

SISTEM PENUNJANG KEPUTUSAN LOKASI TERBAIK JUAL BAKSO DI KOTA MALANG DENGAN METODE BASIS DATA FUZZY

Agis Arizal Fahri, Dinny Wahyu Widarti

Program Studi Teknik Informatika STMIK PPKIA Pradnya Paramita Malang

agisfahri@gmail.com

Abstract

The meatballs are very popular and can be found throughout Indonesia from the cart street vendor to big restaurant. Emerging entrepreneurs meatballs good food at a roadside Malang and building a medium-sized or large like kiosks, shops and malls most locations are not chosen properly and appropriately. The purpose of this research is a decision support system by determining the best location in town selling meatballs poor with fuzzy database method. The method applied in this study is developing a system (SDLC) Software Development Life Cycle. Based on test results concluded that the Decision Support System Application provides information determining the best location to display the options that facilitate decisions on employers in determining the best location and the right to run a business

Keywords: *Decision Support Systems, Location, Database Fuzzy*

1. PENDAHULUAN

Bakso merupakan makanan berbentuk bola daging yang terbuat dari campuran daging sapi giling dan tepung serta bumbu pilihan yang khas yang terkenal dari kota malang dan solo. Bakso sangat populer dan dapat ditemukan di seluruh Indonesia mulai dari gerobak pedagang kaki lima hingga rumah makan besar.

Pengusaha makanan bakso yang bermunculan baik di pinggir jalan kota Malang maupun gedung yang berukuran sedang maupun yang besar seperti kios, ruko dan mall sebagian besar tidak dipilih secara benar dan tepat. Dalam hal ini menyebabkan bertambahnya biaya operasional dan usaha bakso tidak dapat berkembang seperti biaya pembelian bahan mentah yang jauh dari pasar membuat biaya transportasi bertambah.

Setiap pengusaha pasti ingin usahanya berjalan sesuai dengan yang direncanakan sebelumnya untuk menghasilkan keuntungan yang semaksimal

mungkin. Untuk mewujudkan tujuan tersebut diperlukan sebuah strategi pemasaran dengan merencanakan sebuah bauran pemasaran (marketing mix).

Menurut Kotler dan Amstrong, bauran pemasaran (marketing mix) adalah kumpulan alat pemasaran taktis terkendali yang dipadukan perusahaan untuk menghasilkan respons yang diinginkannya di pasar. Bauran pemasaran terdiri dari semua hal yang dapat dilakukan perusahaan untuk mempengaruhi permintaan produknya. Berbagai kemungkinan ini dapat dikelompokkan menjadi empat kelompok variabel yang disebut “empat P”: Product (produk), Price (harga), Place (tempat), dan Promotion (promosi) (Kotler dan Amstrong, 2008:62).

Penulis memilih salah satu dari unsur 4P, yaitu Place (tempat) yang merupakan salah satu strategi pemasaran yang dipersiapkan dengan menentukan tempat atau lokasi berpotensi yang dapat mendorong kemajuan usaha yang dijalankannya tersebut.

Dalam pemilihan sebuah lokasi untuk memulai usaha tidaklah mudah karena dalam pemilihan lokasi harus yang strategis, butuh lokasi yang tepat dan sesuai agar usaha yang dijalankannya dapat dijangkau dan dapat diterima oleh konsumen. Oleh karena itu diperlukan sebuah sistem penunjang keputusan agar dapat menghitung macam-macam kriteria yaitu kedekatan dengan sumber daya, tersedia lokasi, kondisi lingkungan dan fasilitas yang dapat menghasilkan beberapa hasil keputusan yang terbaik untuk menunjang dalam pengambilan keputusan.

Berdasarkan latar belakang tersebut, maka tugas pemrograman khusus ini akan mengangkat judul **“Sistem Penunjang Keputusan Penentuan Lokasi Terbaik Jual Bakso di Kota Malang Dengan Metode Basis data Fuzzy”** yang mana dengan adanya sistem ini dapat mempermudah para pengusaha dalam pengambilan keputusan untuk menentukan lokasi yang tepat yang akan dijadikan tempat usaha.

2. KAJIAN LITERATUR DAN PEGEMBANGAN HIPOTESIS

2.1 Sistem Pendukung Keputusan

2.1.1 Definisi Sistem

Sistem adalah kumpulan dari elemen-elemen yang saling berinteraksi dan berintegrasi untuk mencapai suatu tujuan. Semua yang ada di dunia ini pada dasarnya merupakan sebuah sistem, baik sistem yang sederhana maupun sistem yang kompleks.

Menurut Soeherman, Bonnie dan Pinontoan, Mario (2008:3) sistem diartikan sebagai serangkaian komponen-komponen yang saling berinteraksi dan bekerja sama untuk mencapai tujuan tertentu.

Menurut Jogiyanto, sistem dapat didefinisikan dengan pendekatan prosedur dan dengan pendekatan komponen. Dengan pendekatan prosedur, sistem dapat didefinisikan sebagai kumpulan dan prosedur-prosedur yang mempunyai tujuan tertentu. Contoh sistem yang didefinisikan

dengan pendekatan prosedur ini adalah sistem akuntansi. Sistem ini didefinisikan sebagai kumpulan dan prosedur-prosedur penerimaan kas, pengeluaran kas, penjualan, pembelian dan buku besar (Jogiyanto, 2009: 34).

2.1.2 Pengertian Sistem Pendukung Keputusan

menurut alter dalam buku abdul kadir Pengenalan Sistem Informasi Sistem pendukung keputusan (SPK) atau *Decision Support System* (DSS) adalah sistem informasi interaktif yang menyediakan informasi, pemodelan, dan pemanipulasian data yang digunakan untuk membantu pengambilan keputusan pada situasi yang semistruktur dan situasi yang tidak terstruktur di mana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat (Alter, 2002).

