ISSN: 2303-1425 E-ISSN: 2580-720X

J-INTECH

Journal of Information and Technology

Volume 07 Nomor 01, Bulan Juni Tahun 2019

STIKI

SEKOLAH TINGGI INFORMATIKA & KOMPUTER INDONESIA
Jl. Raya Tidar 100 Malang, 65146
Telp. (0341)560823, Fax (0341)562525

ISSN: 2303-1425 E-ISSN: 2580-720X

J-INTECH

Journal of Information and Technology

Volume 07 Nomor 01, Bulan Juni 2019

LEMBAGA PENELITIAN DAN PENGABDIAN KEPADA MASYARAKAT
STIKI
SEKOLAH TINGGI INFORMATIKA & KOMPUTER INDONESIA
Jl. Raya Tidar 100, Malang; Phone: 0341-560823; Fax: 0341-562525; http://www.stiki.ac.id; mail@stiki.ac.id

PENGANTAR REDAKSI

J-INTECH merupakan jurnal yang diterbitkan oleh Sekolah Tinggi Informatika dan Komputer Indonesia Malang guna mengakomodasi kebutuhan akan perkembangan Teknologi Informasi serta guna mensukseskan salah satu program DIKTI yang mewajibkan seluruh Perguruan Tinggi untuk menerbitkan dan mengunggah karya ilmiah mahasiswanya dalam bentuk terbitan maupun jurnal online.

Pada edisi ini, redaksi menampilkan beberapa karya ilmiah mahasiswa yang mewakili beberapa mahasiswa yang lain, yang dianggap cukup baik sebagai media pembelajaran bagi para lulusan selanjutnya.

Tentu saja diharapkan pada setiap penerbitan memiliki nilai lebih dari karya ilmiah yang dihasilkan sebelumya sehingga merupakan nilai tambah bagi para adik kelas maupun pihak-pihak yang ingin studi atau memanfaatkan karya tersebut selanjutnya.

Pada kesempatan ini kami juga mengundang pihak-pihak dari PTN/PTS lain sebagai kontributor karya ilmiah terhadap jurnal J-INTECH, sehingga Perkembangan IPTEK dapat dikuasai secara bersama- sama dan membawa manfaat bagi institusi masing-masing.

Akhir redaksi berharap semoga dengan terbitnya jurnal ini membawa manfaat bagi para mahasiswa, dosen pembimbing, pihak yang bekerja pada bidang Teknologi Informasi serta untuk perkembangan IPTEK di masa depan.

REDAKSI

ISSN: 2303-1425 E-ISSN: 2580-720X

J-INTECH

Journal of Information and Technology

Volume 07 Nomor 01, Bulan Juni 2019

DAFTAR ISI

Sistem Informasi Penyedia Pemandu Wisata dengan Metode *Simple Additive Weighting (SAW)* Berbasis Android..... 01-16
Rizky Fitriyanto, Eva Handriyantini, Johan Ericka W.P.

Rancangan Bangun Pelayanan Tambal Ban Online Menggunakan Algoritma *Djikstra*..... 17-34
Muhammad Henry Setiawan, Subari

Sistem Pendukung Keputusan Rumah Tangga Miskin (RTM) untuk Program Rastra Di Desa Lolo Menggunakan Metode *Weighted Product* Berbasis Web..... 35-43
Fitri Dayanti, Sugeng Widodo

Sistem Pakar Diagnosa Penyakit Pada Hewan Ternak Sapi Menggunakan *Case Based Reasoning (CBR)* Berbasis Android..... 44-57
Rachmania Indah Permata Sari, Diah Arifah Prastiningtyas, Subari

Game Multiplayer "Mini Car Circuit" Berbasis Android..... 58-66
Syaifuddin Yudha Saputra, Subari

Sistem Penunjang Keputusan Simulasi Penentuan Akreditasi Sekolah Dasar Menggunakan Metode *Analytic Network Process* (ANP) Berbasis Web (Studi Kasus Korwil Dinas Pendidikan Kecamatan Pakisaji)..... 67-71
Ahmad Syaifullah, Laila Isyriyah, Bagus Kristomoyo Kristanto

Penerapan Algoritma *Nearest Neighbor* Untuk Menentukan Rekomendasi Solusi Terhadap Layanan Kantor Teknologi Informasi STIKI Malang 72-79
Yosua Kristanto, Diah Arifah Prastiningtyas, Meivi Kartikasari

Klasifikasi *E-book* Berbahasa Inggris Dengan Menggunakan Metode *K-Means Clustering* Studi Kasus Perpustakaan STIKI Malang..... 80-85
Willyanto Sutikno, Jozua Ferjanus Palandi, Chaulina Alfianti Oktavia

