

**PERANCANGAN SISTEM PENDUKUNG KEPUTUSAN
PENERIMAAN KARYAWAN MENGGUNAKAN
METODE FUZZY LOGIC**

ARIEF PEBRIANTO

Sekolah Tinggi Manajemen Informatika & Komputer Indonesia Mandiri,
Bandung, Indonesia

ABSTRACT

This study examines the solution to the problem of taking decisions for new employee recruitment. Employees are an important element in a company in improving the quality of a company, so the quality of its employees must also have an appropriate standard, especially in hiring new employees. However, the standard of eligibility is sometimes still vague, so it requires a more precise and quality decision. Decision support systems mean to assist in decision making not to replace the duties of a manager so that with the support of data, accurate information. Problems in decision-making that are carried out in the recruitment of new employees that make the company lose out in making decisions for hiring new employees that are not right. The solution to the problem is to design and build a decision support system for hiring new employees. The expected results in this study are to facilitate and minimize inappropriate decision making in hiring new employees. This study uses the fuzzy logic method for its decision support system and is implemented using the NET Framework language with the development software is a visual studio..

Keywords: *Fuzzy Logic, Decision Support System, Employee Acceptance, NET.Framework*

ABSTRAK

Penelitian ini mengkaji tentang solusi dari permasalahan dalam mengambil keputusan untuk penerimaan karyawan baru. Karyawan merupakan unsur penting dalam suatu perusahaan dalam meningkatkan kualitas suatu perusahaan maka kualitas karyawan nya juga harus memiliki standar kelayakan terutama dalam penerimaan karyawan baru. Akan tetapi, standar kelayakan tersebut terkadang masih samar sehingga membutuhkan keputusan yang lebih tepat dan berkualitas. Sistem pendukung keputusan bermaksud untuk membantu dalam pengambilan keputusan bukan untuk menggantikan tugas seorang manajer sehingga dengan dukungan data, informasi yang akurat. Permasalahan dalam pengambilan

keputusan yang dilakukan dalam penerimaan karyawan baru yang membuat perusahaan rugi dalam mengambil keputusan penerimaan karyawan baru yang tidak tepat. Solusi dalam permasalahan adalah merancang dan membangun sistem pendukung keputusan penerimaan karyawan baru. Hasil yang di diharapkan dalam penelitian ini yaitu untuk memudahkan dan meminialisir pengambilan keputusan yang kurang tepat dalam penerimaan karyawan baru. Penelitian ini menggunakan metode fuzzy logic untuk sistem pendukung keputusannya dan diimplementasikan menggunakan Bahasa NET.Farmework dengan perangkat lunak pengembangannya adalah visual studio dan basisdata yang digunakan yaitu MySql.

Kata Kunci : *Fuzzy Logic, Sistem Pendukung Keputusan, Penerimaan Karyawan, NET.Framework.*

1. PENDAHULUAN

Karyawan merupakan unsur penting dalam suatu perusahaan dalam meningkatkan kualitas suatu perusahaan maka kualitas karyawan nya juga harus memiliki standar kelayakan terutama dalam penerimaan karyawan baru. Akan tetapi, standar kelayakan tersebut terkadang masih samar sehingga membutuhkan keputusan yang lebih akurat dan berkualitas (Eliasta, 2015).

Sistem pendukung keputusan bermaksud untuk membantu dalam pengambilan keputusan bukan untuk menggantikan tugas seorang manajer sehingga dengan dukungan data, informasi yang akurat diharapkan dapat membuat keputusan yang lebih akurat dan berkualitas.

Membuat sebuah keputusan sudah menjadi hal yang lumrah dilakukan dalam lingkungan perusahaan untuk penerimaan karyawan dan beberapa perusahaan masih membuat keputusan penerimaan karyawan tanpa adanya sistem pendukung keputusan yang membuat keputusan kurang baik dan tepat sehingga dengan ini dapat memanfaatkan metode Fuzzy Logic untuk membuat sistem pendukung keputusan yang lebih akurat dan berkualitas.

Ada pun input yang dapat digunakan dalam sistem pendukung keputusan metode fuzzy logic. Berdasarkan standar kelayakan perusahaan dalam penerimaan

karyawan dimana standar tersebut meliputi Delapan komponen utama, yaitu, Kepribadian, Nilai Tes Matematika, Nilai Tes Pengetahuan Umum, Nilai Tes Kepribadian, Nilai Tes Peraktik (Magang)dan Nilai Mempunyai SIM C, Nilai

Memiliki Motor, Nilai Pengalaman Kerja, dan Nilai Umur. kemudian Hasil dari Wawancara yang masing – masing komponen memiliki poin - poin lainnya. Setelah di dapatkan layak atau tidaknya calon karyawan tersebut untuk diterima oleh perusahaan.

Pada penelitian ini metode fuzzy logic akan diterapkan untuk sistem pendukung keputusan penerimaan karyawan di perusahaan sehingga dapat membantu mengambil keputusan dalam penerimaan karyawan yang lebih akurat dan berkualitas.

1.1. Identifikasi Masalah

Berdasarkan latar belakang, maka identifikasi masalahnya sebagai berikut:

- 1 Adanya masalah dalam pengambilan keputusan yang tidak akurat di perusahaan.
- 2 Pengambilan keputusan yang dilakukan dalam penerimaan karyawan baru yang membuat perusahaan rugi dalam mengambil keputusan penerimaan karyawan baru

1.2. Tujuan Penelitian

Maksud dan Tujuan pembuatan Tugas Akhir ini adalah merancang dan mengimplementasikan aplikasi sistem pendukung keputusan penerimaan karyawan dengan metode fuzzy logic. Adanya tugas akhir ini diharapkan dapat bermanfaat bagi perusahaan :

1. Membantu memudahkan pengambilan keputusan dalam menerima karyawan baru
2. Meminimalisir pengambilan keputusan yang kurang tepat dalam penerimaan kariyawan baru.

2. METODE PENELITIAN

2.1 . Sitem Pendukung Keputusan

Sistem pendukung keputusan di perkenalkan oleh G. Anthony Gorry dan Michael S. Scott Morton. Keduanya adalah professor dari *Massachusetts Institute of Technology* (MIT) menulis artikel dalam jurnal yang berjudul “ *A Framework for Managent Information System* “. Mereka mengembangkan kerangka pemikiran tentang pemanfaatan aplikasi *computer* pada proses pengambilan keputusan bagi

level manajemen. Berdasarkan kerangka ini dapat didefinisikan bahwa sistem pendukung keputusan ini berkaitan erat dengan sistem informasi atau model analisis yang dirancang untuk membantu para pengambil keputusan dan para profesional agar mendapatkan informasi yang akurat (Diana,2018:18).

DSS sebenarnya merupakan implementasi teori – teori pengambilan keputusan yang telah diperkenalkan oleh ilmu – ilmu seperti *operation research* dan *management science*. Hanya bedanya adalah bahwa jika dahulu untuk mencari penyelesaian masalah yang dihadapi harus dilakukan perhitungan iterasi secara manual (biasanya untuk mencari nilai minimum, maksimum, atau optimum). Saat ini computer telah menawarkan kemampuannya untuk menyelesaikan persoalan yang sama dalam waktu relatif singkat. Sistem pendukung keputusan memberikan dukungan kepada seorang manajer atau kepada sekelompok manajer dalam memecahkan masalah semi terstruktur dengan memberi informasi atau saran mengenai keputusan tertentu, informasi tersebut dapat diberikan dalam bentuk laporan berkala, laporan khusus maupun model matematis. Model tersebut juga mempunyai kemampuan untuk memberikan saran pada tingkat yang bervariasi. Sistem pendukung keputusan ini merupakan pengembangan sistem informasi dalam pengambilan keputusan, yang difokuskan pada dukungan kepada manajemen. Keberadaan sistem pendukung keputusan ini bukan untuk menggantikan tugas – tugas manajer, tetapi untuk menjadi sarana pendukung bagi mereka (Diana,2018:19).

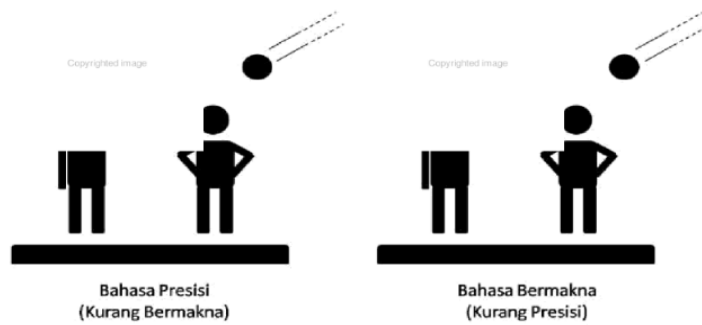
2.2. Fuzzy Logic

Dalam kamus *Oxford*, istilah *fuzzy* didefinisikan sebagai *blurred* (kabur atau remang-remang), *indistinct* (tidak jelas), *imprecisely defined* (didefinisikan secara tidak presisi), dan *confused* (membingungkan). Mendengar definisi-definisi istilah *fuzzy* tersebut, mereka yang belum pernah mendengar istilah “sistem *fuzzy*” bisa salah mengerti. Dalam teori *fuzzy logic*, kata *fuzzy* lebih dipandang sebagai sebuah *technical adjective*. Penggunaan istilah “sistem *fuzzy*” tidak dimaksudkan untuk mengacu pada sebuah sistem yang tidak jelas definisinya, cara kerjanya atau deskripsinya. Sebaliknya, yang dimaksud dengan sistem *fuzzy* adalah sebuah sistem yang dibangun dengan definisi, cara kerja, dan deskripsi yang jelas berdasar pada

teori *fuzzy logic*. Yang ingin ditekankan disini adalah bahwa meskipun sebuah fenomena yang akan dimodelkan dengan sistem *fuzzy* adalah bersifat *fuzzy*, sistem *fuzzy* yang dibangun untuk memodelkan fenomena tersebut mempunyai definisi cara kerja dan deskripsi yang jelas berdasar pada teori *fuzzy logic* (Agus, 2009:1).

Secara umum, *fuzzy logic* adalah sebuah metodologi “berhitung” dengan *variable* kata-kata (*linguistic variable*), sebagai pengganti berhitung dengan bilangan. Tentunya, kata-kata yang digunakan dalam *fuzzy logic* adalah tidak sepresisi dengan bilangan, namun pemakaian kata-kata jauh lebih dekat dengan intuisi manusia dimana manusia bisa langsung “merasakan” nilai dari *variable* kata-kata yang sudah dipakai sehari-hari. *fuzzy logic* memberi ruang dan bahkan mengeksploitasi toleransi terhadap ketidakpresisian. *Fuzzy logic* membutuhkan “ongkos” yang lebih murah dalam memecahkan suatu masalah-masalah yang bersifat *fuzzy*. *Fuzzy logic* telah menjadi area riset yang mengagumkan karena kemampuannya dalam menjembatani bahasa mesin yang serba presisi dengan bahasa manusia bisa yang cenderung tidak presisi, yaitu hanya menekankan pada makna atau arti (*significance*) (Agus, 2009:1)..