2.2 Pengertian Lokasi

Lokasi adalah letak atau tempat.

Menurut Kotler dan Amstron (2008:63), tempat meliputi kegiatan perusahaan yang membuat produk tersedia bagi pelanggan sasaran.

2.3 Basisdata Fuzzy Model Tahani

2.3.1 Basisdata Fuzzy

Database yang umumnya kita gunakan, memiliki data yang lengkap dalam setiap tabelnya. Demikian pula, apabila hendak dibuat suatu query, maka query itupun harus menggunakan data yang ada pada tabel dan kata-kata kunci yang berlaku di **SQL**. Apabila ada data yang kurang lengkap, mengandung ketidakpastian dan ambigu, maka penggunaan database biasa menjadi sulit untuk dilakukan. Oleh karena itulah, logika fuzzy dimanfaatkan logika fuzzy untuk mengantisipasi pemanipulasian data dalam database yang mengandung ketidakpastian, baik dari sisi data maupun query-nya (Kusumadewi, 2010)

2.3.2 Logika Fuzzy

Logika Fuzzy merupakan salah satu komponen pembentuk *soft computing*. (Kusumadewi, 2010:1).

Menurut Kusumadewi (2010:2), ada beberapa alasan mengapa orang menggunakan logika Fuzzy, antara lain:

- a. Konsep logika Fuzzy mudah dimengerti. Karena logika Fuzzy menggunakan dasar teori himpunan, maka konsep matematis yang mendasari penalaran fuzzy tersebut mudah untuk dimengerti.
- b. Logika fuzzy sangat fleksibel, artinya mampu beradaptasi dengan perubahan-perubahan, dan ketidakpastian yang menyertai perubahan permasalahan.
- c. Logika fuzzy memiliki toleransi terhadap data yang tidak tepat. Jika diberikan sekelompok data yang cukup homogen, dan kemudian ada beberapa data yang “eksklusif”, maka logika fuzzy memiliki kemampuan untuk menangani data eksklusif tersebut.
- d. Logika fuzzy mampu memodelkan fungsi-fungsi nonlinier yang sangat kompleks.
- e. Logika fuzzy dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan. Dalam hal ini, sering dikenal dengan nama Fuzzy Expert System menjadi bagian terpenting.
- f. Logika fuzzy dapat bekerjasama dengan teknik-teknik kendali secara konvensional. Hal ini umumnya terjadi pada aplikasi di bidang teknik mesin maupun teknik elektro.
- g. Logika fuzzy didasarkan pada bahasa alami. Logika fuzzy menggunakan bahasa sehari-hari sehingga mudah dimengerti.

2.3.3 Himpunan Fuzzy

Menurut Kusumadewi (2010:3), pada himpunan tegas (*crisp*), nilai keanggotaan suatu item x dalam himpunan A , yang sering ditulis dengan $\mu_A(x)$, memiliki dua kemungkinan, yaitu:

- Satu (1), yang berarti suatu item menjadi anggota dalam suatu himpunan, Atau,
- Nol (0), yang berarti suatu item tidak menjadi anggota suatu himpunan.

Menurut Kusumadewi (2010:6), himpunan fuzzy memiliki 2 atribut, yaitu:

- a. Linguistik, yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti: MUDA, PROBAYA, TUA.
- b. Numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel seperti: 40, 25, 50, dsb.

Ada beberapa hal yang perlu diketahui dalam memahami sistem fuzzy, yaitu:

- a. Variabel Fuzzy
Variabel fuzzy merupakan variabel yang hendak dibahas dalam suatu fuzzy. Contoh: umur, temoeratur, permintaan, dsb.
- b. Himpunan Fuzzy
Himpuna Fuzzy merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel fuzzy.
- c. Semesta Pembicaraan
Semesta Pembicaraan merupakan keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel fuzzy.
- d. Domain
Domain himpunan fuzzy adalah keseluruhan nilai yang dizinkan dalam semesta pembicaraan dan

boleh dioperasikan dalam suatu himpunan fuzzy.

2.3.4 Fungsi Keanggotaan

Fungsi Keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik input data kedalam nilai keanggotaannya (sering disebut dengan derajat keanggotaan) yang memiliki interval 0 sampai 1.

2.3.5 Operator Dasar Zadeh untuk operasi himpunan fuzzy

Seperti halnya himpunan konvensional, ada beberapa operasi yang didefinisikan secara khusus untuk mengkombinasi dan memodifikasi himpunan fuzzy. Nilai keanggotaan sebagai hasil dari operasi 2 himpunan sering dikenal dengan nama *Fuzzy Set* atau α -predikat. Ada 3 operator dasar yang diciptakan oleh Zadeh, yaitu

1. Operasi AND

Operator ini berhubungan dengan operasi interseksi pada himpunan α -predikat sebagai hasil operasi dengan operator AND diperoleh dengan mengambil nilai keanggotaan terkecil antar elemen pada himpunan-himpunan yang bersangkutan.

$$\mu_{A \cap B} = \min(\mu_A(x), \mu_B(y))$$

2. Operasi OR

Operator ini berhubungan dengan operasi union pada himpunan α -predikat sebagai hasil operasi dengan operasi OR diperoleh dengan mengambil nilai keanggotaan terbesar antar elemen pada himpunan-himpunan yang bersangkutan.

$$\mu_{A \cup B} = \max(\mu_A(x), \mu_B(y))$$

3. Operator NOT

Operator ini berhubungan dengan operasi komplemen pada himpunan α -predikat sebagai hasil operasi dengan operator NOT diperoleh dengan mengurangi nilai keanggotaan elemen pada

himpunan yang bersangkutan dari 1.

$$\mu_{A'} = 1 - \mu_A(x)$$

2.4 UML (*Unified Modelling Language*)

UML (*Unified Modelling Language*) 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. A. S Rosa dan Salahuddin, M (2013:133).