Aplikasi Presensi Mahasiswa Berbasis Wifi *Direct Peer to Peer* pada Perangkat Android (Studi Kasus : STIKI Malang)..... 86-89
Khoirun Nafisah, Koko Wahyu Prasetyo

Penerapan Metode Jaringan Syaraf Tiruan Radial Basis Function Untuk Identifikasi Jenis Mangga Berdasarkan Pola Daun 90-96
Rhesal Mahadyanto, Diah Arifah Prastiningtyas, Febry Eka Purwiantono

ISSN: 2303-1425 E-ISSN: 2580-720X

J-INTECH

Journal of Information and Technology

Volume 07 Nomor 01, Bulan Juni 2019

Pelindung

Yayasan Perguruan Tinggi Teknik Nusantara

Penasehat

Ketua STIKI

Pembina

Kepala Program Studi

Editor In Chief

Subari, S.Kom, M.Kom

Section Editor

Siti Aminah, S.Si, M.Pd

Layout Editor

Nira Radita, S.Pd., M.Pd

Tata Usaha/Administrasi

Muh. Bima Indra Kusuma

Penerapan Algoritma *Nearest Neighbor* untuk Menentukan Rekomendasi Solusi Terhadap Layanan Kantor Teknologi Informasi STIKI Malang

Yosua Kristanto[1], Diah Arifah Prastiningtyas[2], Meivi Kartikasari[3]
[1,2]Program Studi Teknik Informatika, Sekolah Tinggi Informatika & Komputer Indonesia (STIKI) Malang
[3]Program Studi Manajemen Informatika, Sekolah Tinggi Informatika & Komputer Indonesia (STIKI) Malang
[1]yosuakristanto17@gmail.com, [2]diah@stiki.ac.id, [3]meivi.k@stiki.ac.id

ABSTRAK

Knowledgebase adalah sebuah sistem yang terdiri dari heuristik sejumlah rule-rule / aturan-aturan yang tersusun secara sistemtis dan spesifik, juga relasi antara data dan aturan / rule dalam pengambilan kesimpulan. Dengan adanya knowledgebase pengetahuan yang dimiliki para pakar akan tetap terjaga dan dapat disalurkan kepada rekan kerja yang lain pada saat seorang pakar sudah tidak lagi bekerja atau berada pada instansi atau divisi terkatit. Untuk mendukung sistem dalam menentukan rekomendasi solusi penanganan terhadap layanan Kantor Teknologi Informasi STIKI Malang digunakan algoritma Nearest Neighbor. Algoritma Nearest Neighbor merupakan teknik sederhana untuk mencari jarak terdekat dari tiap-tiap kasus (cases) yang ada didalam database, dan seberapa mirip ukuran kemiripan (similarity) setiap source case yang ada didalam database dengan target case. Dalam pengujian algoritma Nearest Neighbor untuk menentukan rekomendasi solusi penanganan, dilakukan sampling terhadap data keluhan yang masuk ke KTI STIKI Malang. Penulis berharap dengan diterapkannya algoritma Nearest Neighbor ini dapat memperoleh tingkat akurasi yang cukup kuat.
***Kata Kunci**: Algoritma, Metode, Website, Nearest Neighbor*

1. PENDAHULUAN

Internet merupakan salah satu teknologi informasi yang sangat diminati oleh banyak kalangan. Berkat adanya internet berbagai hal dapat dilakukan dengan lebih mudah, seperti kegiatan mencari dan menyebarkan informasi. Hal ini tentu memberikan dampak positif yang bersifat menguntungkan, namun dibalik itu juga terdapat dampak negatif yang dapat merugikan pihak tertentu seperti menyebarkan komentar-komentar negatif yang bersifat kritik kepada suatu pihak yang dilakukan tidak pada tempatnya sehingga dapat memberikan kesan negatif kepada pihak yang dikritik.

Kritik adalah suatu penilaian yang dikemukakan dalam tulisan atau lisan mengenai suatu hal tertentu berdasarkan tolak ukur atau kaidah tertentu. Berdasarkan jenisnya kritik dibagi menjadi dua, yaitu kritik yang bersifat membangun (dapat berupa saran) dan kritik yang bersifat keluhan.

Pada umumnya keluhan tercipta sebagai akibat dari kejadian yang tidak diinginkan atau hal yang terjadi tidak sesuai harapan. Keluhan dapat bersifat membangun apabila disampaikan kepada pihak dan tempat yang tepat. Namun dapat menjadi isu negatif dan penebar kebencian apabila tidak disampaikan dengan tepat dan dikonsumsi oleh masyarakat yang tidak memiliki dasar pengetahuan akan hal terkait. Pada penelitian yang berjudul "Sistem Pakar *Troubleshooting* Jaringan Komputer Menggunakan Algoritma *Backward Chaining*" (Widianto, Firlian, 2017) yang pernah dilakukan sebelumnya mengemukakan kelemahan pada metode yang digunakan yaitu tidak adanya *knowledge base system* yang mampu menyimpan keluhan/masalah beserta solusi penanganannya. Sedangkan kekurangan pada metode *Backward Chaining* adalah efisiensi. Sistem *Backward Chaining* memudahkan pencarian *depth first search*, namun sistem tidak efisien dalam hal pencarian penyelesaiannya (Azizah, 2013).

