukung proses pengambilan keputusan dan pengawasan di lingkungan organisasi tersebut.

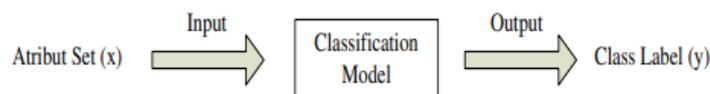
Pada penelitian (Effendi, E, 2023) , Komponen-komponen dalam sistem informasi meliputi perangkat keras (hardware), perangkat lunak (software), basis data (database), jaringan (network), serta sumber daya manusia yang terlibat dalam penggunaan, pengelolaan, dan pengembangan sistem tersebut.

Dengan adanya sistem informasi yang baik dan efisien, organisasi dapat mengoptimalkan informasi yang dimilikinya tentang orang, tempat, dan hal lain

yang relevan dalam lingkungan organisasi atau luarannya. Informasi yang tersedia tersebut dapat menjadi landasan bagi pengambilan keputusan yang lebih baik dan strategis.

2.2.2. Metode *Naive Bayes*

Menurut (Heliyanti, 2022), Metode Naive Bayes adalah metode yang menggunakan proses Klasifikasi. Klasifikasi sendiri adalah proses untuk menemukan model atau fungsi yang menjelaskan atau membedakan konsep atau kelas data. Tujuan yang dikandung yaitu dapat memperkirakan kelas dari suatu objek yang labelnya tidak diketahui. Dalam mencapai tujuan tersebut, proses klasifikasi membentuk suatu model yang dapat membedakan kelas data berdasarkan aturan atau fungsi tertentu. Model akan berisikan aturan seperti “jika – maka” , dapat berupa pohon keputusan atau formula matematis.



Gambar 2. 1 Blok Diagram Model Klasifikasi

Algoritma Naive Bayes merupakan salah satu algoritma yang menggunakan teknik klasifikasi dengan metode probabilitas dan statistik yang dikemukakan oleh ilmuwan Inggris Thomas Bayes. Pada klasifikasi ini yaitu memprediksi peluang di masa depan berdasarkan pengalaman dimasa sebelumnya sehingga saat ini dikenal dengan Teorema Bayes. Klasifikasi Naive Bayes diasumsikan bahwa ada atau tidak ciri tertentu sebuah kelas, tidak ada hubungannya dengan ciri dari kelas lainnya.

$$P(ai | vj) = \frac{nc + mp}{n+m}$$

Keterangan:

$P(a_i | v_j)$ = Peluang atribut-atribut (inputan) jika diketahui keadaan v_j

$a_i (a_1 a_2 \dots a_n)$ = atribut (inputan)

$v_j (v_1 v_2 \dots v_m)$ = teknisi ke- j

n_c = jumlah record pada data jika $v = v_j$ dan $a = a_i$

m_p = 1/ banyaknya teknisi

m = jumlah parameter gejala

n = jumlah record pada data jika $v = v_j$ tiap class

Untuk menyelesaikan persamaan tersebut, dilakukan tahapantahapan sebagai berikut:

- a. Menentukan nilai n_c untuk setiap class
- b. Menghitung nilai $P(a_i | v_j)$ dan menghitung nilai $P(v_j)$
- c. Menghitung $P(a_i | v_j) \times P(v_j)$ untuk tiap v
- d. Menentukan hasil klasifikasi yaitu v yang memiliki hasil perkalian yang terbesar.

2.2.3. Agile Development

Pada (Risah, 2020), *Agile Development* adalah “tentang masukan dan perubahan”, dan menekankan bahwa pengembangan perangkat lunak adalah proses yang tidak secara kaku terpaku pada tahapan yang direncanakan tetapi juga mempertimbangkan kemungkinan perubahan berdasarkan pengalaman yang dihadapi, dimana masukan yang didapat disetiap prosesnya adalah bagian yang penting untuk mencapai tujuan yang diharapkan dan hasil yang sesuai dengan prediksi.

2.2.4. *Scrum*

1. Pengertian *Scrum*

Menurut Shon Hardji (2021), *Scrum* adalah sebuah kerangka kerja dimana orang-orang dapat mengatasi masalah kompleks adaptif, dimana pada saat bersamaan mereka juga memberikan produk dengan nilai tinggi secara produktif dan kreatif.