1. Use Case Diagram

Diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat difahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor atau *use case*.

- Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat sendiri, jadi walaupun simbol dari aktor adalah orang, tapi aktor belum tentu merupakan orang.
- *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

2. Activity Diagram

Diagram *activity* adalah *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal seperti berikut:

- Rancangan proses bisnis dimana setiap urutan aktifitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
- Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antar muka tampilan.
- Rancang pengujian dimana setiap aktifitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
- Rancang menu yang ditampilkan pada perangkat lunak.

3. Class Diagram

Diagram *Class* atau *Class* diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

- Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
- Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

4. Sequence Diagram

Diagram *Sequence* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram *sequence* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

2.5 Basis Data (Database)

Menurut A. S, Rosa dan Shalahuddin, M (2013:43), basis data adalah sistem terkomputerisasi yang tujuannya utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Namun

pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat.

2.6 ERD (Entity Relationship Diagram)

ERD menurut A.S. Rosa, M. Salahuddin (2013:49), ERD (*Entity Relationship Diagram*) adalah pemodelan awal basis data yang paling banyak digunakan. ERD digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan ERD.

2.7 Kamus Data

Kamus data berfungsi membantu pelaku sistem untuk mengartikan aplikasi secara detail dan mengorganisasi semua elemen data yang digunakan dalam sistem secara persis sehingga pemakai dan penganalisis sistem mempunyai dasar pengertian yang sama tentang masukan, keluaran, penyimpanan dan proses.

Kamus data sering disebut juga dengan sistem data *dictionary* adalah catalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi.

Tahap perancangan sistem, kamus data digunakan untuk merancang input, merancang laporan-laporan dan *database* yang dibuat berdasarkan arus data yang ada di DAD (Diagram Arus Data), (Ladjamudin, 2005:70-71). Jika menggunakan UML, maka kamus data berdasarkan pada Class Diagram.

2.8 MYSQL

Menurut Firdaus (2007:12). MySQL atau dibaca “mai-se-kuel” adalah sebuah program pembuat *database* yang bersifat open source, yang artinya siapa saja boleh menggunakannya.

MySQL sebenarnya produk yang berjalan pada *platform* linux, karena sifatnya open source. MySQL dapat dijalankan pada semua platform baik *Windows*, *Linux*, MySQL juga merupakan program pengakses *database* yang bersifat jaringan sehingga

dapat digunakan untuk aplikasi multi user (banyak user).

Sebenarnya *software* MySQL mempunyai 2 macam lisensi. Lisensi pertama bersifat open source dengan menggunakan GNU (*General Public License*) dan lisensi kedua berupa *Standart Comercial License* yang dapat dibeli dari MySQL AB.

MySQL merupakan sistem manajemen *database*. *Database* merupakan struktur dari penyimpanan data. Untuk menambah, mengakses, dan memproses data yang disimpan dalam sebuah *database* komputer maka diperlukan sistem manajemen *database* seperti MySQL Server.

MySQL merupakan sistem manajemen *database* terhubung (*relation database management system*). *Database* terhubung menyimpan data pada tabel-tabel terpisah. Hal tersebut akan menambah kecepatan dan fleksibilitasnya. Kata SQL pada MySQL merupakan singkatan dari *Structure Query Language*. SQL merupakan bahasa standar yang digunakan untuk mengakses *database* dan ditetapkan oleh ANSI/ISO SQL Standard.

2.9 Pengertian Java

Menurut Suarga (2007:2) Java adalah nama salah satu bahasa pemrograman komputer yang berorientasi objek, diciptakan oleh satu tim dari perusahaan Sun Microsystem, perusahaan workstation UNIX (*Sparc*) yang cukup terkenal. JAVA diciptakan berdasarkan C++, dengan tujuan *platform independent* (dapat dijalankan pada berbagai jenis hardware tanpa kompilasi ulang), dengan slogan *Write once Run Anywhere* (WORA). Dibanding bahasa C++, JAVA pada hakikatnya lebih sederhana dan memakai objek secara murni.

Bahasa java merupakan bahasa yang *multi platform* bahasa yang dapat digunakan di semua *platform* atau bekerja di semua sistem operasi.

3. METODE PENELITIAN

Metode yang digunakan dalam penelitian ini mengacu kepada SDLC (*System Development Life Cycle*) (Rosa, 2013), dengan tahapan:

1. Inisiasi (*initiation*)

Pada tahap ini diawali dengan pembuatan proposal

2. Pengembangan konsep sistem (*system concept development*)

Mendefinisikan lingkup konsep termasuk dokumen lingkup sistem, analisis manfaat biaya, manajemen rencana, dan pembelajaran kemudahan sistem.

3. Perencanaan (*planning*)

Mengembangkan rencana manajemen proyek dan dokumen perencanaan lainnya. Menyediakan dasar untuk mendapatkan sumber daya (*resource*) yang dibutuhkan untuk memperoleh solusi.

4. Analisis kebutuhan (*requirements analysis*)

Menganalisis kebutuhan pemakai sistem perangkat lunak (*user*) dan mengembangkan kebutuhan *user*. Membuat dokumen kebutuhan fungsional serta pembuatan gambaran sistem, seperti membuat use case diagram, activity diagram, sequence diagram dan class diagram, serta perancangan desain Entity Relationship Diagram (ER-Diagram).