Berdasarkan permasalahan yang telah dikemukakan sebelumnya dapat disimpulkan bahwa tidak adanya sebuah *knowledge base system* yang dapat menyimpan permasalahan dan memberikan rekomendasi solusi penanganan keluhan. Menurut penelitian yang terdahulu metode yang efektif dalam hal menyimpan dan memberikan rekomendasi solusi penanganan adalah *Nearest Neighbor* (NN) yang memiliki tingkat akurasi sebesar 85%. NN merupakan metode yang mampu mencari jarak terdekat dari tiap-tiap kasus (*cases*) yang ada di dalam *database*, dan seberapa mirip ukuran kemiripan (*similarity*) setiap *source case* yang ada di dalam *database* dengan target *case* (Kartikasari, 2015).

2. ANALISA DAN PERANCANGAN

Untuk mendapatkan *output* berupa rekomendasi solusi penanganan menggunakan algoritma *Nearest Neighbor* diperlukan 4 tahapaan, di antaranya:

- Merancang metode NN.
- Melakukan *filter* keluhan pada *knowledgebase*.

- Mencari nilai *similarity* setiap kasus baru dengan kasus lama.
- Menentukan nilai *similarity* tertinggi.
- Mengadaptasi rekomendasi solusi dari *knowledgebase*.

Merancang metode NN

Sebelum melakukan perhitungan menggunakan algoritma *Nearest Neighbor* untuk menentukan rekomendasi solusi harus ditentukan standar pembobotan terlebih dahulu. Standar pembobotan ditunjukkan pada tabel 1.

Tabel 1. Standar Pembobotan

	Keluhan	**Detail keluhan**	**Kategori keluhan**	**Fasilitas Untuk**
Bobot *similarity*	1 atau 0,3	1 atau 0,3	1 atau 0,5	1 atau 0,5
Bobot kritaria	3	7	5	9

Agar sistem mampu memberikan rekomendasi solusi penanganan yang tepat maka data kasus baru yang masuk harus melalui 3 tahap pengelolaan, yaitu tahap mencari *similarity*, tahap mencari *similarity* tertinggi, serta tahap mengambil rekomendasi solusi penanganan dari *database*. Penentuan rekomendasi solusi penanganan akan dicontohkan dengan kasus baru yang ditunjukkan pada tabel 2.

Tabel 2. Tabel Keluhan Baru

Jenis keluhan	Keluhan	Detil keluhan	Kategori keluhan	Fasilitas untuk	Penyebab	Solusi
Wifi	Wifi bermasalah	Sinyal wifi rendah	Normal	Maha siswa/ umum	?	?

Melakukan Filter Keluhan pada *Knowledgebase*

Tahap pertama adalah melakukan filter terhadap keluhan "wifi".

Tabel 3. Tabel Filter Keluhan "wifi"

No	Keterangan		
1	*Input* data	Jenis keluhan	Wi-Fi
		Wifi bermasalah	Wifi bermasalah
		Detil keluhan	Jaringan wifi lambat
		Kategori keluhan	Normal
		Fasilitas untuk	Mahasiswa/umum
	Output data	Penyebab	Terjadi interferensi sinyal *wireless*
		Solusi	*Reboot* ulang *Access Point*
2	*Input* data	Jenis keluhan	
		Wifi bermasalah	
		Detil keluhan	SSID hilang
		Kategori keluhan	*High*
		Fasilitas untuk	*Staf/unit*
	Output data	Penyebab	*Access Point* mati
		Solusi	*Reboot* pada *Access Point* dan memastikan *access point* di area tersebut aktif
3	*Input* data	Jenis keluhan	
		Wifi bermasalah	POE
		Detil keluhan	POE bermasalah
		Kategori keluhan	Normal
		Fasilitas untuk	Staf/unit
	Output data	Penyebab	Listrik tidak stabil
		Solusi	Mengganti perangkat yang baru

Langkah selanjutnya adalah membandingkan kasus baru dengan semua kasus lama pada hasil *filter* dan dicari bobot *similarity* nya dengan menggunakan rumus perhitungan berikut ini:

$$Total\ similarity = \frac{\sum_{i=1}^{n} W_i\ .\ sim(f_i^T, f_i^S)}{\sum_{i=1}^{n} W_i}$$

Merancangn Metode NN

Mencari Nilai Similarity Setiap Kasus Baru dengan Kasus Lama

Hasil perhitungan nilai *similarity* antara kasus baru dengan kasus lama yang pertama ditunjukkan pada tabel 4.