Dengan *fuzzy logic*, sistem kepakaran manusia bisa diimplementasikan ke dalam bahasa mesin dengan mudah dan efisien. Gambar 2.1. memberikan ilustrasi penggunaan bahasa presisi (dengan bilangan) dan bahasa kata-kata. Bahasa presisi yang diperlukan mesin sulit “dirasakan” oleh manusia (yaitu, kurang bermakna dari sudut pandang bahasa manusia), disamping itu deskripsinya bisa cukup Panjang. Sebaliknya *variable* kata-kata bisa *simple*, singkat, dan langsung bisa “dirasakan” oleh manusia, namun kurang presisi dari sudut pandang bahasa mesin. Disinilah peran dari sistem *fuzzy*, yaitu untuk menjembatani komunikasi yang lebih efektif dan efisien antara mesin dan manusia. Atau, bisa juga dibayangkan sistem *fuzzy* adalah sebuah mesin penerjemah bahasa manusia sehingga bisa dimengerti oleh mesin dan juga sebaliknya (Agus, 2009:2).



GAMBAR 2.1. Bahasa Presisi vs Bahasa Bermakna (Agus, 2009:2).

Fuzzy Logic diperkenalkan oleh Prof. Lotfi Zadeh pada tahun 1965. Merupakan metode yang mempunyai kemampuan untuk memproses variabel yang bersifat kabur atau yang tidak dapat dideskripsikan secara eksak/pasti seperti misalnya suhu, kebisingan, dan pencahayaan (Hermawanto, 2006:1).

Dalam *fuzzy logic* variabel yang bersifat kabur tersebut direpresentasikan sebagai sebuah himpunan yang anggotanya adalah suatu nilai crisp dan derajat keanggotaannya (*membership function*) dalam himpunan crisp. Proses-proses dalam *fuzzy logic* adalah fuzzifikasi, penalaran (*reasoning*), dan defuzzifikasi (Deny, 2006 :1):

1. Fuzzifikasi adalah merupakan proses untuk mendapatkan himpunan derajat keanggotaan dari sebuah nilai numerik masukan (*crisp*).
2. Penalaran adalah proses untuk mendapatkan aksi keluaran dari suatu kondisi input dengan mengikuti aturan-aturan (*IF-THEN Rules*) yang telah ditetapkan yang disebut sebagai *inference/reasoning*.

2.3. Karyawan/Pegawai

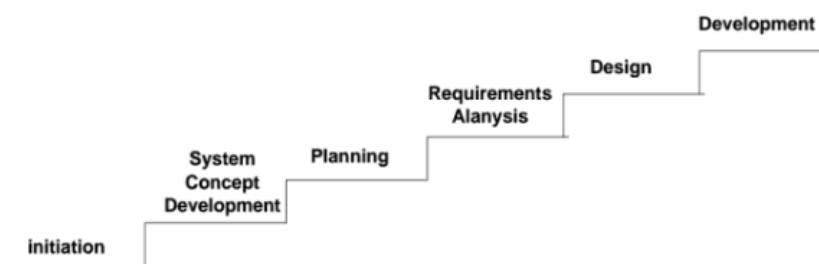
Definisi pegawai adalah orang pribadi yang bekerja pada pemberi kerja baik sebagai pegawai tetap atau pegawai tidak tetap/tenaga kerja lepas berdasarkan perjanjian atau kesepakatan kerja baik secara tertulis maupun tidak tertulis, untuk melaksanakan suatu pekerjaan dalam jabatan atau kegiatan tertentu dengan memperoleh imbalan yang dibayarkan berdasarkan periode tertentu, penyelesaian pekerjaan atau ketentuan lain yang ditetapkan pemberi kerja, termasuk orang pribadi yang melakukan pekerjaan dalam jabatan negeri atau badan usaha milik negara atau badan milik daerah (Prditya, 2016:6).

Para pegawai adalah aset organisasi yang paling berharga. Pengetahuan dan keahlian mereka mempengaruhi kualitas barang dan jasa yang diberikan ke para pelanggan. Di dalam perusahaan manufaktur, dimana tenaga kerja biasanya hanya mencerminkan sebagian dari total biaya langsung, para pegawai adalah penggerak biaya dalam hal kualitas kerja mereka mempengaruhi baik produktivitas keseluruhan maupun tingkat kecacatan produk (Prditya, 2016:6).

2.4. System Development Life Cycle (SDLC) Waterfal

Metode perancangan atau paradigma perangkat lunak yang di gunakan adalah metode *System Development Life Cycle (SDLC)* atau sering disebut metode waterfall.

(Sri Mulyani, 2016:24) SDLC adalah Sebuah metode yang digunakan untuk mengembangkan sebuah sistem . SDLC adalah sebuah proses logika yang digunakan oleh seorang system analist untuk mengembangkan sebuah sistem informasi yang melibatkan requirements, validation, training dan pemilik sistem. SDLC identik dengan teknik pengembangan sistem *warterfall* karena tahapannya menurun dari atas kebawah. Berikut Tahapan dari SDLC (Sumber Sri Mulyani, 2016:24) :



Gambar 2.2. Tahapan Pengembangan Sistem SDLC Waterfall

(Sri Mulyani,2016:25).

1. Initiation - Planing

Merupakan tahap dimana sistem digambarkan secara global beserta tujuan yang akan direncanakan terhadap sistem yang akan dikembangkan. Tahap ini identic dengan tahap analisis.

2. *Requirement Gathering and Analysis*

Pada tahap ini analis mencoba untuk menguraikan permasalahan sistem untuk menggambarkan situasi yang sedang berjalan, kemudian pada tahap ini juga analis mencoba mendesain sebuah solusi yang diberikan kepada user.

3. *Design*

Pada tahap Ini solusi – solusi yang sudah digambarkan secara global pada tahap *requirement gathering and analysis* di uraikan secara detail baik dalam bentuk *diagram, layouts, Busines rules*, dan dokumentasi – dokumentasi lain yang dibutuhkan

4. *Development*

Pada tahap ini sistem mulai di bangun atau dikembangkan. Dan juga Pada tahap ini sistem yang sudah di bangun atau dikembangkan dan dicoba untuk mendapatkan hasil akhir.

2.5. *Flowchart*

Flowchart adalah penyajian yang sistematis tentang proses dan logika dari kegiatan penanganan informasi atau penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. *Flowchart* menolong analis dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian (Anarku,2009:1).

System flowchart adalah urutan proses dalam system dengan menunjukkan alat media *input, output* serta jenis media penyimpanan dalam proses pengolahan data. Program *flowchart* adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program (Anarku,2009:1).







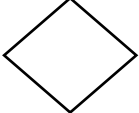

2.5.1. **Pedoman-Pedoman Dalam Membuat *Flowchart***

Jika seorang analis dan programmer akan membuat *flowchart*, ada beberapa petunjuk yang harus diperhatikan, seperti (Sumber Anarku,2009:1). :

1. *Flowchart* digambarkan dari halaman atas ke bawah dan dari kiri ke kanan.

2. Aktivitas yang digambarkan harus didefinisikan secara hati-hati dan definisi ini harus dapat dimengerti oleh pembacanya.
3. Kapan aktivitas dimulai dan berakhir harus ditentukan secara jelas.
4. Setiap langkah dari aktivitas harus diuraikan dengan menggunakan deskripsi katakerja, misalkan Melakukan penggandaan diri.
5. Setiap langkah dari aktivitas harus berada pada urutan yang benar.
6. Lingkup dan *range* dari aktifitas yang sedang digambarkan harus ditelusuri dengan hati-hati. Percabangan-percabangan yang memotong aktivitas yang sedang digambarkan tidak perlu digambarkan pada *flowchart* yang sama. Simbol konektor harus digunakan dan percabangannya diletakan pada halaman yang terpisah atau hilangkan seluruhnya bila percabangannya tidak berkaitan dengan sistem.
7. Gunakan simbol-simbol *flowchart* yang standar.

TABEL 2.1. Notasi atau Simbol-Simbol Flowchart

Simbol	Nama	Fungsi
	Terminator	Permulaan atau akhir program
	Garis Alir (<i>Flow Line</i>)	Arah alir program
	<i>Preparation</i>	Proses inisialisasi
	Proses	Proses pengolahan data
	<i>Input atau Output</i>	Proses input atau output data
	Sub Program (<i>Predifined Process</i>)	Proses menjalankan sub program
	<i>Decision</i>	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untu langkah selanjutnya
	<i>On Page Connector</i>	Penghubung bagian-bagian flowchart yang berada satu

		halaman
	<i>Off Page Connector</i>	Penghubung bagian-bagian flowchart yang halaman berada
	<i>Storage</i>	<i>Input atau Output</i> media penyimpanan
	<i>Display Output</i>	Penapilan <i>Output</i> dilayar terminal
	<i>Printer</i>	Menggambar suatu Dokumen atau kegiatan mencetak suatu informasi dengan mesin



2.6. Data Flow Diagram (DFD)

Definisi *Data flow diagram* adalah sebuah Teknik grafis yang menggambarkan desain informasi yang diaplikasikan pada saat data bergerak dari *input* menjadi *output*. *Data flow diagram* dapat digunakan untuk menyajikan sebuah system atau perangkat lunak pada setiap tingkat abstraksi. *Data flow diagram* memberikan suatu mekanisme bagi pemodelan fungsional dan pemodelan aliran informasi (Al Fatta, 2009:32).

Dalam Menggambar *data flow diagram*, dikenal empat simbol dasar yang digunakan untuk memetakan gerakan diagram aliran data adalah sebagai berikut (Sumber Al Fatta, 2009:32):

TABEL 2.2. Simbol-simbol *Data flow diagram* (Sumber Maniah, Dini, 2017:46).