2. Tim *Scrum*

Sri Mulyono (2018), Dalam Tim *Scrum* terdapat tiga peran yaitu *Product Owner*, Tim Pengembang dan *Scrum Master*. Tim *Scrum* harus bersifat fleksibel terhadap perubahan yang sewaktu-waktu terjadi, menentukan cara sendiri untuk menyelesaikan pekerjaannya dan bersifat produktif sehingga pekerjaan yang dikerjakan juga memberikan hasil. Tim *Scrum* mengerjakan pekerjaan secara berkala dan bertahap disertai dengan dokumentasi pekerjaan yang telah dikerjakan dan diselesaikan.

a. *Product Owner*

Product Owner adalah yang bertanggung jawab untuk memaksimalkan bisnis dari produk yang dihasilkan oleh *Development Team*. *Product Owner* berperan untuk memastikan product yang dikerjakan dan dikembangkan menghasilkan nilai tambah untuk pelanggan dan perusahaan.

Seluruh anggota Tim *Scrum* harus menghormati setiap keputusan yang dibuat oleh *Product Owner*. Namun sebagai *Product Owner* juga tetap menghargai terhadap saran dan masukan yang diberikan oleh anggota Tim *Scrum*.

b. Tim Pengembang (*Development Team*)

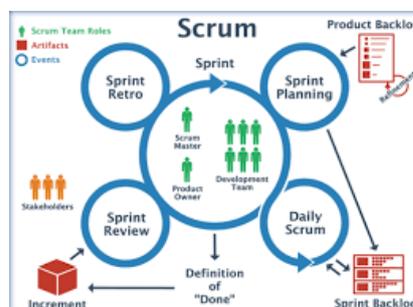
Tim Pengembang yaitu yang bekerja untuk menghasilkan tambahan potongan produk “Selesai”, yang berpotensi untuk dirilis di setiap akhir Sprint. Tim Pengembang dibentuk dan didukung oleh organisasi untuk mengatur dan mengelola pekerjaannya secara mandiri. Kekompakan tim pengembang akan meningkatkan efisiensi dan efektifitas dari Tim Pengembang secara keseluruhan.

c. *Scrum Master*

Scrum Master adalah seseorang yang memimpin dan melayani *Scrum Team*. *Scrum Master* yang nantinya membantu orang-orang diluar *Scrum Team* untuk dapat memahami interaksi mana yang bermanfaat dan yang tidak. *Scrum Master* yang membantu memaksimalkan nilai bisnis yang dihasilkan *Scrum Team*.

3. Acara-acara *Scrum*

Acara wajib yang diselenggarakan *Scrum* yang berguna untuk terjadinya kesinambungan dan mengurangi adanya acara lain yang tidak tercantum dalam *Scrum*. Setiap acara *Scrum* memiliki durasi waktu. Pada saat *Sprint* dimulai durasi waktu tidak dapat di perpendek maupun diperpanjang. Acara *Scrum* dapat diakhiri pada saat tujuan dari acara tersebut telah tercapai.



Gambar 2. 2 Acara-acara Scrum

a. *Sprint*

Sprint merupakan Jantung dari *Scrum*. Yang dimana pengertian dari *Sprint* sendiri yaitu sebuah batasan waktu yang konsisten selama satu bulan kalender atau kurang dan sepanjang proses pengembangan produk yang digunakan untuk menyelesaikan sesuatu. *Sprint* baru akan dimulai kembali pada saat *Sprint* sebelumnya berakhir. Apabila jangka waktu *Sprint* terlalu panjang, maka apa yang dibangun dapat berubah, kompleksitas dapat meningkat dan resiko yang ada juga bertambah. Setiap *Sprint* memiliki tujuan yang akan dikembangkan, perancangan yang fleksibel akan membimbing pengembangan, pekerjaan yang akan dilakukan dan hasil dari produk.

b. *Sprint Planning*

Sprint Planning adalah merencanakan pekerjaan yang akan dikerjakan di *Sprint*. Perencanaan ini dilakukan secara kolaboratif oleh seluruh anggota *Scrum Team*. *Sprint Planning* memiliki batasan waktu maksimal yaitu 8 (delapan) jam dengan jangka waktu 1 (satu) bulan.

c. *Daily Scrum*

Daily Scrum adalah acara untuk *Development Team* yang memiliki batasan waktu 15 menit. Acara *Daily Scrum* dilakukan setiap hari selama *Sprint* berlangsung. Pada acara ini, *Development Team* membuat rencana kerja untuk 24 jam ke depan. Acara ini juga mengedepankan performa *Development Team* dalam melakukan inspeksi pekerjaan sebelumnya dan perencanaan terhadap pekerjaan selanjutnya di dalam *Sprint*. *Daily Scrum* biasanya dilakukan pada tempat dan waktu yang sama.