5. Desain (*design*)

Mentransformasikan kebutuhan detail menjadi kebutuhan yang sudah lengkap, dokumen desain sistem fokus pada bagaimana dapat memenuhi fungsi-fungsi yang dibutuhkan.

6. Pengembangan (*development*)

Mengonversi desain ke sistem informasi yang lengkap termasuk bagaimana memperoleh dan melakukan instalasi lingkungan sistem yang dibutuhkan; membuat basis data dan mempersiapkan prosedur kasus pengujian; mempersiapkan berkas atau *file* pengujian, pengkodean, pengkompilasi, memperbaiki dan membersihkan program; peninjauan pengujian.

7. Integrasi dan pengujian (*integration and test*)

Mendemonstrasikan sistem perangkat lunak bahwa telah memenuhi kebutuhan yang dispesifikasikan pada dokumen kebutuhan fungsional. Dengan diarahkan oleh staff penjamin kualitas (*quality assurance*) dan *user*. Menghasilkan laporan analisis pengujian.

8. Implementasi (*implementation*)

Termasuk pada persiapan implementasi, implementasi perangkat lunak pada lingkungan produksi (lingkungan pada *user*) dan menjalankan resolusi dari permasalahan yang teridentifikasi dari fase integrasi dan pengujian.

9. Operasi dan pemeliharaan (*operations and maintenance*)

Mendeskripsikan pekerjaan untuk mengoperasikan dan memelihara sistem informasi pada lingkungan produksi (lingkungan pada *user*), termasuk implementasi akhir dan masuk pada proses peninjauan.

10. Dokumentasi dan Penyusunan Laporan

Penyusunan laporan merupakan tahapan akhir, namun setiap pelaksanaan penelitian selalu didokumentasikan, dengan maksud bisa berguna pada pengembangan selanjutnya.

4. HASIL DAN PEMBAHASAN

4.1 Pemodelan (Desain Sistem)

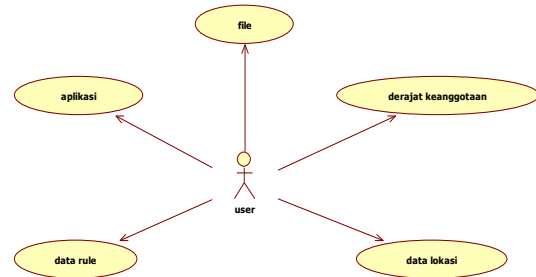
A. UML Sistem yang Diusulkan

UML (*Unified Modelling Language*) 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. A. S Rosa dan Salahuddin, M (2013:133).

1. Use Case

Diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem

informasi yang akan dibuat. Secara kasar *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

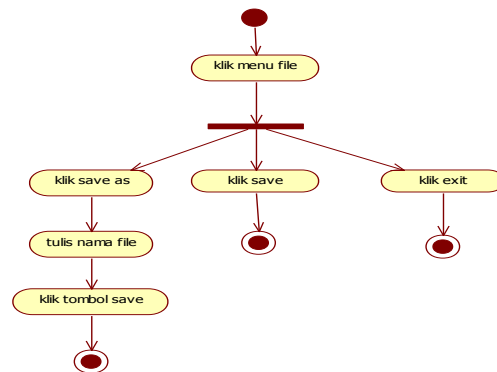


Gambar 3.1 Use Case SPK penentuan lokasi terbaik

2. Activity Diagram

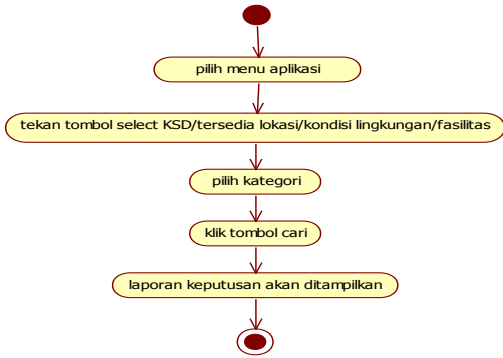
Diagram *activity* adalah *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

a. Activity Diagram file



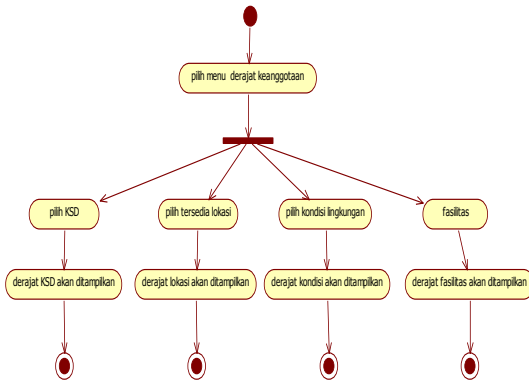
Gambar 3.2 Activity Diagram file

b. Activity Diagram aplikasi



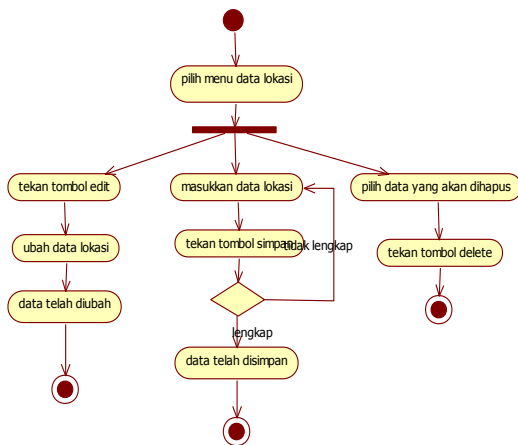
Gambar 3.3 Activity Diagram aplikasi

c. Activity Diagram Derajat Keanggotaan



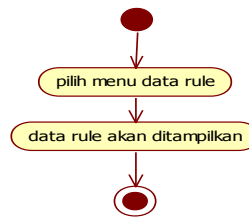
Gambar 3.4 Activity Diagram Derajat Keanggotaan