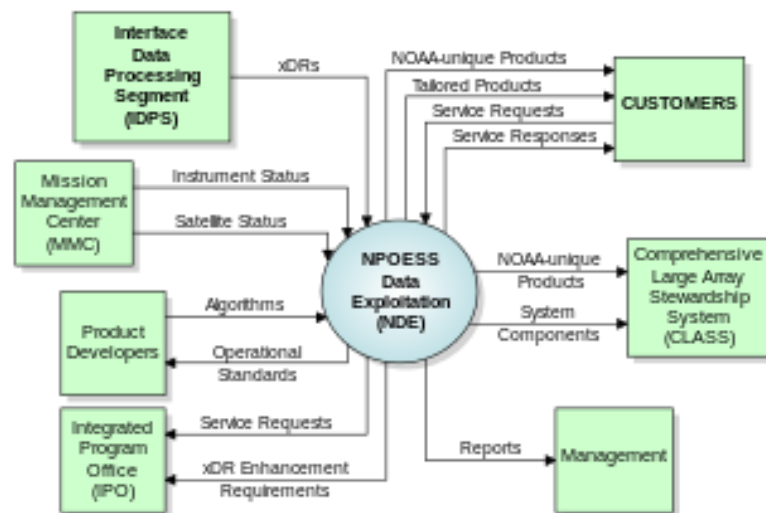
Simbol	Nama Simbol	Keterangan
	<i>External Entity</i> (Entitas)	Menyatakan <i>external</i> entitas sistem (sumber atau tujuan data yang digambarkan). Satu entitas dapat disimbolkan dengan satu notasi.
	<i>Data Flow</i> (Arus Data)	Arus data merupakan arus data yang bergerak dalam sistem. Sebuah arus dapat terdiri dari banyak potongan data

		yang dihasilkan dan satu simbol dapat disimbokan dengan arah panah megarah ke tujuan.
 <p>Gane & sarson atau Yourdon</p>	Proses	Simbol ini digunakan untuk menunjukan proses adanya tranformasi. Arus data yang keluar dalam suatu proses selalu diberi label yang berbeda dari arus data masuk.
 <p>Yourdon data store 1 data store Gane & sarson</p>	Penyimpanan Data	Penyimpanan data mewakili tempat lokasi data, seperti file basisdata.

Adapun tahapan *data flow diagram* dari sistem presensi karyawan berbasis pengenalan wajah disajikan sebagai berikut :

1. Diagram Konteks atau Diagram *Level 0*

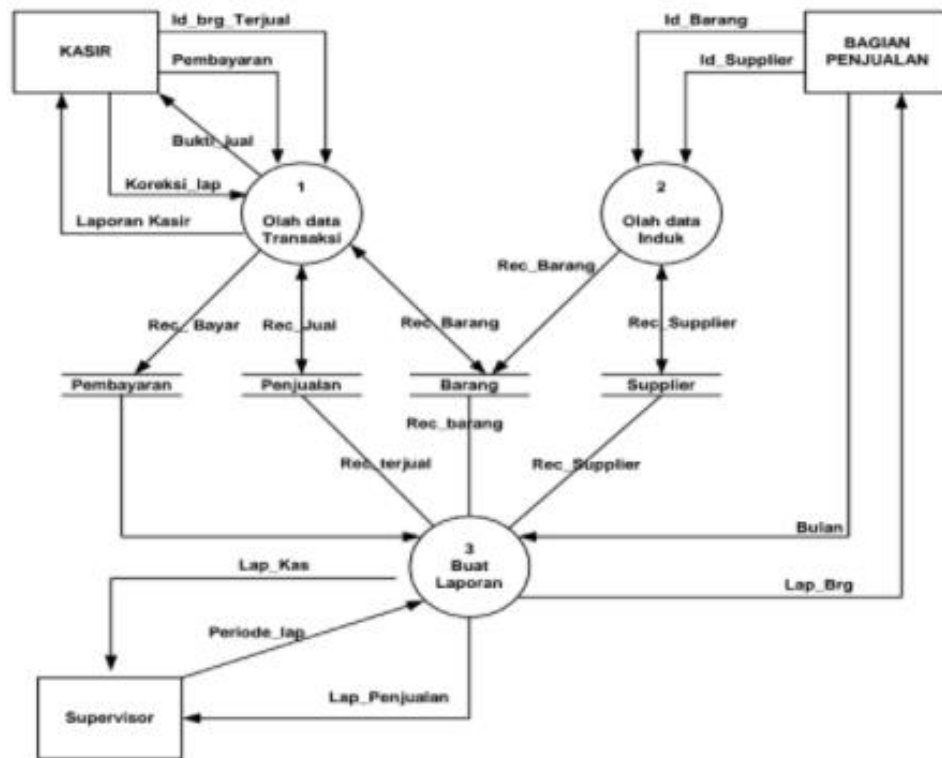
Diagram ini dibuat untuk menggambarkan sumber serta tujuan data yang Akan di proses atau dengan kata lain diagram tersebut digunakan untuk menggambarkan sistem secara umum atau global dari keseluruhan system yang ada (Tata, 2012).



GAMBAR 2.3. Diagram Konteks (en.wikipedia.org)

2. Diagram *Level 1*

Diagram *level 1* adalah pengembangan atau *devmpose* dari diagram konteks dan bisa mencakup sampai sembilan proses. Diagram *level 1* menunjukkan semua proses utama yang menyusun keseluruhan sistem. *Level 1* ini juga menunjukkan bagaimana proses – proses utama terhubung dengan entitas eksternal serta dilakukannya penambahan data store. Setiap proses diberi nomor bilangan bulat, umumnya dimulai dari sudut sebelah kiri atas diagram dan mengarah ke sudut sebelah kanan bawah. Penyimpanan data utamdari sistem (mewakili file master) dan semua entitas eksternal ke dalam diagram *level 1* (Maniah, Dini, 2017).



GAMBAR 2.4. Data flow diagram level 1 (Maniah, Dini, 2017).

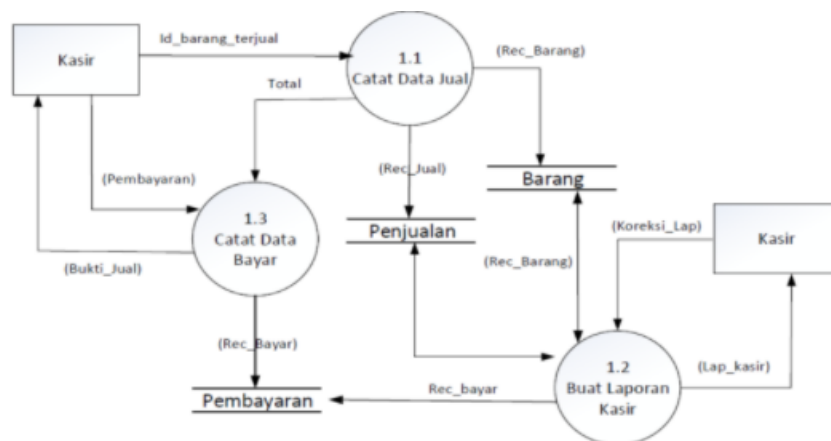
3. Diagram *level* Berikutnya

Diagram ini merupakan dekomposisi dari diagram *level 1*. Setiap proses dalam diagram *level 1* bisa dikembangkan untuk menciptakan diagram *level* berikutnya yang lebih mendetail. Proses pada diagram *level 1* yang dikembangkan itu disebut *parent* proses (proses induk) dan diagram yang dihasilkan disebut *child* diagram (diagram anak). Aturan utama untuk

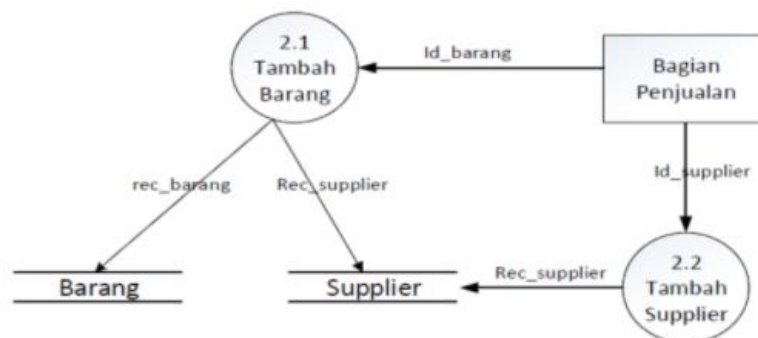
menciptakan diagram anak adalah keseimbangan vertical, dimana menyatakan bahwa suatu diagram anak tidak bisa menghasilkan keluaran atau menerima masukan dimana proses induknya juga tidak menghasilkan atau menerima. Semua aliran data yang menuju atau keluar dari proses induk harus ditunjukkan mengalir ke dalam atau keluar dari diagram anak.

Bila proses induk memiliki aliran data yang terhubung ke penyimpanan data, diagram anak bisa memasukkan penyimpanan data tersebut. Selain itu Diagram pada *level* yang lebih rendah bisa memasukkan penyimpanan data yang tidak ditunjukkan dalam proses induk.

Sebagai catatan bahwa bisa saja penyusunan DFD tidak mencapai level 2 atau mungkin harus dilanjutkan ke level berikutnya (*level 3, level 4, dan seterusnya*) (Maniah, Dini, 2017).



GAMBAR 2.5. Data flow diagram level 2 Proses 1 (Maniah, Dini, 2017).



GAMBAR 2.6. Data flow diagram level 2 Proses 2 (Maniah, Dini, 2017).

2.7. Database MySQL

MySQL merupakan database server open source yang cukup populer keberadaannya. Dengan berbagai keunggulan yang dimiliki, membuat *software database* ini banyak digunakan oleh para praktisi untuk membangun suatu project. Adanya fasilitas API (*Application Programming Interface*) yang dimiliki oleh MySQL, memungkinkan bermacam-macam aplikasi komputer yang ditulis dengan berbagai bahasa pemrograman dapat mengakses data MySQL (Wahana Komputer,2010).

Database adalah sebuah struktur yang umumnya terbagi dalam 2 hal, yaitu sebuah *database* relasional dan Sebuah *database flat*. *Database* relasional lebih mudah dipahami daripada *database flat* karena *database* relasional mempunyai bentuk yang sederhana serta mudah dilakukan operasi data. MySQL sendiri adalah sebuah *database* relasional. *Database* yang memiliki struktur relasional terdapat tabel-tabel untuk menyimpan data. Pada setiap tabel terdiri dari kolom dan baris serta sebuah kolom untuk mendefinisikan jenis informasi apa yang harus disimpan (Wahana Komputer,2010).

2.7.1. Tabel Pada Database

Tabel adalah inti konsep *databases*. Tujuannya adalah menyimpan informasi. Tabel satu dengan lainnya bisa dihubungkan. Satu *database* dapat mengandung banyak tabel dalam jumlah tak terbatas. Jumlah record dalam tabel juga umumnya tak dibatasi, batasnya hanya pada kapasitas penyimpanan dalam disk yang digunakan. Tiap tabel memiliki beberapa kolom yang menentukan nilai-nilai apa yang disimpan. Setiap baris dari tabel bisa mengandung nilai kolom-kolom yang ditentukan di tabel. Selain itu, tiap kolom bisa ditentukan tipe data apa saja yang didukung. Tipe data menentukan jenis data yang bisa disimpan. Tipe data bisa dibatasi seperti jenis numeric, text, huruf, tanggal, atau yang sebagainya (Wahana Komputer,2010).