d. *Sprint Review*

Sprint Review diselenggarakan di akhir *Sprint* untuk menginspeksi *Increment* dan mengadaptasi *Product Backlog*. Hasil dari *Sprint Review* adalah *Product Backlog* yang sudah dikerjakan maupun di revisi. Dan juga menampilkan *Product Backlog* terbaru yang mungkin dikerjakan di *Sprint* berikutnya.

e. *Sprint Retrospektive*

Sprint Retrospektive adalah sebuah kesempatan bagi *Scrum Team* untuk menginspeksi dirinya sendiri dan membuat perencanaan mengenai peningkatan yang akan dilakukan pada *Sprint* selanjutnya. *Sprint Retrospektive* terselenggara setelah *Sprint Review* dan sebelum *Sprint Planning*. Adanya *Sprint Retrospektive*, menjadikan Mengimplementasikan peningkatan di *Sprint* berikutnya adalah bentuk peningkatan dan adaptasi terhadap *Scrum Team* itu sendiri. Meskipun peningkatan dapat dilakukan sepanjang *Sprint*, *Sprint Retrospektive* adalah kesempatan yang fokus pada inspeksi dan adaptasi.

4. Artefak-artefak *Scrum*

Artefak Scrum mempresentasikan pekerjaan atau nilai bisnis guna terciptanya transparansi dan kesempatan untuk menginspeksi dan mengadaptasi.

a. *Product Backlog*

Product Backlog pada intinya yaitu adalah suatu daftar pekerjaan yang harus dikerjakan. Daftar tersebut diurutkan dari atas sampai bawah menurut kebutuhannya dan harus dilakukan dalam mengerjakan suatu produk. Daftar *Product Backlog* berisikan seluruh fitur, fungsi, kebutuhan, peningkatan dan perbaikan yang perlu dilakukan terhadap produk yang akan dibuat. *Product*

Backlog bersifat dinamis dan dapat berubah terus menerus agar produk menjadi layak, kompetitif dan bermanfaat. *Product Backlog* akan menjadi bukti pengerjaan produk hingga produk tersebut dikatakan “Selesai”.

b. *Sprint Backlog*

Sprint Backlog menampilkan seluruh pekerjaan yang menurut *Development Team* perlu dikerjakan untuk mencapai *Sprint Goal*. Untuk memastikan adanya peningkatan, *Sprint Backlog* setidaknya terdapat satu proses peningkatan dengan prioritas tertinggi dari hasil pertemuan *Sprint Retrospective* sebelumnya.

c. *Increment*

Increment adalah hasil dari *Product Backlog* yang diselesaikan dalam *Sprint* dan total nilai bisnis *Increment* yang baru harus “Selesai”, yang artinya *Increment* tersebut harus berada pada kondisi yang melalui peningkatan.

2.2.5. Konsep Dasar *Helpdesk*

Menurut Rachmawati (2019), Beberapa pengertian mengenai *helpdesk* yang terdapat dalam beberapa literatur:

- a. *Helpdesk* adalah nama yang umum digunakan untuk pusat bantuan terhadap end-user. *Helpdesk* sekarang ini sebagai fungsi layanan dan bertanggung jawab untuk memberikan bantuan atau layanan kepada pengguna untuk menyelesaikan masalah dan memecahkan masalah yang dialami pengguna.
- b. *Helpdesk* merupakan tempat dimana masalah dilaporkan dan akan mendapatkan solusi dari setiap masalah yang diajukan. Pengguna yang mengajukan bias berasal dari internal maupun eksternal, sehingga menjadikan

fungsi ini sangat krusial dalam menjaga kelancaran operasional perusahaan serta memberikan bantuan yang dibutuhkan oleh pengguna.