d. Activity Diagram Data Lokasi



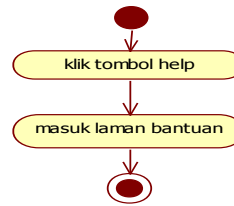
Gambar 3.5 Activity Diagram Data Lokasi

e. Activity Diagram Data Rule



Gambar 3.6 Activity Diagram Data Rule

f. Activity Diagram Help

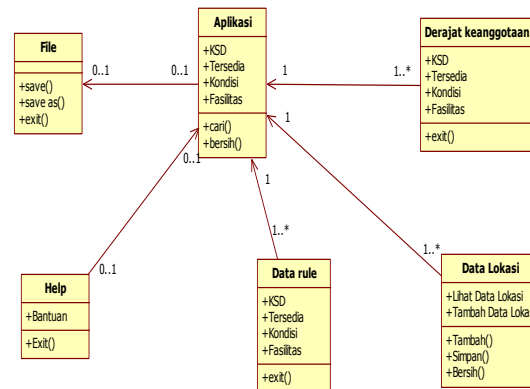


Gambar 3.7 Activity Diagram Help

3. Class Diagram

Diagram *Class* atau *Class* diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

seperti pada gambar 3.8 :



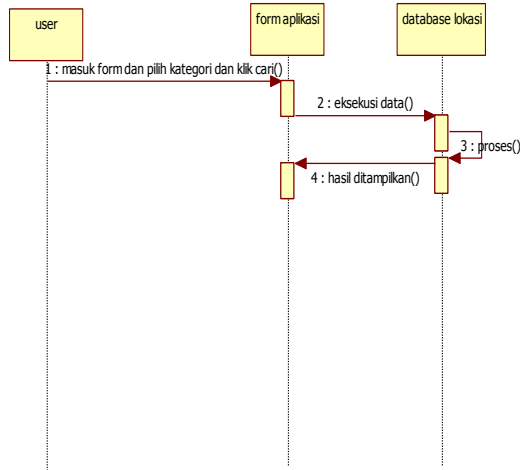
Gambar 3.8 Class Diagram Spk

4. Sequence Diagram

Diagram *Sequence* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan

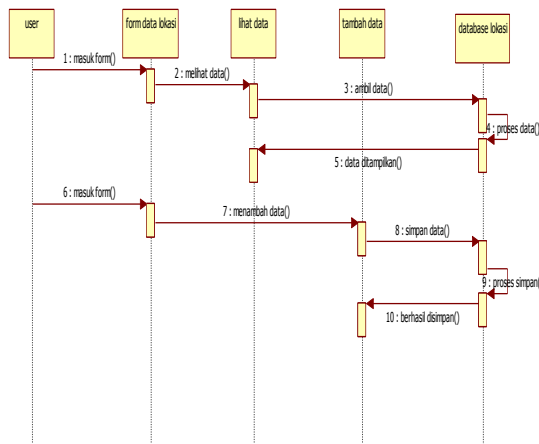
message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram *sequence* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

a. Diagram Sequence Aplikasi



Gambar 3.9 Diagram Sequence Aplikasi

b. Diagram Sequence Data Lokasi



Gambar 3.10 Diagram Sequence Data Lokasi

Program yang telah selesai dibangun kemudian diuji untuk mengetahui apakah program tersebut telah berjalan sesuai yang diinginkan maupun untuk menanggulangi kemungkinan terjadinya *error* program saat

program diimplementasikan. Pengujian dilakukan dengan memakai metode *Blackbox* yaitu menguji masukkan yang diberikan oleh user baik masukan yang sesuai aturan maupun yang tidak sesuai.

B. Implementasi Sistem

Implementasi sistem merupakan tahapan yang dilakukan setelah melakukan tahap perancangan sistem yaitu menjelaskan mengenai pembuatan sistem serta tampilan dari aplikasi yang telah dibuat, yang sesuai dengan analisis dan perancangan sebelumnya. Setelah tahap implementasi dilakukan maka dibutuhkan sebuah pengujian sistem untuk membuktikan bahwa aplikasi dapat berjalan sesuai dengan yang diharapkan.

C. Instalasi Program

Dalam mengimplementasikan aplikasi ini dibutuhkan 3 buah komponen penting, yaitu

1. Hardware (Perangkat Keras)

Hardware adalah komponen yang terdapat pada sebuah komputer yang diperlukan agar komputer tersebut dapat beroperasi. Dalam mengimplementasi sistem ini perangkat keras yang digunakan yaitu *notebook*. Adapun spesifikasi komputer dari perangkat keras tersebut adalah :

- a) RAM 2048 Mb.
- b) Intel atom 1,5 Ghz
- c) *Harddisk* 250 Gb.

2. Software (Perangkat Lunak)

Pengertian *software* menurut Rosa dan Shalahuddin (2013:2) adalah program komputer yang terasosiasi dengan dokumen perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan. Adapun *software* yang dibutuhkan dalam mengimplementasi sistem ini adalah sebagai berikut :

- a) Sistem operasi Windows 7
- b) XAMPP 3.1.0.3.1.0

3. Brainware (Pengguna)

Brainware diperlukan untuk mengoperasikan sistem informasi penjualan dan pembelian. Sistem ini memerlukan

brainware agar mampu berfungsi secara maksimal. Selain itu *brainware* juga berfungsi untuk melakukan perawatan terhadap *hardware* maupun *Software*.