Tabel 4. Tabel *Similarity* Kasus Baru Dengan Kasus Lama 1

Kasus Lama	Kasus Baru	Bobot	Similarity	Bobot x Similarity
Wifi bermasalah	Wifi bermasalah	3	1	3
Jaringan wifi lambat	Sinyal wifi rendah	7	0.3	2.1
Normal	Normal	5	1	5
Mahasiswa /umum	Mahasiswa /umum	9	1	9

$$\text{Similarity } (1,1) = \frac{(3 + 2.1 + 5 + 9)}{3 + 7 + 5 + 9}$$
$$= \frac{19.1}{24}$$
$$= 0.795833$$

Hasil perhitungan nilai *similarity* antara kasus baru dengan kasus lama yang kedua ditunjukkan pada tabel 5.

Tabel 5. Tabel *similarity* kasus baru dengan kasus lama 2.

Kasus Lama	Kasus Baru	Bobot	Similarity	Bobot x Similarity
Wifi bermasalah	Wifi bermasalah	3	1	3
SSID hilang	Sinyal wifi rendah	7	0.3	2.1
High	Normal	5	0.5	2.5
Staf/unit	Mahasiswa /umum	9	0.5	4.5

$$Similarity\ (2,1) = \frac{(3 + 2.1 + 2.5 + 4.5)}{3 + 7 + 5 + 9}$$

$$= \frac{12.1}{24}$$
$$= 0.504166$$

Hasil perhitungan nilai *similarity* antara kasus baru dengan kasus lama yang ketiga ditunjukkan pada tabel 6.

Tabel 6. Tabel *Similarity* Kasus Baru Dengan Kasus Lama 3

Kasus Lama	Kasus Baru	Bobot	Similarity	Bobot x Similarity
POE	Wifi bermasalah	3	0.3	0.9
POE bermasalah	Sinyal wifi rendah	7	0.3	2.1
Normal	Normal	5	1	5
Staf /unit	Mahasiswa /umum	9	0.5	4.5

$$Similarity\ (3,1) = \frac{(0.9 + 2.1 + 5 + 4.5)}{3 + 7 + 5 + 9}$$
$$= \frac{12.5}{24}$$
$$= 0.520833$$

Setelah dicari nilai *similarity* antar kasus, maka langkah berikutnya adalah mencari jarak terdekat antar kasus. Berdasarkan detil perhitungan nilai *similarity* antara kasus baru dengan 3 contoh kasus lama pada *database* kasus, maka nilai *similarity* dari masing-masing kasus dapat ditunjukkan dalam tabel 7.

Tabel 7. Tabel nilai *similarity*

Kasus	Nilai *similarity*
Kasus lama 1 dengan kasus baru 1	0.795833
Kasus lama 2 dengan kasus baru 1	0.504166
Kasus lama 3 dengan kasus baru 1	0.520833
Nilai *similarity tertinggi*	**0.795833**

Menentukan Nilai *Similarity* Tertinggi

Jarak terdekat antara kasus lama dengan kasus baru ditandai dengan nilai *similarity* tertinggi. Solusi yang akan diberikan oleh sistem pada kasus baru berdasarkan pada nilai *similarity* tertinggi antara kasus baru dengan kasus lama nomor 1. tabel 7 menunjukkan bahwa kasus baru memiliki *similarity* tertinggi (0.795833) dengan kasus lama nomor 1.

Mengadaptasi Rekomendasi Solusi dari *database*

Tahap selanjutnya adalah tahap mengambil rekomendasi solusi dari *database*. Solusi yang tampak pada tabel 8 sebuah kasus dianggap mutlak mirip jika nilai *similarity* nya adalah mendekati 1 dan dianggap tidak mirip sama sekali apabila nilai *similarity*-nya adalah 0. Jika terdapat 2 (dua) kasus atau lebih yang memiliki nilai *similarity* sama dan tertinggi, maka solusi yang diadaptasi adalah dari kasus yang terbaru pada *database*.

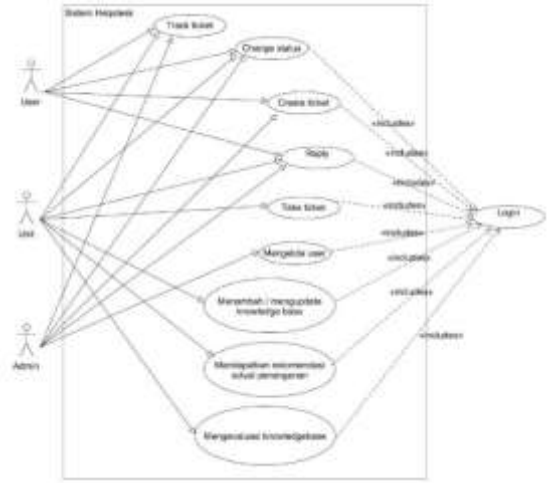
Tabel 8. Tabel Rekomendasi Solusi

Jenis keluhan	Keluhan	Detil keluhan	Kategori keluhan	Fasilitas untuk	Penyebab	Solusi
Wi-Fi	Wifi berasalah	Jaringan wifi lambat	Normal	Mahasiswa /umum	Terjadi interferensi sinyal wireless	*Reboot* ulang *Access Point*