- c. *Helpdesk* adalah lapisan pertama yang dihubungi oleh end-user bila mereka mendapati masalah. *Helpdesk* yang akan berusaha untuk menangani, memecahkan masalah. Tetapi jika ada kegagalan dalam menangani masalah, tim *helpdesk* akan mengirimkan kepada tim senior. *End-user* selalu menghubungi *helpdesk* saat melaporkan masalah atau menanyakan progress masalah yang sudah diajukan. *End-user* dilarang langsung menghubungi tim *helpdesk* yang inti/lebih dalam.
- d. *Helpdesk* adalah titik utama pada saat pengguna menanyakan atau menemukan masalah yang berhubungan dengan mencapai tujuan yang diinginkan. *Helpdesk* memiliki basis data yang berisi catatan tentang masalah-masalah yang muncul selama operasional sehari-hari dan solusi yang diberikan untuk mengatasi masalah tersebut. *Helpdesk* akan merespon *complaint*, dan membuat prioritas pengerjaan masalah. Staf *helpdesk* juga harus mempunyai pengetahuan yang luas. Hal ini diperlukan agar sebuah masalah dapat segera dikerjakan dengan solusi yang benar. Jika PIC Asuransi tidak langsung menghubungi *helpdesk*, maka penanganan masalah menjadi tidak terkoordinasi dan pengetahuan hilang setelah solusi dilakukan.

Helpdesk memiliki beberapa kelebihan, antara lain:

- a. *Helpdesk* dapat memberikan solusi secara cepat terhadap pertanyaan, laporan dan keluhan yang masuk.

- b. *Helpdesk* mampu melakukan pengecekan terhadap status permasalahan yang dilaporkan dan mengatur penugasan tugas untuk staff yang terlibat.
- c. Melalui *helpdesk*, efisiensi kerja perusahaan dapat ditingkatkan dalam menyelesaikan berbagai permasalahan yang timbul.

Cara Kerja *Helpdesk*

- a. Menampung, memberikan prioritas terhadap *request ticket* melalui berbagai jalur seperti *E-Mail, Website, atau Telephone*.
- b. Melakukan *assignment request ticket* kepada unit-unit yang bertanggung jawab dan memiliki kemampuan untuk menyelesaikan permasalahan tersebut.
- c. *Helpdesk* melakukan pemantauan terhadap durasi dan status penyelesaian, serta mencatat seluruh tahapan tindakan yang telah diambil dan status penyelesaian request ticket.

Keluhan yang dapat disampaikan PIC Asuransi dengan melampirkan beberapa keterangan, yaitu:

- a. Nama
- b. E-mail
- c. Kategori Masalah
- d. Subject Masalah
- e. Gejala

Konsep Perancangan Aplikasi *Helpdesk*

Perancangan adalah strategi untuk memecahkan dan mengembangkan solusi terbaik bagi permasalahan itu termasuk bagaimana mengorganisasi sistem kedalam subsistem – subsistem, serta alokasi subsistem – subsistem ke komponen –

komponen perangkat keras, perangkat lunak, serta prosedur – prosedur.

2.2.6. Pengertian *Web*

Menurut Robi Abdulloh (2015), *World Wide Web* atau WWW atau juga dikenal dengan *WEB* adalah salah satu layanan yang digunakan oleh pengguna komputer yang terhubung ke internet. *Web* ini menyediakan informasi bagi pengguna computer yang terhubung ke internet dari sekedar informasi yang kurang berguna atau informasi yang tidak berguna sama sekali sampai informasi yang serius, dari informasi yang gratisan sampai informasi yang harus berbayar. *Website* dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya itu baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman.

Web dibagi menjadi 2 jenis, antara lain:

a. Web Statis

Web Statis adalah sebuah web yang isi halaman tidak bisa dirubah-rubah dan tidak memiliki *database* sebagai tempat penyimpanan kontennya. Apabila ingin mengubah isi dari *website*, harus melakukan perubahan pada file perogram terdsebut. Contoh pemograman statis ini adalah HTML dan CSS Contoh website statis adalah berisi profil perusahaan.

b. Web Dinamis

Web Dinamis adalah sebuah web yang isinya dapat dirubah sewaktu-waktu tanpa melalui perubahan *code* pada file web, akan tetapi melalui halaman

administrator. *Content* yang tampil dalam *web* dinamis tersimpan pada sebuah *database* sehingga orang awam pun bisa merubah isi *content web* tanpa penguasaan bahasa *pemograman web* yang luas. Biasanya *web* ini tersusun dari bahasa pemograman yaitu HTML, CSS dan berhubungan dengan *database* MySQL. Contoh website dinamis adalah seperti Friendster, Multiply, dll

2.2.7. MySQL

Menurut A. Hidayah (2019), *MySQL* adalah suatu pemrograman / sistem manajemen *database* (kumpulan data yang terstruktur) yang menggunakan basis bahasa *Structured Query Language (SQL)*. *SQL* merupakan bahasa *query* yang terstandarisasi secara internasional untuk mengakses *database relasional*. *MySQL* merupakan sistem manajemen *database* yang dapat diandalkan dan penggunaannya mudah untuk dipahami. *MySQL* didesain untuk menangani *database* yang besar dengan cepat, memiliki tingkat keamanan dan konektifitas yang tinggi.