D. Rencana Pengujian

Rencana pengujian yang akan dilakukan dengan pengujian sistem secara *alpha* dan *beta*. Pengujian sistem informasi penjualan dan pembelian barang menggunakan data uji berdasarkan data yang telah didapat dari Admin dan ketua koperasi. Rencana pengujian selengkapnya terlihat pada Tabel 4.1:

Tabel 4.1 Rencana Pengujian Aplikasi

No	Item Uji	Detail Pengujian	Jenis Pengujian
1	Input data lokasi baru	Input data lokasi	<i>BlackBox</i>
2	Ubah data lokasi	Ubah data lokasi	<i>Black Box</i>
3	Hapus data lokasi	Menghapus data lokasi	<i>Black Box</i>
4	Perhitungan aplikasi	Perhitungan aplikasi	<i>Black Box</i>

E. Kegiatan Pengujian

Kegiatan pengujian, jenis pengujian yang dilakukan adalah pengujian alpha. Pengujian alpha adalah pengujian yang dilakukan oleh *programmer* atau sistem analisis sebelum sistem diimplementasikan. Berdasarkan rencana pengujian yang telah disusun, maka dapat dilakukan pengujian sebagai berikut:

1. Input Data Lokasi

Tabel 4.2 Pengujian Tambah Data Lokasi

Kasus dan Hasil Uji					
No	Data Masukkan		Yang Diharapkan	Hasil Pengamatan	Kesimpulan
	Atribut	Masukkan			
1.	Nama Lokasi	Jalan Semeru	Penyimpanan berhasil	Penyimpanan berhasil	Sukses
	Kedekatan SD	0.5			
	Tersedia	11			
	Kondisi	14			
	Fasilitas	16			
2.	Nama Lokasi	Jalan Semeru	Penyimpanan gagal karena Kedekatan SD belum diisi	Penyimpanan gagal karena Kedekatan SD belum diisi	Sukses
	Kedekatan SD	<kosong>			
	Tersedia	11			
	Kondisi	14			
	Fasilitas	16			
3.	Nama Lokasi	Jalan Semeru	Penyimpanan gagal karena kondisi belum di isi	Penyimpanan gagal karena kondisi belum di isi	Sukses
	Kedekatan SD	0.5			
	Tersedia	11			
	Kondisi	<Kosong>			
	Fasilitas	16			
4.	Nama Lokasi	Jalan Semeru	Penyimpanan gagal karena Kedekatan SD dan Tersedia belum diisi	Penyimpanan gagal karena Kedekatan SD dan Tersedia belum diisi	Sukses
	Kedekatan SD	<kosong>			
	Tersedia	<kosong>			
	Kondisi	14			
	Fasilitas	16			
5	Nama Lokasi	Jalan Semeru	Penyimpanan gagal karena fasilitas kosong	Penyimpanan gagal karena fasilitas kosong	Sukses
	Kedekatan SD	0.5			
	Tersedia	11			
	Kondisi	14			
	Fasilitas	<kosong>			

2. Ubah Data Lokasi

Tabel 4.3 Pengujian Ubah Data Lokasi

Kasus dan Hasil Uji					
No	Data Masukkan		Yang Diharapkan	Hasil Pengamatan	Kesimpulan
	Atribut	Masukkan			
1.	Nama Lokasi	Jalan Semeru	Penggubahan berhasil	Penggubahan berhasil	Sukses
	Kedekatan SD	0.5			
	Tersedia	11			
	Kondisi	14			
	Fasilitas	16			
2.	Nama Lokasi	Jalan Semeru	Penggubahan gagal karena Kedekatan SD belum diisi	Penggubahan gagal karena Kedekatan SD belum diisi	Sukses
	Kedekatan SD	<kosong>			
	Tersedia	11			
	Kondisi	14			
	Fasilitas	16			
3.	Nama Lokasi	Jalan Semeru	Penyimpanan gagal karena kondisi belum di isi	Penggubahan gagal karena kondisi belum di isi	Sukses
	Kedekatan SD	0.5			
	Tersedia	11			
	Kondisi	<Kosong>			
	Fasilitas	16			
4.	Nama Lokasi	Jalan Semeru	Penggubahan gagal karena Tersedia dan kondisi belum diisi	Penggubahan gagal karena Tersedia dan kondisi belum diisi	Sukses
	Kedekatan SD	0.5			
	Tersedia	<kosong>			
	Kondisi	<kosong>			
	Fasilitas	16			
5	Nama Lokasi	Jalan Semeru	Penggubahan gagal karena fasilitas kosong	Penggubahan gagal karena fasilitas kosong	Sukses
	Kedekatan SD	0.5			
	Tersedia	11			
	Kondisi	14			
	Fasilitas	<kosong>			

3. Pengujian Delete Data Lokasi

Tabel 4.4 Pengujian Hapus Data Lokasi

Kasus dan Hasil Uji					
No	Data Masukkan		Yang Diharapkan	Hasil Pengamatan	Kesimpulan
	Atribut	Masukkan			
1.	Nama Lokasi	Jalan Semeru	Data terhapus dari database	Penggubahan data berhasil	Sukses
	Kedekatan SD	0.5			
	Tersedia	11			
	Kondisi	14			
	Fasilitas	16			