Perancangan Algoritma

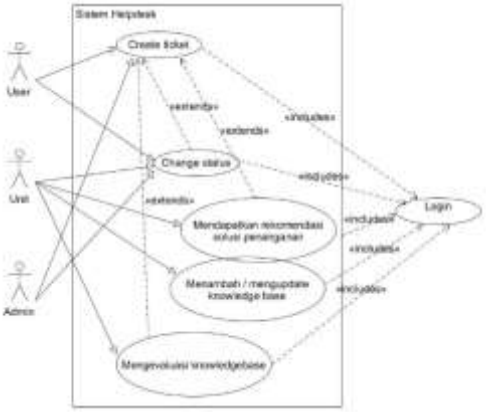
Untuk memberikan solusi atas keluhan menggunakan algoritma *Nearest Neighbor*. Alasan digunakannya algoritma *Nearest Neighbor* adalah karena algoritma ini dapat mengukur kedekatan antara keluhan yang masuk dengan kasus-kasus sebelumnya yang ada pada *knowledge base*.

Untuk gambaran sistem keluhan yang akan dibangun ditunjukkan pada bagan *use case* dalam gambar 1.



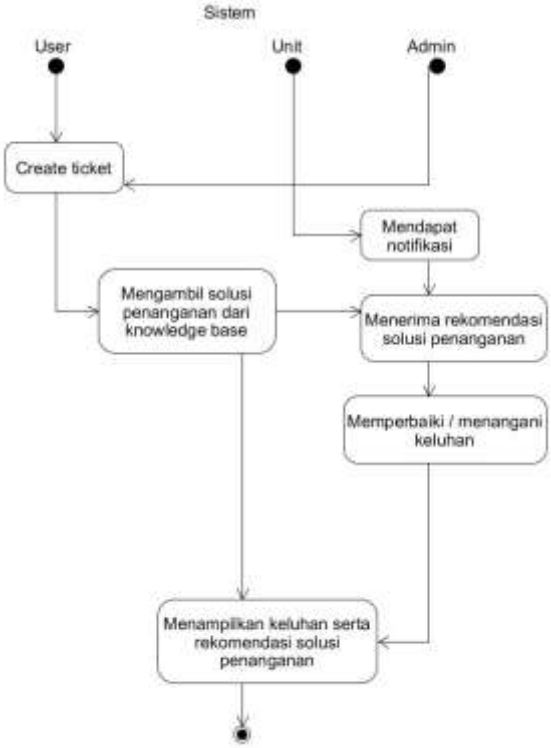
Gambar 1. *Use Case* Sistem *Helpdesk*

Untuk dapat mengubah status keluhan dan mendapatkan rekomendasi solusi penanganan, pengguna harus *login* ke dalam sistem terlebih dahulu. UML proses mengubah status dan mendapatkan solusi penanganan ditunjukkan dalam gambar 2.



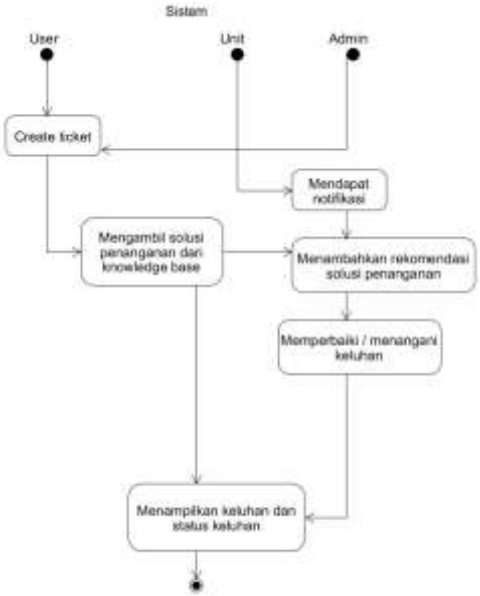
Gambar 2. *Use Case* Proses Mengubah Status dan Mendapatkan Rekomendasi Solusi

Proses penanganan keluhan dengan data kasus yang telah ada pada *database* dijelaskan didalam Gambar 3.