2.7.2. Query Pada Database

Query adalah sebuah perintah mengambil informasi tertentu yang disimpan di tabel atau multitabel. Untuk membuat *query*, harus menentukan parameter-parameter dari informasi yang dicari, Misalnya, Mengambil data pelanggan yang telah membeli barang tipe X selama 3 bulan yang lalu. Dengan demikian harus

mengisi nama variabel dan priode. Informasi yang ditampilkan juganantinya bisa diurutkan, difilter dan diatur cara menampilkannya (Wahana Komputer,2010).

2.8 Microsoft .Net Framework

Microsoft .Net Framework (Juga dikenal sebagai Bahasa pemrograman.Net) adalah teknologi yang menyediakan infrastruktur untuk membangun aplikasi yang berjalan pada sistem operasi windows. Saat ini .Net Framework umumnya telah terintegrasi dalam distribusi standar windows (mulai dari Windows Server 2003 hingga Windows 10) (Wali, 2014).

.Net Framework terdiri dari *Common Langue Runtame* (CLR) dan *class library runtime*. CLR adalah dasar dari .Net Framework seperti namanya, CLR menyediakan insfratukture umumuntuk semua bahasa .Net. Insfratuktur ini bertanggung jawab untuk mengambil kontrol dari eksekusi aplikasi dan mengelola tugas-tugas seperti manajemen memori, akses ke sistem sumber daya, keamanan dan sebagainya (Wali, 2014).

.Net Framework Class Library adalah koleksi dari *reusebke types* yang sangat terintergrasi secara melekat dengan CLR, *class library* bersifat berorientasi terhadap objek yang akan menyediakan tipe dari fungsi-fungsi managed code (Wali, 2014).

Microsoft .Net Framework menawarkan bahasa .Net Menggunakan *Visual Studio* (Wali, 2014).

2.9. Visual Studio

(Jubilee, 2013) *Visual studio* adalah IDE (*Integrated Development Environment*) yang dapat digunakan untuk mengembangkan aplikasi-aplikasi windows. *Visual Studio* dirancang untuk focus pada produktifitas. *Tool* ini disebut juga *Rapid Application Development tools* (RAD tools) karena dirancang dan dilengkapi untuk meningkatkan produktifitas. Versi baru dari *visual studio* ini mudah digunakan dan mudah dipelajari. Fitur-fitur dalam *visual studio* versi terbaru dibuat lebih sederhana untuk mempermudah pengguna dalam mempelajarinya dan memenuhi kebutuhan para *programmer*.

Dengan *visual studio* ini, dapat membuat jenis-jenis aplikasi berikut (Sumber Jubilee, 2013) :

1. Aplikasi *Windows*, aplikasi yang mempunyai antarmuka dengan tombol, jendela, *menu*, *toolbar*, dan seterusnya, seperti Microsoft Word atau Internet Explorer.
2. Aplikasi Konsol, Aplikasi yang tidak mempunyai antarmuka dan hanya menggunakan menggunakan teks untuk berkomunikasi dengan pengguna. (Biasanya, aplikasi ini berupa jendela *command* atau jendela DOS).
3. *Component* atau *class libraries*, sebuah kumpulan *tool-tool* yang dibuat untuk menunjang pengembangan aplikasi lainnya.

Daftar dibawah ini merupakan fitur utama dari *visual studio*. Kebanyakan fitur yang ada dalam daftar ini mengedepankan filosofi RAD. Berikut daftar beserta deksripsinya :

1. *Built-in Starter Kits*, aplikasi lengkap yang berisi contoh dan latihan-latihan pemrograman.
2. *Beginner's Targeted Documentation and Tutorials*, cara mudah dan cepat untuk memperoleh informasi dan contoh-contoh pemrograman.
3. *IntelliSense*, menyediakan petunjuk *syntax* secara *real-time*, mengoreksi, serta melengkapi sintaks secara otomatis.
4. *Code Snippets*, menyediakan kode untuk berbagai jenis tugas pemrograman untuk membantu menyelesaikan tugas secara otomatis. *Code snippets* menunjukkan cara yang direkomendasikan untuk menyelesaikan sesuatu. Fitur ini terintegrasi secara langsung ke dalam lingkungan pengembangan, dan dapat diperluas, Microsoft dan para anggota komunitas online memasok *code snippets* baru secara berkala.
5. *Data-enabled Application*, aplikasi ini memungkinkan untuk terhubung dengan Microsoft SQL Server dan menambahkan database dan kode untuk mengakses data dalam aplikasi.
6. *Windows From Designer*, dengan *tool* ini, dapat dengan mudah merancang aplikasi *windows* dengan fitur seperti *snap line* yang memastikan kontrol anda tertata rapi pada *form* dan *autocorrect* yang memberikan informasi *syntax* secara *real-time*.

7. *XML Web Services*, produk ini menyediakan *tool-tool* yang mudah digunakan dan *wizard* yang akan membantu untuk terhubung dengan *web services* berbasis XML dan menggunakannya dalam aplikasi.
8. *Windows Form Controls*, kumpulan kontrol yang lengkap dari *visual studio*. Kontrol-kontrol ini akan membantu membuat aplikasi antar muka yang menarik.
9. *Smart Tags*, kebanyakan kontrol dalam *windows form* memiliki *smart tags*. *Smart tag* dilambangkan dengan sebuah segitiga kecil atau sebuah ikon dan sebuah segitiga kecil yang melekat pada kontrol. Fitur ini memberikan akses pada semua hal yang dapat dilakukan untuk suatu kontrol.
10. *Refactoring*, IDE *visual studio* memiliki dukungan *refactoring*. *Refactoring* memungkinkan programmer untuk melakukan perintah-perintah secara otomatis saat proses *restructuring*. *Restructuring code* adalah saat mengubah sebagian kode dan mengaplikasikan perubahan pada semua *file* dari elemen itu. Bisa juga mengganti nama *variable* yang ada pada *project*, membuat *variable* lokal menjadi parameter dan masih banyak lagi. Fitur ini menampilkan *preview* sehingga dapat melihat perubahannya sebelum melakukannya.
11. *Click-One Deployment*, fitur ini memungkinkan menerbitkan dengan mudah aplikasi melalui internet, local area network (LAN), jaringan public, atau dengan media CD atau DVD. Fitur ini juga memudahkan dalam menerbitkan update aplikasi.
12. *Edit dan Continue*, sementara melakukan *debugging* aplikasi *edit dan Continue* memungkinkan memodifikasi kode, re-eksekusi kode, menambah fungsi, atau memperbaiki aplikasi secara langsung tanpa harus menghentikan proses eksekusi aplikasi.
13. *Debugger Visualizers*, Sementara melakukan *debugging* aplikasi, *visualizer* memberi presentasi *visual* yang mudah dimengerti mengenai data dalam aplikasi. Fitur ini menampilkan representasi data yang tersimpan dalam aplikasi.
14. *Community Access dan Start Page*, Dengan fitur ini, dapat mengakses informasi-informasi tambahan dari komunitas online dan dari berbagai sumber bantuan online, termasuk RSS (*Real Simple Syndication*) *feeds*. (RSS adalah

sejenis format file XML yang digunakan secara luas oleh komunitas weblog dan situs web berita).

15. *Simplified Development Environment*, semua fitur dalam lingkungan pengembangan telah dirancang agar dapat dengan mudah mengakses fungsional, *tool-tool* dan ojek yang ada.

Visual studio memiliki banyak fitur menarik untuk membantu programmer dalam membuat aplikasi secara cepat dan menyenangkan. fitur ini akan menyediakan pedoman yang mempermudah membuat aplikasi.

2.10. Testing

2.10.1. Pengertian *Testing*

Menurut Quadri dan Farooq (2010: 1), pengujian *software* adalah proses verifikasi dan validasi apakah sebuah aplikasi *software* atau program memenuhi persyaratan bisnis dan persyaratan teknis yang mengarahkan desain dan pengembangan dan cara kerjanya seperti yang diharapkan dan juga mengidentifikasi kesalahan yang penting yang digolongkan berdasarkan tingkat *severity* pada aplikasi yang harus diperbaiki.

Menurut Nidhra dan Dondeti (2012:1), pengujian *software* adalah teknik yang sering digunakan untuk verifikasi dan validasi kualitas suatu software. Pengujian *software* adalah prosedur untuk eksekusi sebuah program atau sistem dengan tujuan untuk menemukan kesalahan.

2.10.2. Behavioral Test (*Black-Box Test*)

Tester menggunakan *behavioral test* (disebut juga *Black-Box Tests*), sering digunakan untuk menemukan bug dalam high level operations, pada tingkatan fitur, profil operasional dan skenario *customer*. *Tester* dapat membuat pengujian fungsional *black box* berdasarkan pada apa yang harus sistem lakukan. *Behavioral testing* melibatkan pemahaman rinci mengenai domain aplikasi, masalah bisnis yang dipecahkan oleh sistem dan misi yang dilakukan sistem. *Behavioral test* paling baik dilakukan oleh penguji yang memahami desain sistem, setidaknya pada tingkat yang tinggi sehingga mereka dapat secara efektif menemukan bug umum untuk jenis desain.

Menurut Nidhra dan Dondeti (2012:1), *black box testing* juga disebut *functional testing*, sebuah teknik pengujian fungsional yang merancang *test case* berdasarkan informasi dari spesifikasi.

3. HASIL DAN PEMBAHASAN

3.1. Analisis Sistem

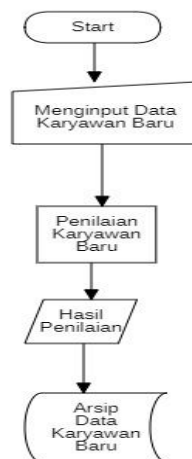
Analisis ini bertujuan untuk mengidentifikasi permasalahan-permasalahan yang terdapat pada sistem yang akan dibangun. Bisa dikatakan analisis sistem merupakan penguraian dari suatu sistem untuk mengidentifikasi masalah-masalah dalam sistem yang akan dibuat atau sedang berjalan. Sehingga akan dapat diusulkan kebutuhan-kebutuhan untuk memperbaikinya.

3.2. Analisis Masalah

Analisis masalah merupakan langkah dimana langkah ini diperlukan untuk mengetahui permasalahan apa saja yang terjadi di dalam sistem yang sedang berjalan. Adapun identifikasi masalah pada penelitian ini adalah menggunakan

3.3. Analisis Sistem Yang Sedang Berjalan

Analisis ini menjelaskan kondisi sistem yang berjalan sebelum penelitian.