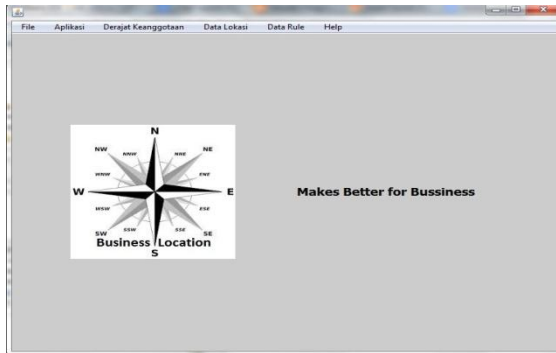
F. Hasil Pengujian

Setiap halaman menu dalam *program* akan diuji, dengan cara memasukkan data pada kolom *inputan* yang ada. Kolom *inputan* akan diisi dengan data-data yang berbeda, baik berbeda karakter maupun nilai, *inputan* yang dimasukkan adalah *input* yang mungkin masuk pada saat implementasi program. Hasil yang keluar akan dicocokkan dengan hasil yang diharapkan.

Berdasarkan pengujian sistem yang telah dilakukan, berikut adalah tampilan dari hasil pengujian:

1. Form Halaman Utama

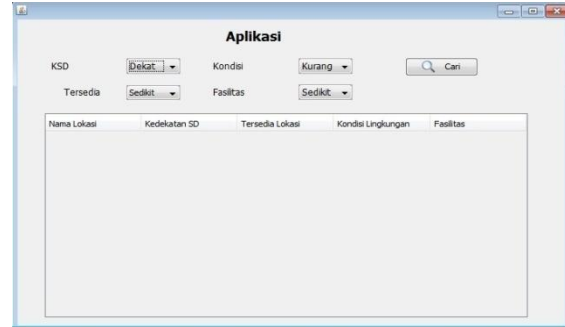
Sebelum admin dan kasir mengolah data penjualan dan pembelian, maka Admin dan Kasir harus login terlebih dahulu. gambar 4.1 form login admin jika berhasil maka akan keluar peringatan “Login Sukses”. Login sukses karena *password* dan *username* sesuai begitu juga dengan login kasir.



Gambar 4.1 Form Halaman Utama

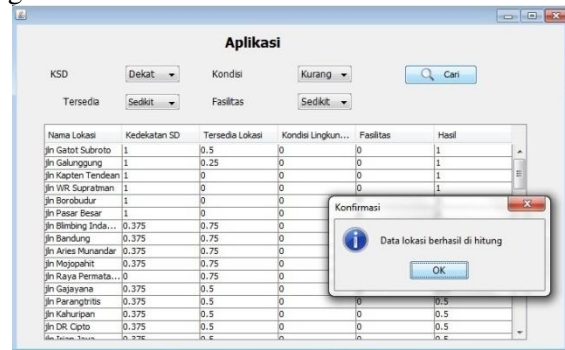
2. Form Aplikasi

Form aplikasi berfungsi untuk menunjukkan data lokasi yang paling terbaik, strategis dan berpotensi bagi pelaku usaha bakso. Untuk menunjukkan hasilnya harus pilih kategori yang diinginkan kemudian klik tombol cari dan data lokasi yang dicari akan muncul dalam kolom tabel. Form aplikasi seperti gambar 4.2.



Gambar 4.2 Form Aplikasi yang masih kosong

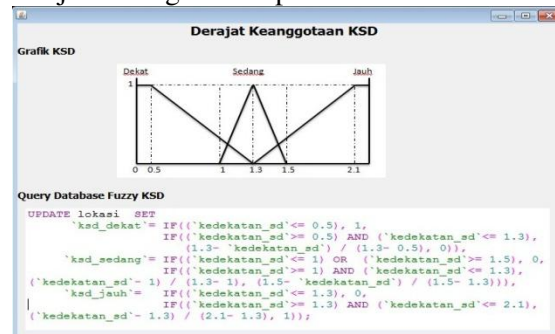
Data yang sudah terinput, akan masuk masuk pada tabel item baru. Seperti gambar 4.3.



Gambar 4.3 Form Aplikasi Yang Sudah dieksekusi

3. Form Derajat Keanggotaan

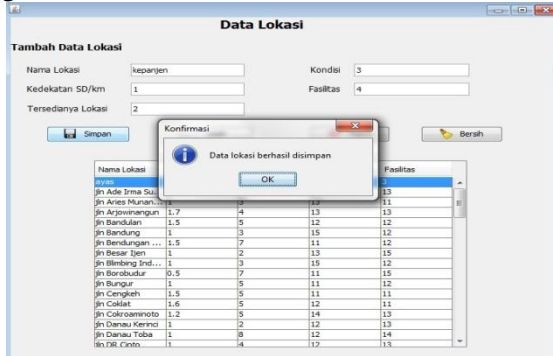
Form derajat keanggotaan berisi derajat keanggotaan tiap variabel untuk menunjukkan pemetaan titik input data kedalam nilai keanggotaannya yang memiliki interval antara 0 dan 1. Form derajat keanggotaan seperti Gambar 4.4.



Gambar 4.4 Form Derajat Keanggotaan

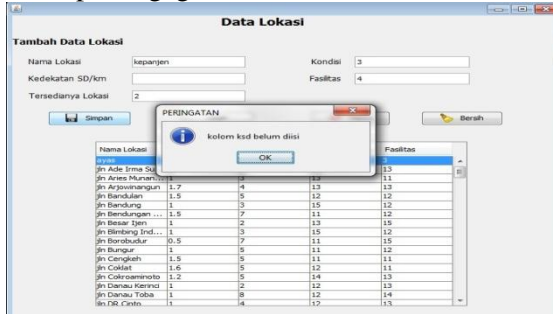
4. Form Data Lokasi

Form data lokasi berfungsi untuk mengisi data lokasi yang disimpan dalam database. Jika data yang diinputkan semakin banyak maka hasilnya akan lebih baik. Form data lokasi ketika input data berhasil seperti gambar 4.5.