Kehandalan suatu sistem *database* (DBMS) dapat diketahui dari cara kerja *optimizer*-nya dalam melakukan proses perintah-perintah *SQL*. Sebagai *database server*, *MySQL* dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam query data. Hal ini terbukti untuk query yang dilakukan oleh single user, kecepatan query *MySQL* bisa sepuluh kali lebih cepat dari *PostgreSQL* dan lima kali lebih cepat dibandingkan *Interbase*.

2.2.8. ERD

Menurut (Alfian et al., 2020), ERD menggambarkan hubungan antara satu entitas yang memiliki sejumlah atribut dengan entitas lain. Berikut gambar rancangan ERD untuk ticketing helpdesk:

Sukamto dan Salahudin (2014) dalam (Abdurahman et al., 2018), “*Entity Relationship Diagram*” (ERD) dikembangkan berdasarkan teori himpunan dalam bidang matematika. Entity Relationship Diagram (ERD) digunakan untuk pemodelan basis data relational.

2.2.9. Flowchart

Menurut (Ridwan, 2019), “*Flowchart* adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. *Flowchart* merupakan cara penyajian dari suatu algoritma” (Ginting, 2013).

Menurut (Pt & Mega, 2020), *Flowchart* atau bagan alir adalah “Bagan (chart) yang menunjukkan alir (flow) di dalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi.

Sedangkan Menurut Romney & Steinbart dalam (Rofifah, 2020), Bagan alir (*flowchart*) adalah teknik analitis bergambar yang digunakan untuk menjelaskan beberapa aspek dari sistem informasi secara jelas, ringkas, dan logis. Bagan alir mencatat cara proses bisnis dilakukan dan cara dokumen mengalir melalui organisasi.

2.2.10. Workflow

(Chaffey, 1998), Workflow merupakan alur proses kerja yang di dalamnya dapat dilakukan pelacakan dan memberikan suatu tugas kepada partisipan dengan aturan tertentu. Pada workflow partisipan bisa berarti suatu manusia, aplikasi ataupun proses yang lain. Workflow dapat membantu suatu perusahaan sehingga seluruh alur kerjanya dapat terlacak dan juga teratur sehingga bisa mengontrol suatu alur proses pekerjaan.

2.2.11. Diagnosis

Menurut Darimi (2016), Diagnosis gejala adalah istilah yang mengacu pada upaya untuk memastikan atau mengidentifikasi penyakit atau kerusakan yang dialami oleh seorang pengguna, penderita atau masyarakat. Berdasarkan pendapat para ahli, dapat disimpulkan bahwa diagnosis adalah proses yang menentukan jenis masalah dengan menyelidiki latar belakang penyebabnya atau dengan menganalisis gejala-gejala yang terlihat.

2.2.12. Gap Analisis

Gap analysis adalah alat atau proses mengidentifikasi kesenjangan dan perbedaan antara situasi organisasi saat ini dan apa yang seharusnya di organisasi, dan digunakan untuk merancang rencana implementasi organisasi dan untuk meningkatkan efektivitas pekerjaan (Kim & Ji 2018).

2.2.13. Conceptual Data Model (CDM)

Menurut Diaz,Doddy (2016) , Sebuah Conceptual Data Model (CDM) menggambarkan secara keseluruhan konsep struktur basis data yang dirancang untuk suatu aplikasi. CDM merupakan konsep yang berkaitan dengan pandangan pemakai terhadap

data yang disimpan dalam basis data. CDM dibuat sudah dalam bentuk tabel-tabel tanpa tipe data yang menggambarkan relasi antar tabel untuk keperluan implementasi ke basis data. CDM merupakan hasil penjabaran lebih lanjut dari ERD.

2.2.14. Physical Data Model (PDM)

Menurut Diaz, Doddy (2016) , Sebuah Physical Data Model (PDM) menggambarkan secara detail konsep rancangan struktur basis data yang dirancang untuk suatu program aplikasi. PDM merupakan hasil *generate* dari CDM.