GAMBAR 3.1. *Flowchart* sistem yang sedang berjalan

3.4. *Planning*

Dalam tahap ini penelitian memfokuskan pada penjadwalan pengerjaan penelitian. Pada penelitian ini terdapat beberapa proses yang harus dilakukan dari

tahap *communication* hingga *implementation* dan *testing* maka dari itu diperlukan penjadwalan yang tepat agar penelitian ini dapat selesai pada waktunya. Berikut penjadwalan penelitian berdasarkan aktifitas yang dilakukan dengan skala waktu. Tabel 3.6 dibawah ini akan menjelaskan penjadwalan penelitian

TABEL 3.1. Penjadwalan Penelitian.

Tahap	No	Aktivitas	November				Desember				Januari			
			1	2	3	4	1	2	3	4	1	2	3	4
<i>Scheduling</i>	1	Penjadwalan												
<i>Initiation - Planing</i>	1	Pengumpulan Data												
	2	Analisis Permasalahan dan Kebutuhan												
	3	Penjadwal Penelitian												
<i>Requirement Gathering and Analysis</i>	1	Analisis Proses Sistem Yang Akan dibangun												
<i>disign</i>	1	Perancangan DFD dan Database Program												
	2	Perancangan Interface												
<i>Build Or Coding And Development</i>	1	Pengkodean												
	2	Testing												

3.5. Requirement Gathering and Analysis

3.5.1. Analisis Proses

3.5.1.1. Analisis Proses Pendukung Keputusan

Analisi sistem pendukung keputusan ini direncanakan untuk sitem pendukung keputusan dalam aplikasi yang menggunakan metode *fuzzy logic*. Analisis ini bertujuan untuk mengetahui bagaiman cara kerja dari metode *fuzzy logic* dalam sitem pendukung keputusan untuk penerimaan kariyawan baru.

3. 5.1.2. Analisis Proses *Fuzzy Logic*

3. 5.1.2.1 Kriteria Penilaian Karyawan Baru

Pengumpulan data dilakukan dengan mewawancarai Pegawai dari Perusahaan. Wawancara dilakukan untuk mengetahui apa yang dibutuhkan dalam penilaian karyawan baru. Dari Hasil wawancara yang dilakukan, ditetapkan beberapa kriteria dan nilai bobot dari setiap kriteria, yaitu

TABEL 3.2. Tabel Variabel Penilaian.

Variabel (Kriteria)
Nilai Test Matematika
Nilai Test Kepribadian
Nilai Test Pengetahuan Umum
Nilai Test Praktik
Memiliki Sim C
Memiliki Motor
Nilai Pengalaman Kerja
Nilai Umur

Dari 5 kriteria yang digunakan dalam penelitian ini mempunyai nilai linguistik terendah dan tertinggi yang masing-masing mempunyai himpunan *fuzzy* (himpunan crisp).

TABEL 3.3. Variabel Penilaian Karyawan Baru.

Nama Variabel (Kriteria)	Himpunan Fuzzy	[Nilai]
Nilai Test Matematika	Kurang, Cukup, dan Baik	[0 - 100]
Nilai Test Kepribadian	Kurang, Cukup, dan Baik	[0 - 100]
Nilai Test Pengetahuan Umum	Kurang, Cukup, dan Baik	[0 - 100]
Nilai Test Praktik	Kurang, Cukup, dan Baik	[0 - 100]
Memiliki Sim C	Memiliki dan Tidak Memiliki	[0 - 1]
Memiliki Motor	Memiliki dan Tidak Memiliki	[0 - 1]
Nilai Pengalaman Kerja	Kurang, Cukup, dan Baik	[0-4]
Nilai Umur	Kurang, Cukup, dan Baik	[20-50]

Dari 5 Himpunan *Fuzzy* memiliki tingkatan fuzzy masing -masing seperti yang tertera dalam tabel 3.4.

TABEL 3.4. Tingkatan *Fuzzy*.

Nama Variabel (Kriteria)	Tingkatan Fuzzy	Himpunan Fuzzy
Nilai Test Matematika	0 – 60	Kurang
	61 - 79	Cukup
	80 - 100	Baik
Nilai Test Kepribadian,	0 – 60	Kurang
	61 - 79	Cukup
	80 - 100	Baik
Nilai Test Pengetahuan Umum	0 – 60	Kurang
	61 - 79	Cukup
	80 - 100	Baik
Nilai Test Praktik	0 – 60	Kurang
	61 - 79	Cukup
	80 - 100	Baik
Memiliki Sim C	0	Tidak Memiliki
	1	Memiliki
Memiliki Motor	0	Tidak Memiliki
	1	Memiliki
Nilai Pengalaman Kerja	0	Kurang
	1-2	Cukup
	3-4	Baik
Nilai Umur	50-38	Kurang
	37-30	Cukup
	30-20	Baik

Dari 5 himpunan kriteria dan dari setiap 3 himpunan *fuzzy* memiliki bobotnya masing – masing seperti yang tertera dalam tabel 3.5.

TABEL 3.5. Pembobotan.

Variabel (Kriteria)	Pembobotan		
	Baik	Cukup	Kurang
Nilai Test Matematika	20	10	8
Nilai Test Kepribadian	10	7	6
Nilai Test Pengetahuan Umum	13	9	7
Nilai Test Praktik	13	9	8
Nilai Pengalaman Kerja	7	4	3
Nilai Umur	7	4	3
Memiliki SIM C	Tidak Memiliki	Memiliki	-
	0	20	-
Memiliki Motor	0	10	-

Jadi peserta dinyatakan bisa diterima dengan bobot minimal 75 dan peserta tidak bisa diterima bobot berada di bawah 75.

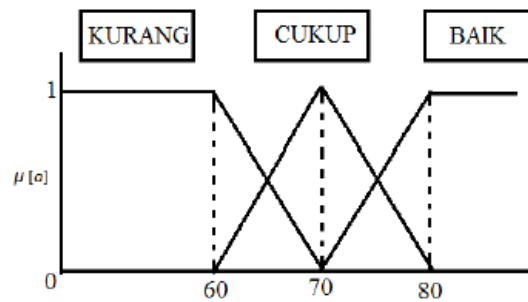
3.5.1.2.2 Fuzzikasi dan Penalaran.

1. Nilai test matematika, nilai test kepribadian, nilai test pengetahuan umum, nilai test praktik. Nilai pada variabel terdiri dari 3 himpunan *fuzzy* dengan tingkatan *fuzzy*, yaitu :

Contoh :

- Kurang = (Misalkan 60)
- Cukup = (Misalkan 70)
- Baik = (Misalkan 80)

Dengan grafik fungsi keanggotaan akan tampil seperti gambar 3.1.



GAMBAR 3.1. Grafik fungsi keanggotaan Nilai test matematika, nilai test kepribadian, nilai test pengetahuan umum, nilai test praktik.

Maka penalaran pada nilai test dapat dirumuskan sebagai berikut :

IF $x \geq Min$ dan $x \leq Max = Kurang$

IF $x \geq Min$ dan $x \leq Max = Cukup$

IF $x \geq Min$ dan $x \leq Max = Baik$

Keterangan

x = Nilai Inputan Nilai Test

Min = Nilai Minimum Tingkatan Fuzzy

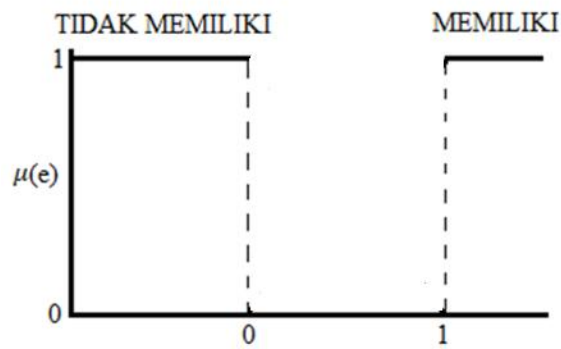
Max = Nilai Maximum Tingkatan Fuzzy

2. Kriteria Memiliki Motor dan Sim C

Nilai pada variabel terdiri dari 2 himpunan *fuzzy*, yaitu

- Tidak Memiliki (0)
- Memiliki (1)

Dengan grafik fungsi keanggotaan akan tampil seperti gambar 3.2.



GAMBAR 3.2. Grafik fungsi keanggotaan memiliki motor dan sim c

Maka penalaran pada memiliki sim c dan motor dapat dirumuskan sebagai berikut :

IF e = f = Tidak Memiliki

IF e = f = Memiliki

Keterangan

e = Nilai Inputan Memiliki Motor dan sim c

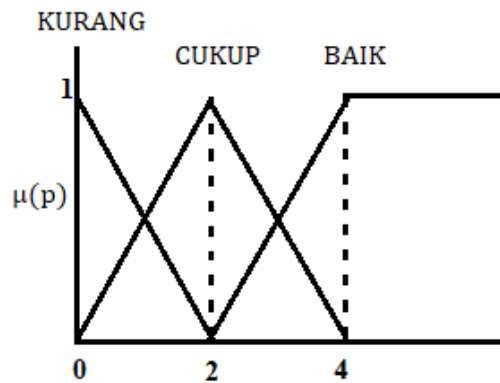
f = Nilai Tingkatan *Fuzzy*

3. Kriteria Nilai Pengalaman Kerja

Nilai Pengalaman Kerja.. Memiliki 3 himpunan *fuzzy* dengan tingkatan *fuzzy*, yaitu:

Contoh :

- Kurang = (Misalkan 0)
- Cukup = (Misalkan 2)
- Baik = (Misalkan 4)



GAMBAR 3.3. Grafik fungsi keanggotaan memiliki Nilai Pengalaman Kerja

Maka penalaran pada nilai pegalaman kerja dapat dirumuskan sebagai berikut :

IF $p = 0 = Kurang$

IF $p \geq Min$ dan $p \leq Max = Cukup$

IF $p \geq Min$ dan $p \leq Max = Baik$

Keterangan

p = Nilai Inputan Nilai Pengalaman Kerja

Min = Nilai Minimum Tingkatan Fuzzy

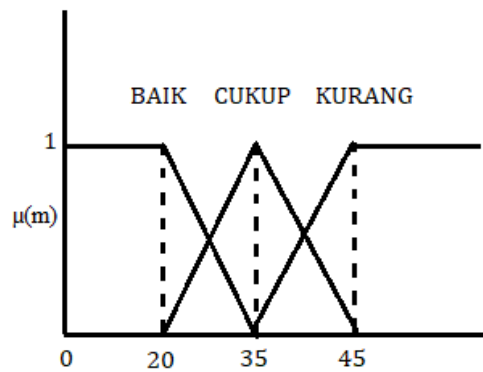
Max = Nilai Maximum Tingkatan Fuzzy

4. Kriteria Nilai Umur

Nilai Pengalaman Kerja.. Memiliki 3 himpunan *fuzzy* dengan tingkatan *fuzzy*, yaitu:

Contoh :

- Kurang = (Misalkan 45)
- Cukup = (Misalkan 35)
- Baik = (Misalkan 20)



GAMBAR 3.4. Grafik fungsi keanggotaan memiliki Nilai Umur

Maka penalaran pada nilai pengalaman kerja dapat dirumuskan sebagai berikut :

$IF m \geq Min \text{ dan } m \leq Max = Kurang$

$IF m \geq Min \text{ dan } m \leq Max = Cukup$

$IF m \geq Min \text{ dan } m \leq Max = Baik$

Keterangan

m = Nilai Inputan Nilai Umur

Min = Nilai Minimum Tingkatan Fuzzy

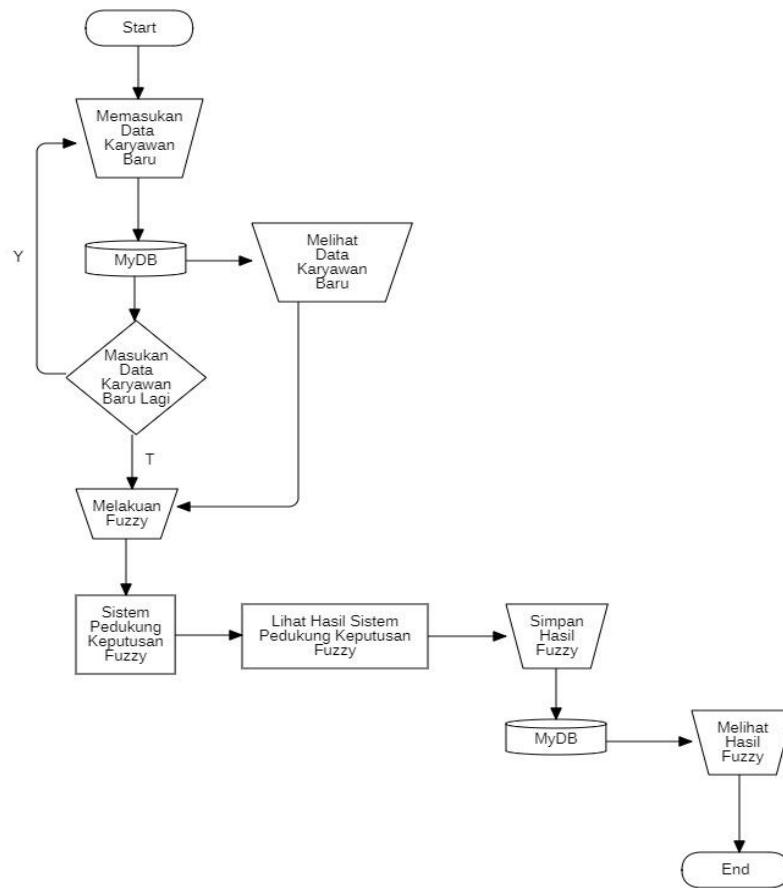
Max = Nilai Maximum Tingkatan Fuzzy

3.5.1.3. Analisis Proses Aplikasi

Analisis ini direncanakan untuk sistem aplikasi yang sedang berjalan bertujuan untuk mengetahui bagaimana cara kerja suatu sistem dan mengetahui masalah yang dihadapi sistem untuk dapat dijadikan sebagai landasan usulan perancangan. Tahap analisis ini diperlukan untuk mengetahui bagaimana proses sistem berjalan. Tahap perancangan ini, penulis menggambarkan sistem yang berjalan pada aplikasi dalam bentuk *flowchart*.

Adapun rancangan yang di gambarkan dengan menggunakan *flowchart* seperti di bawah ini :

FLOWCHART SISTEM SEDANG BERJALAN



GAMBAR 3.5. *Flowchart* sistem yang di usulkan.

Pada *flowchart* diatas adalah rancangan sistem dari awal aplikasi berjalan hingga akhir aplikasi di hentikan.

Adapun sistem yang akan dibangun ini adalah sebuah sistem aplikasi sistem pendukung keputusan penerimaan karywan baru berbasis *desktop*, dimana aplikasi ini dibuat sebagai pendukung keputusan dalam penerimaan karyawan. Ini sistem yang dirancang dalam bentuk *flowchart*.

3.6. Design

3.5.1. Desain Perangkat Lunak

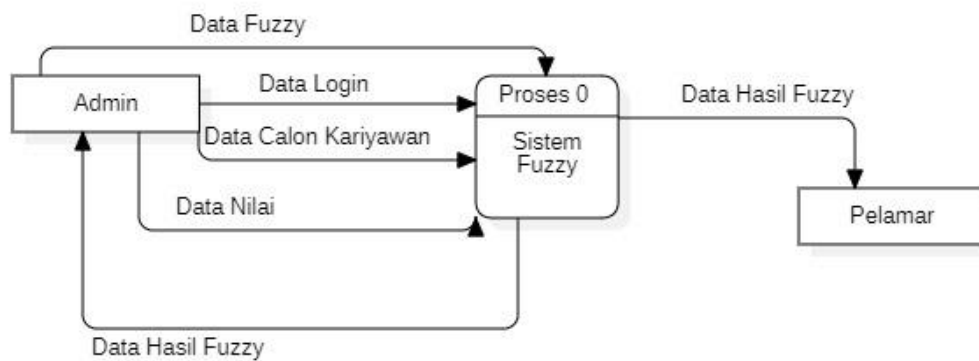
Desain atau perancangan merupakan tahap untuk memenuhi kebutuhan *user* mengenai gambaran yang jelas tentang perancangan dari sistem yang akan dibuat setiap diimplementasikan. Adapun dalam tahap perancangan ini terdapat sistem dengan notasi DFD (*Data Flow Diagram*), perancangan sistem, dan perancangan *interface*.

3.6.2. Desain Perancangan Sistem

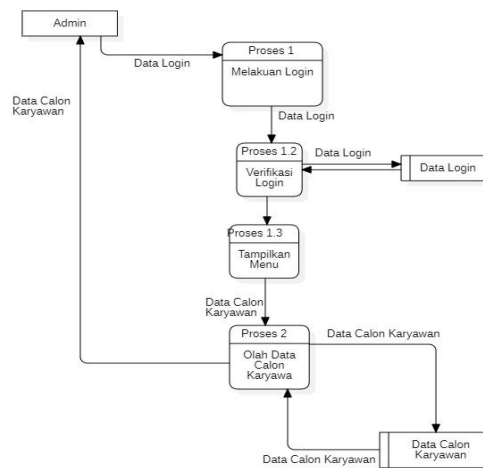
Dalam perancangan sistem perangkat lunak, penulis menggunakan model berorientasi objek dengan menggunakan notasi DFD (*Data Flow Diagram*) dan Perancangan sistem perangkat lunak ini juga memiliki Relasional basisdata.

3.6.2.1. *Data Flow Diagram* (DFD)

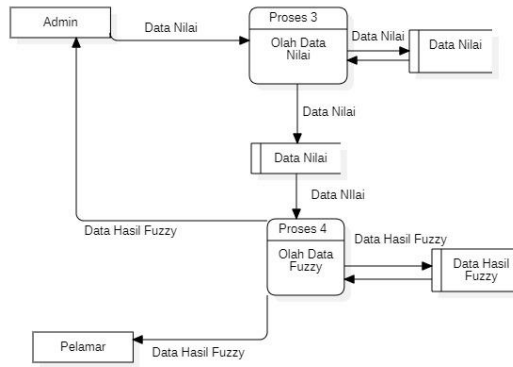
Data Flow Diagram yang menggambarkan sistem perangkat lunak ini di tunjukan pada gambar di bawah ini :



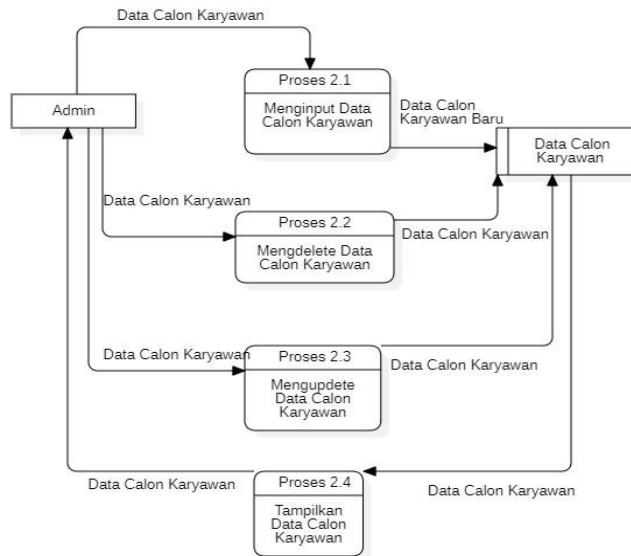
GAMBAR 3.6. Diagram Konteks.



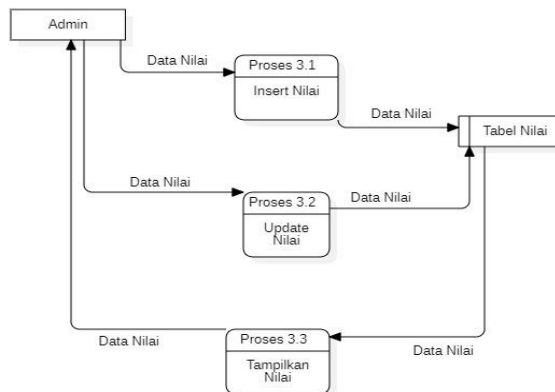
GAMBAR 3.7. *Data flow diagram level 1* Proses 1 – Proses 2.



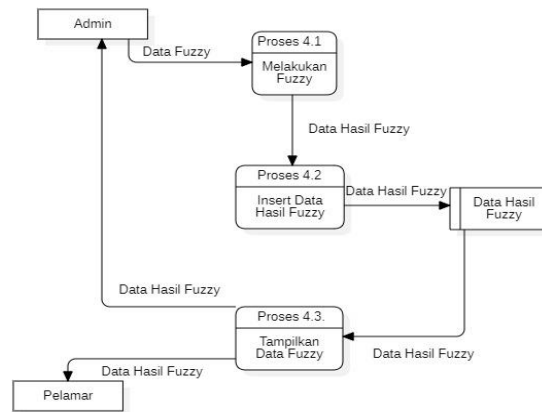
GAMBAR 3.8. *Data flow diagram level 1* Proses 3 – Proses 4.