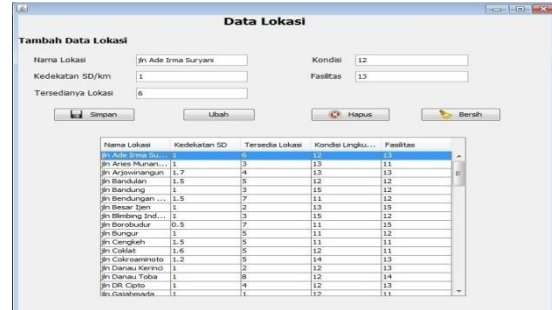
Gambar 4.5 Form ketika Data Lokasi tersimpan

Jika input data lokasi baru gagal tersimpan maka akan tampil seperti gambar 4.6 inputan gagal karena ksd belum terisi.



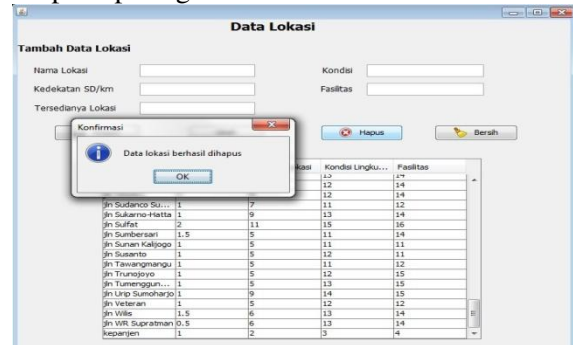
Gambar 4.6 Form ketika Data Lokasi gagal tersimpan

Jika ingin mengubah data yang ada di dalam tabel maka form yang akan ditampilkan seperti gambar 4.7.



Gambar 4.7 Form ketika mengubah data lokasi

Jika data lokasi terhapus maka akan tampil seperti gambar 4.8



Gambar 4.8 Form ketika Data Lokasi terhapus

4. Form Data Rule

Form data rule merupakan form yang berisi aturan-aturan dalam penghitungan setiap derajat keanggotaan yang berupa kemungkinan yang akan dipakai. Form data rule seperti gambar 4.9 .

ksd	tersedia	kondisi	fasilitas
dekat	sedikit	kurang	sedikit
dekat	sedikit	kurang	sedang
dekat	sedikit	kurang	lengkap
dekat	sedikit	sedang	sedikit
dekat	sedikit	sedang	sedang
dekat	sedikit	sedang	lengkap
dekat	sedikit	baik	sedikit
dekat	sedikit	baik	sedang
dekat	sedikit	baik	lengkap
dekat	sedang	kurang	sedikit
dekat	sedang	kurang	sedang
dekat	sedang	kurang	lengkap
dekat	sedang	sedang	sedikit
dekat	sedang	sedang	sedang
dekat	sedang	sedang	lengkap
dekat	sedang	baik	sedikit
dekat	sedang	baik	sedang

Gambar 4.9 Form Data Rule

5. Form Help

Form help berisi bantuan tentang penggunaan aplikasi ini. Form help seperti gambar 4.10.



Gambar 4.10 Form Help

5.KESIMPULAN

Dalam pemilihan sebuah lokasi untuk memulai usaha harus tepat karena pemilihan lokasi usaha merupakan salah satu poin penting yang harus diperhatikan demi kelanjutan usaha yang akan dibangun. Untuk mendapatkan lokasi usaha yang tepat, dibuatkanlah sebuah Sistem Penunjang Keputusan Penentuan Lokasi Terbaik Jual Bakso di Kota Malang Dengan Metode Basisdata Fuzzy agar pencarian lokasi usaha lebih tepat.

Berdasarkan hasil pengujian disimpulkan bahwa Aplikasi Sistem Penunjang Keputusan memberikan informasi penentuan lokasi terbaik dengan menampilkan keputusan rekomendasi pilihan yang memudahkan pengusaha dalam menentukan lokasi yang terbaik dan tepat untuk menjalankan usaha.

6.REFERENSI

- Agustin, Risa. tt. *Kamus Lengkap Bahasa Indonesia*. Surabaya. Serba Jaya.
- Firdaus. 2007. *7 Jam Belajar Interaktif PHP & MySQL dengan Dreamweaver*. Yogyakarta. Maxikom.
- Jogiyanto HM. 2009. *Sistem Teknologi Informasi*. Andi Offset. Yogyakarta.

Kadir, Abdul. 2014. *Pengenalan Sistem Informasi*. Andi Offset. Yogyakarta.

Kusumadewi, S, dan Purnomo, H, 2010. *Aplikasi Logika Fuzzy Sistem Pendukung Keputusan*. Graha Ilmu. Yogyakarta.

Khotler, Philip, dan Armstrong, Gary, 2008. *Prinsip-prinsip Pemasaran*. Erlangga. Jakarta.

Ladjamudin, Al-Bahra Bin. 2005. *Analisis dan Desain Sistem Informasi*. Edisi Pertama 2005. Yogyakarta: Penerbit Graha Ilmu.

Rosa dan Shalahuddin. 2013. *Rekayasa Perangkat Lunak*. Bandung: Informatika.

Soeherman, Bonnie dan Pinontoan, Marion. 2008. *Designing information system*. Jakarta: PT. Elex Media Komputindo

Suarga. 2009. *Dasar Pemrograman Komputer Dalam Bahasa Java*. Andi Offset. Yogyakarta.