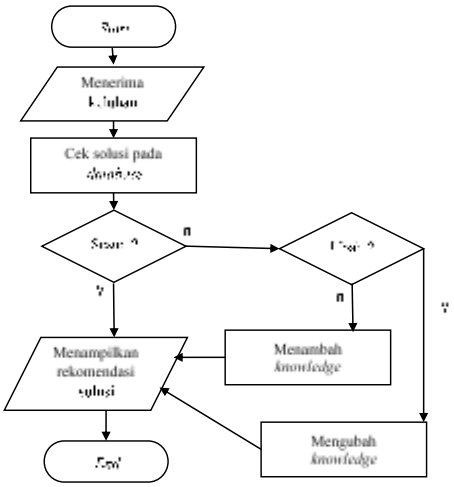
Gambar 3. *Activity Diagram* Proses Penanganan Kasus Serupa

Proses penanganan keluhan dengan data kasus yang belum ada dalam *database* ditunjukkan dalam gambar 4.



Gambar 4. *Activity Diagram* Proses Penanganan Kasus Baru

Agar *knowledge base* dapat selalu berkembang dengan data penanganan kasus yang terbaru dan penanganan kasus yang belum ada, maka perlu dilakukan penambahan ilmu atau *update* pengetahuan pada basis data. Untuk proses penambahan ilmu pada *knowledge base system* dijelaskan pada bagan *flowchart* dalam gambar 5.

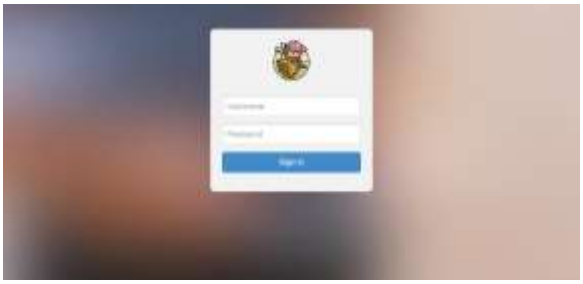


Gambar 5. *Flowchart* Menambahkan *Knowledgebase*

3. IMPLEMENTASI DAN PEMBAHASAN

Halaman Login

Pada saat aplikasi pertama dibuka, pengguna akan diarahkan menuju ke halaman *login* untuk memasukkan *username* dan *password* secara benar. Tampilan halaman *login* ditunjukkan pada gambar 6.



Gambar 6. Halaman *Login*

Halaman Dashboard

Setelah pengguna memasukkan *username* dan *password*nya dengan benar, maka pengguna akan diarahkan menuju ke halaman *dashboard* atau beranda. Pada halaman *dashboard* akan ditampilkan daftar keluhan yang pernah dilaporkan oleh pengguna. Tampilan tatap muka halaman beranda ditunjukkan pada gambar 7.

Gambar 7. Halaman *Dashboard*

Halaman Detil Keluhan

Halaman detil keluhan dapat diakses dengan klik tombol amplop di sisi kanan keluhan. Pada halaman detil keluhan akan ditampilkan keluhan yang telah dimasukkan secara detil, serta terdapat fitur *reply* pada bagian bawah detil keluhan. Tampilan menu detil keluhan ditunjukkan pada gambar 8.

Gambar 8. Halaman Detil Keluhan

Halaman *Insert* Keluhan

Pengguna yang telah *login* ke dalam aplikasi dapat mengirimkan keluhannya dengan menekan tombol "Tambah keluhan" pada halaman beranda. Pengguna diharuskan mengisi jenis keluhan, keluhan, serta detil keluhan dan keterangan yang bersifat opsional. Tampilan *insert* jenis keluhan ditunjukkan pada gambar 9.

Gambar 9. Halaman *Insert* Jenis Keluhan

Setelah pengguna memasukkan jenis keluhan, maka pengguna akan diarahkan menuju halaman *insert* keluhan seperti yang ditunjukkan pada gambar 10.

Gambar 10. Halaman *Insert* Keluhan

Langkah berikutnya setelah pengguna memasukkan keluhannya adalah pengguna wajib untuk mengisikan apa yang menjadi detil keluhannya melalui sebuah *form* detil keluhan seperti yang ditunjukkan pada gambar 11.

Gambar 11. Halaman *Insert* Detil Keluhan

Langkah terakhir untuk memasukkan keluhan pengguna adalah dengan mengisi *form* kategori keluhan, pada *form* ini pengguna diharuskan untuk memilih salah satu dari beberapa pilihan kategori keluhan serta fasilitas milik. Pada *form* ini juga terdapat teks area yang mengijinkan pengguna untuk memberikan informasi tambahan atas keluhan yang dilaporkan. *Form input* kategori keluhan ditunjukkan pada gambar 12.