GAMBAR 3.9. *Data flow diagram level 2* Proses 2.



GAMBAR 3.10. *Data flow diagram level 2* Proses 3.



GAMBAR 3.11. *Data flow diagram level 2* Proses 4.

3.7. Development

3.7.1 Build Or Coding

Dalam tahap ini penelitian berfokus pada pengkodean menggunakan Bahasa .Net Framwork, dan setelah itu dilakukan pengujian hasil menggunakan metode *blackbox testing*. *Blackbox testing* menguji spesifikasi suatu fungsi atau modul, apakah berjalan sesuai dengan semestinya dan sesuai dengan apa yang diharapkan atau tidak.

3.7.1.1. Implementasi

Setelah aplikasi di analisis dan dirancang secara rinci, maka selanjutnya menuju tahap implementasi. Implementasi merupakan tahap meletakkan sistem sehingga siap untuk dioperasikan. Berikut ini adalah tahap implementasi aplikasi sistem pendukung keputusan.

3.7.1.1.1. Implementasi *Haradware* dan *Software*

Dalam tahap ini peneliti memerlukan spesifikasi *hardware* dan *software* untuk mengimplementasikan aplikasi. Untuk lebih jelasnya akan dijelaskan dibawah ini :

3.7.1.1.1.1. *Hardware*

Hardware yang digunakan sebagai berikut :

1. *Processor* Intel i3 2.0 Ghz

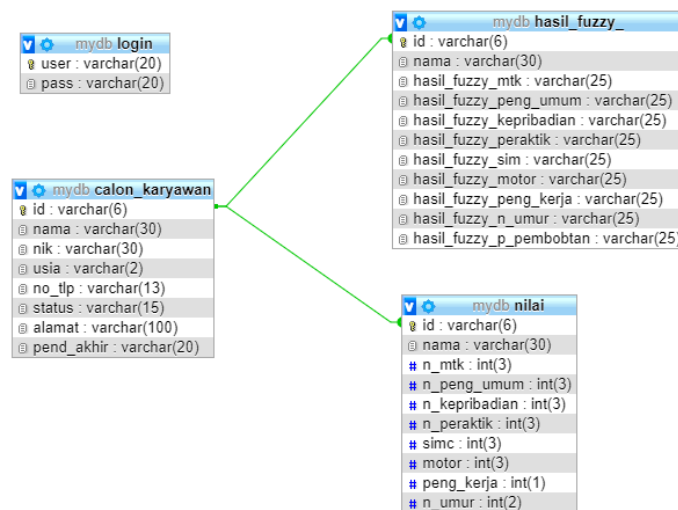
2. Memory RAM 4GB DDR3L 2133Mhz
3. Hardisk 500 GB SATA 5400RPM

3.7.1.1.1.2. Software

1. Sistem Operasi Windows 10
2. Visual Studio 2019
3. Xampp v3.3.2

3.7.1.2. Implementasi Database

Pada Implementasi ini menjelaskan tentang tabel dan relasi tabel dalam database. Disini database menggunakan MySQL dengan nama database mydb dan memiliki 4 tabel dengan relasi antara entitas id pada tabel calon_karyawan sebagai *primary key* dan entitas id pada tabel nilai dan hasil_fuzzy_ sebagai *foreign key*. Untuk lebih jelasnya bisa dilihat pada gambar 3.12.

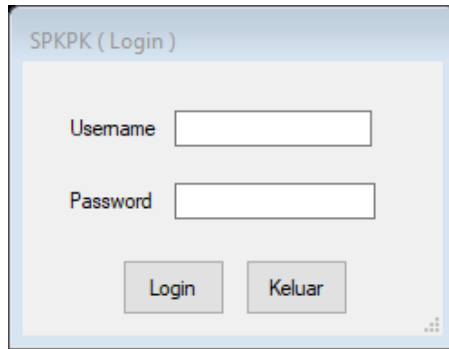


GAMBAR 3.12. Database, tabel data, dan relasi tabel

3.7.1.3. Implementasi Antar Muka

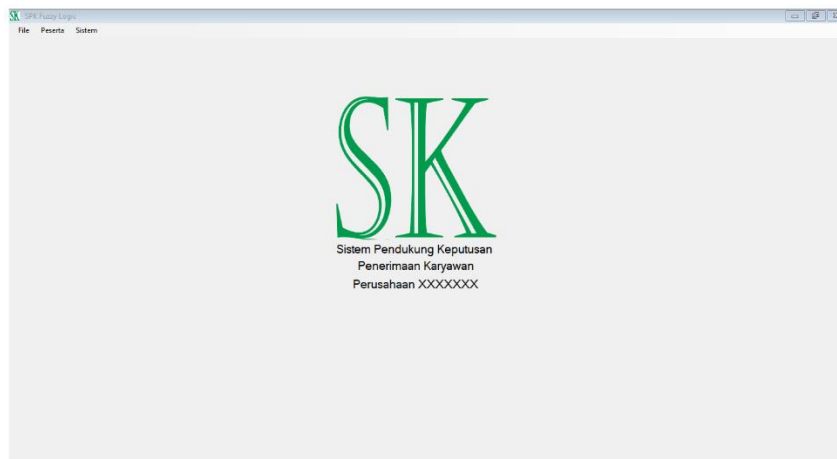
Pada implementasi ini menjelaskan tentang tampilan dari form – form yang diimplementasikan bisa dilihat di bawah ini :

Pada Gambar 3.13. menjelaskan tampilan login.

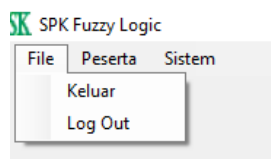


GAMBAR 3.13. Tampilan login aplikasi.

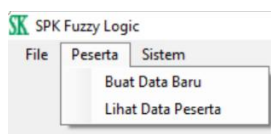
Pada Gambar ini menjelaskan tampilan menu, jadi pengguna (*user*) dapat memilih bagian yang diinginkan. Tertera dalam gambar 3.14. sampai gambar 3.17.



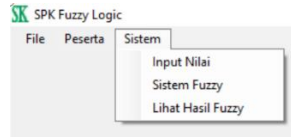
GAMBAR 3.14. Tampilan menu.



GAMBAR 3.15. Tampilan menu file.

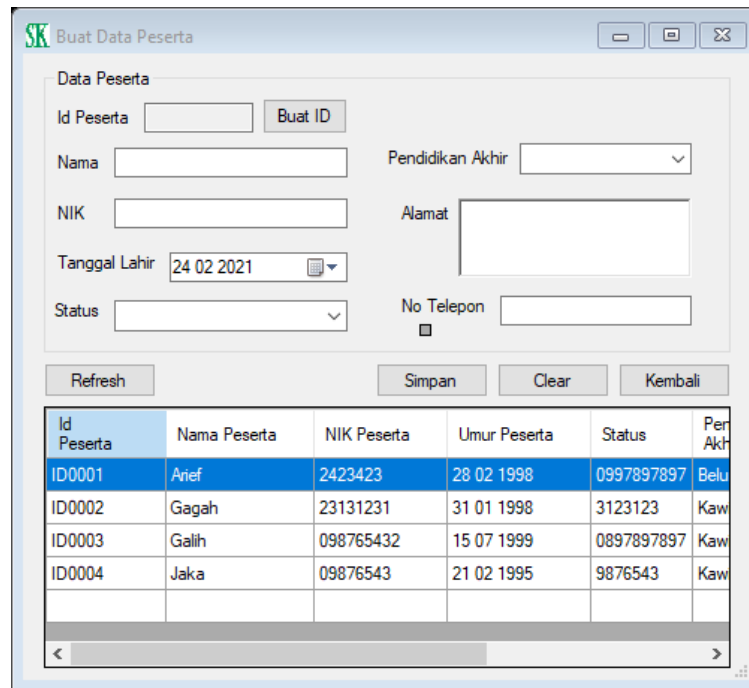


GAMBAR 3.16. Tampilan menu peserta.



GAMBAR 3.17. Tampilan menu sistem.

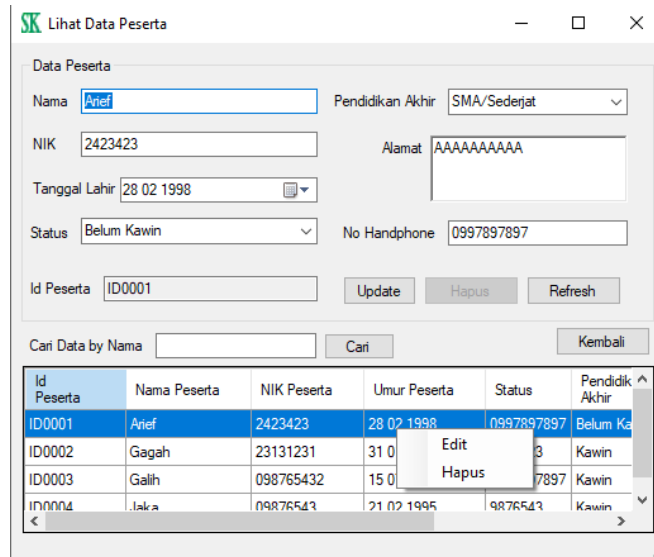
Pada gambar 3.18. menunjukkan hasil pemilihan Peserta menu Buat Data Baru.



Id Peserta	Nama Peserta	NIK Peserta	Umur Peserta	Status	Per Akh
ID0001	Arief	2423423	28 02 1998	0997897897	Belu
ID0002	Gagah	23131231	31 01 1998	3123123	Kaw
ID0003	Galih	098765432	15 07 1999	0897897897	Kaw
ID0004	Jaka	09876543	21 02 1995	9876543	Kaw

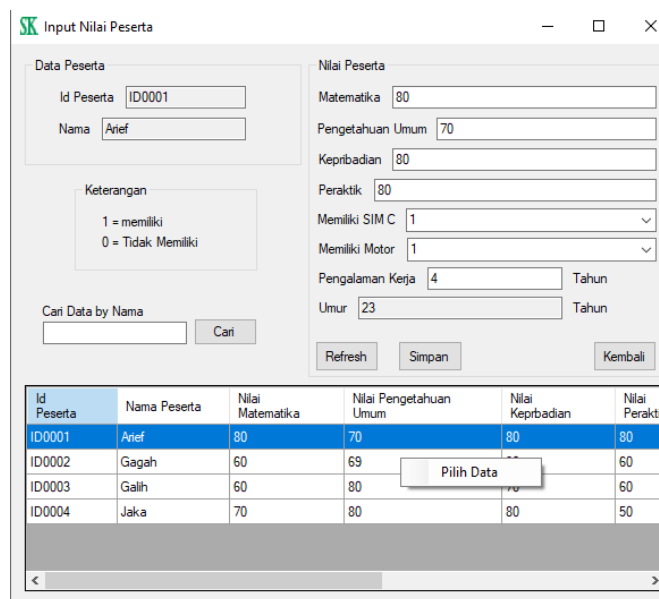
GAMBAR 3.18. Hasil pemilihan Peserta menu Buat Data Baru.