Gambar 12. Halaman *Insert* Kategori Keluhan

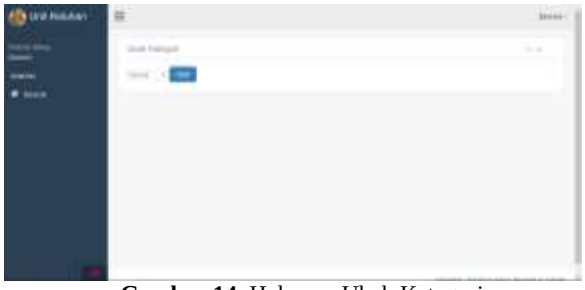
Apabila pengguna tidak menemukan perihal yang ingin dilaporkan pada saat *insert* keluhan dan detil keluhan, maka pengguna dapat memilih opsi lainnya kemudian menuliskan apa yang menjadi keluhannya. Namun apabila pengguna tidak menemukan apa yang ingin dilaporkan pada *form* pertama (*form input* jenis keluhan) maka pengguna juga diberikan opsi lainnya yang akan diarahkan

menuju halaman *insert* keluhan lainnya seperti yang ditunjukkan pada gambar 13.

Gambar 13. Halaman *Insert* Keluhan Lainnya

Halaman Ubah Variabel

Setelah keluhan diterima oleh staf, staf yang bersangkutan dapat mengganti isi dari variabel "kategori" dan "fasilitas milik" yang memiliki pengaruh terhadap perhitungan algoritma *Nearest Neighbor* untuk mencari nilai *similarity*. Selain itu, ada juga variabel yang juga dapat diganti oleh staf dan pengguna yaitu variabel "status" yang berisi "open" apabila keluhan belum ditangani, "hold" apabila keluhan sedang dalam pengerjaan, serta "close" apabila keluhan telah dikerjakan. Tampilan *form* ganti "kategori" ditunjukkan pada gambar 14.



Gambar 14. Halaman Ubah Kategori

Tampilan halaman ubah fasilitas ditunjukkan pada gambar 15.

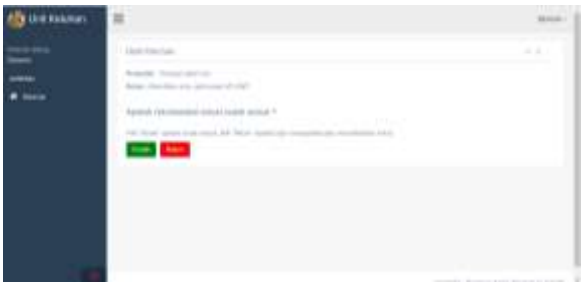
Gambar 15. Halaman Ubah Fasilitas

Tampilan halaman ubah status ditunjukkan pada Gambar 16.

Gambar 16. Halaman Ubah Status

Halaman *Get Solution*

Pada halaman detil keluhan milik staf, terdapat sebuah tombol yang akan mengarah ke *form get solution* yang menampilkan "penyebab" dan rekomendasi "solusi" penanganan yang diambil dari basis data pengetahuan melalui perhitungan algoritma *Nearest Neighbor*. Pada *form* tersebut juga terdapat pertanyaan apakah jawaban yang didapat sudah sesuai atau belum. *Form get solution* ditunjukkan pada gambar 17.



Gambar 17. *Form Get Solution*

Pada *form get solution* akan disediakan dua pilihan yaitu "sudah", yang berarti rekomendasi solusi penanganan sudah tepat, dan "belum" yang berarti rekomendasi solusi penanganan belum tepat. Apabila staf memilih "sudah" maka staf akan dibawa kembali menuju halaman detil keluhan, dengan kolom "penyebab" dan "solusi" yang telah terisi dengan jawaban yang telah diambil dari database oleh sistem menggunakan metode *Nearest Neighbor*.

Namun apabila staf memilih "belum", maka staf akan disajikan pertanyaan berikutnya dengan dua tombol pilihan yaitu "tambah *knowledgebase*" untuk menambahkan pengetahuan pada basis data pengetahuan apabila kasus tersebut adalah kasus baru. Atau "*edit knowledge*" untuk mengubah atau memodifikasi basis data pengetahuan apabila "penyebab" serta rekomendasi "solusi" penanganan dirasa kurang sesuai dan perlu dilakukan perubahan. *Form* modifikasi basis data pengetahuan ditunjukkan pada gambar 18.

Gambar 18. *Form* Modifikasi Basis Data Pengetahuan

Apabila staf memilih "tambahkan *knowledgebase*" maka staf akan diarahkan menuju halaman *form* tambah *knowledge*. *Form* tambah *knowledge* ditunjukkan pada gambar 19.

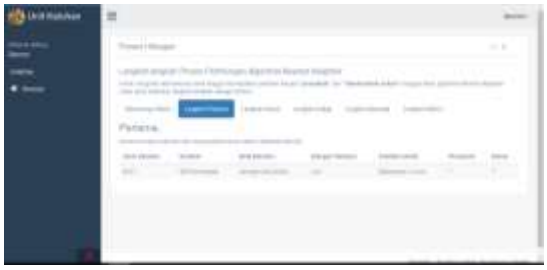
Gambar 19. *Form* Menambah Basis Data Pengetahuan

Sedangkan apabila staf memilih "*edit knowledge*" maka staf akan diarahkan menuju *form* ubah *knowledge*. *Form* ubah *knowledge* ditunjukkan pada gambar 20.

Gambar 20. *Form* Ubah Basis Data Pengetahuan

Halaman Algoritma

Pada halaman ini akan ditampilkan langkah-langkah, mulai dari rancangan sistem *Nearest Neighbor*, langkah pertama sampai dengan langkah terakhir secara rinci sebagai bukti digunakannya metode *Nearest Neighbor*. Tampilan halaman algoritma ditunjukkan pada gambar 21.



Gambar 21. Halaman algoritma *Nearest Neighbor*

Menu Pengaturan Pengguna

Pada halaman admin terdapat sebuah menu pengaturan pengguna, yaitu sebuah menu yang memiliki fungsi memodifikasi data pengguna yang hanya dapat diakses oleh *user* yang memiliki hak akses administrator. Tampilan menu pengaturan pengguna ditunjukkan pada gambar 22.

Gambar 22. Halaman Menu Pengaturan Pengguna

4. KESIMPULAN DAN SARAN

Setelah dilakukan analisa, perancangan, implementasi, dan pengujian algoritma *Nearest Neighbor* untuk menentukan rekomendasi solusi penanganan maka dapat disimpulkan bahwa aplikasi yang telah dibuat kurang tepat untuk menentukan rekomendasi solusi terhadap layanan Kantor Teknologi Informasi STIKI Malang berdasarkan kasus sebelumnya pada basis data pengetahuan. Pada saat tahap evaluasi penerapan algoritma ini, penulis mengambil kuisioner kepada sejumlah pakar untuk mengukur tingkat kepuasan yang didapat. Tingkat kepuasan yang dihasilkan adalah 46%, hal ini disebabkan karena pada saat memberikan kuisioner belum diadakan pelatihan atau arahan penggunaan sistem.

Berdasarkan penelitian yang telah dilakukan, membutuhkan beberapa syarat yang dapat dijadikan acuan pengembangan. Beberapa saran yang dapat dijadikan acuan untuk pengembangan aplikasi atau sistem lebih lanjut adalah sebagai berikut:

- Penambahan fitur notifikasi melalui email meningkatkan respon staf dalam menanggapi keluhan yang masuk.
- Disarankan perlu dilakukan riset lebih lanjut dengan menggunakan algoritma lain untuk mengetahui apakah hasil akurasi lebih baik daripada algoritma Nearest Neighbor.

5. REFERENSI

[1] Burhanudin. (2006). "Analisis Atas Kepuasaffdan Penyesalan Pada Niat Melakukan Pembelian Ulang dan Niat Melakukan Komplain".

[2] Dewobroto, W. (2005). "Simulasi Numerik Berbasis Komputer Sebagai Solusi Pencegah Bahaya Akibat Kegagalan Bangunan". Vol. 2 No. 2 Oktober 2006: 108-125.

[3] Dharwiyati, S. (2003). "Pengantar Unified Modeling Language (UML)". Ilmukomputer.com.

[4] Feldman, R., & Sanger, J. (2007). "The Text Mining Handbook Advanced Approaches in Analyzing Unstructured Data". New York: Cambridge University Press.

[5] Ganesan, K. (2006). "Text Mining, Analytics & More".

[6] Han, J., & Kamber, M. (2006). "Data Mining Concepts and Techniques".

[7] Kartikasari, M. (2015). "Penerapan Case Base Reasoning pada Sistem Pendukung Keputusan Penanganan Komplain Penyewa Mall". Universitas Brawijaya.

[8] Marlinda. (2004). Sistem Basis Data Yogyakart. ANDI.

[9] Putri. (2012). Sistem Pendukung Keputusan Cerdas Dalam Penentuan Penerima Beasiswa". Politeknik Negeri Bali.

[10] Ritayani. (2016). "Pengantar Algoritma dan Pemrograman". Fakultas Ilmu Komputer Universitas Almuslim.

[11] Tarmuji, Ali. 2008. "Tinjauan Umum Tentang Helpdesk Dan Framework Terkait". Fakultas Teknik Industri Universitas Ahmad Dahlan.

[12] Triswan Y. (2009). "Pengenalan PHP". IlmuKomputer.com.

[13] Turban, E., Aronson, J.E. & Liang, T.P. 2005. "Decision Support System and Intelligent System". New Jersey: Pearson Education. Inc.

[14] Utami. (2014). "Pengertian Sistem Informasi". Universitas Widyatama.

[15] Zohar, Y. E. (2002). "Introduction to Text Mining: Automated Learning Group National Center For Supercomputing Application". Illinois: University of Illioni.