Pada gambar 3.19. menunjukkan hasil pemilihan Peserta menu Lihat Data Peserta.



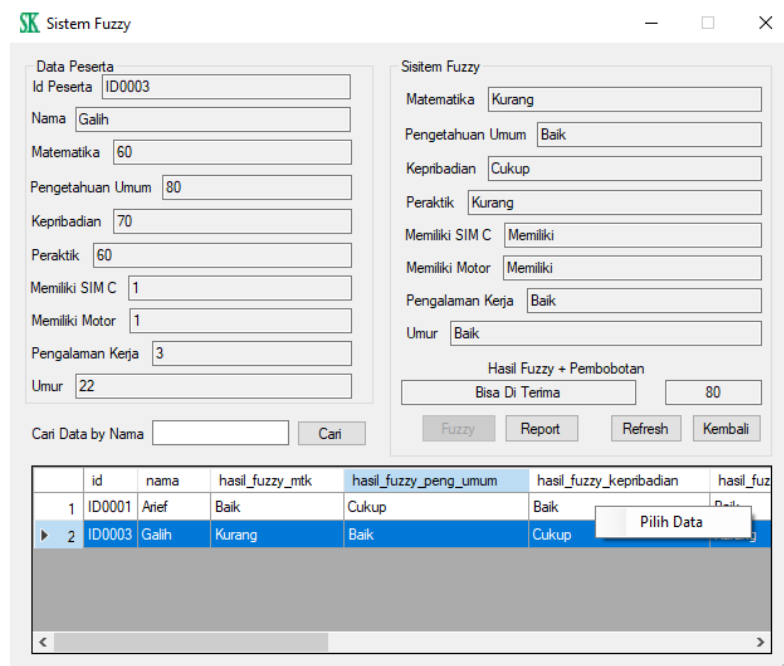
GAMBAR 3.19. Hasil pemilihan Peserta menu Lihat Data Peserta.

Pada gambar 3.20. menunjukkan hasil pemilihan Sistem Menu Input Nilai.



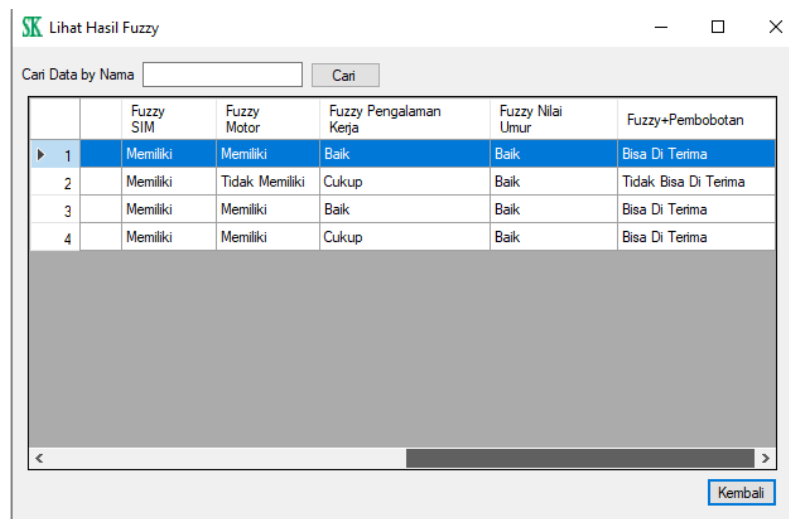
GAMBAR 3.20. Hasil pemilihan Sistem Menu Input Nilai.

Pada gambar 3.21. menunjukkan hasil pemilihan Sistem Menu Sistem Fuzzy.



GAMBAR 3.21. Hasil pemilihan Sistem Menu Sistem Fuzzy.

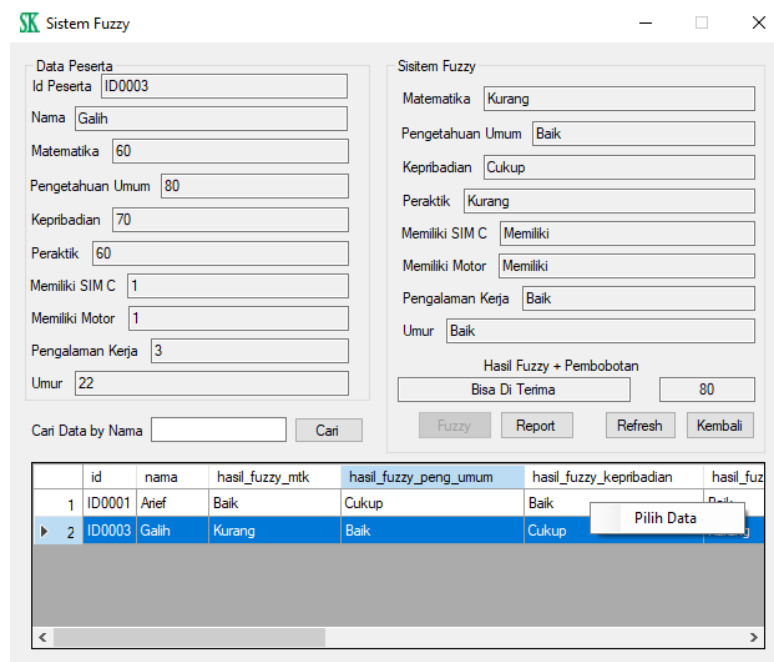
Pada gambar 3.22. menunjukkan hasil pemilihan Sistem Menu Lihat Hasil Fuzzy.



GAMBAR 3.22. Hasil pemilihan Sistem Menu Lihat Hasil Fuzzy.

3.7.1.4. Implementasi Sistem Pendukung Keputusan Metode *Fuzzy Logic*

Pada gambar 3.23. menunjukkan hasil dari pengimplementasian sistem pendukung keputusan metode *Fuzzy Logic*.



GAMBAR 3.23. Implementasian SPK metode *Fuzzy Logic*.

3.7.2. Testing

Testing atau pengujian merupakan bagian penting dalam siklus pengembangan perangkat lunak. Pengujian dilakukan untuk menjamin kualitas dan juga mengetahui kelemahan dari perangkat lunak. Tujuan dari pengujian perangkat lunak adalah untuk menjamin bahwa perangkat lunak yang dibangun itu berkualitas.

3.7.2.1. Testing *Balckbox*

Pengujian *blackbox* digunakan untuk mengetahui fungsi – fungsi khusus dari aplikasi atau perangkat lunak yang dibuat.

TABEL 3.6. Rancangan Pengujian.

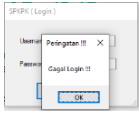
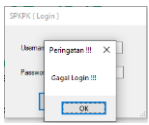

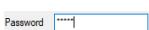
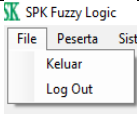
No	Kelas Uji	Item Uji	Butir Pengujian	Jenis Pengujian
1	Menu / Button	Form Login	Memilih Button Keluar	Balckbox
			Memilih Button Login	Balckbox
		Form Menu	Memilih Menu File	Balckbox
			Memilih Menu Keluar	Balckbox
			Memilih Menu Log Out	Balckbox


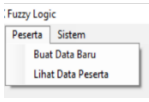
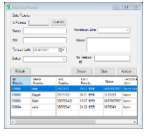
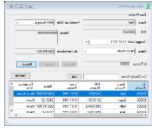
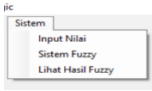
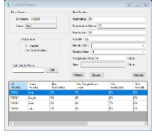
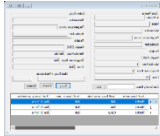
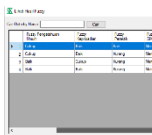
			Memilih Menu Peserta	Balckbox
			Memilih Menu Buat Data Baru	Balckbox
			Memilih Menu Lihat Data Peserta	Balckbox
			Memilih Menu Sistem	Balckbox
			Memilih Menu Input Nilai	Balckbox
			Memilih Menu Sistem Fuzzy	Balckbox
			Memilih Menu Lihat Hasil	Balckbox
		Form Buat Data Baru	Memilih Button Buat ID	Balckbox
			Memilih Button Refresh	Balckbox
			Memilih Button Simpan	Balckbox
			Memilih Button Clear	Balckbox
			Memilih Button Keluar	Balckbox
		Form Lihat Data Peserta	Memilih Button Refresh	Balckbox
			Memilih Button Hapus	Balckbox
			Memilih Button Update	Balckbox
			Memilih Button Kembali	Balckbox
		Form Input Nilai	Memilih Button Refresh	Balckbox
			Memilih Button Simpan	Balckbox
			Memilih Button Kembali	Balckbox
		Form Sistem Fuzzy	Memilih Button Fuzzy	Balckbox
			Memilih Button Simpan	Balckbox
			Memilih Button Kembali	Balckbox
			Memilih Button Report	
		From Lihat Hasil	Memilih Button Kembali	Balckbox

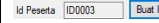
		Cari Data by Nama	Memilih Button Cari	Balckbox
2	Context Menu Strip	Form Lihat Data Peserta	Memilih Menu Edit	Balckbox
			Memilih Menu Hapus	Balckbox
		Form Input Nilai	Memilih Menu Pilih Data	Balckbox
		Form Sistem Fuzzy	Memilih Menu Pilih Data	Balckbox
		Form Lihat Hasil	Memilih Menu Pilih Data	Balckbox
3	TextBox	Form Login	TextBox Password	Balckbox
		Form Buat Data Baru	TextBox Nama	Balckbox
			TextBox NIK	Balckbox
			TextBox Umur	Balckbox
			TextBox Status	Balckbox
			TextBox No Telepon	Balckbox
		Form Lihat Data Peserta	TextBox Nama	Balckbox
			TextBox NIK	Balckbox
			TextBox Umur	Balckbox
			TextBox Status	Balckbox
			TextBox No Telepon	Balckbox
		Form Input Nilai	TextBox Matematika	Balckbox
			TextBox Pengetahuan Umum	Balckbox
			TextBox Kepribadian	Balckbox
			TextBox Peraktik	Balckbox

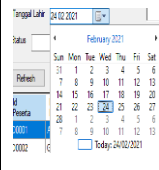
Setelah dilakukan pengujian dengan menggunakan meode *blackbox* maka didapatkn hasil pengujian pada tabel berikut ini:

TABEL 3.7. Pengujian *BlackBox Test*

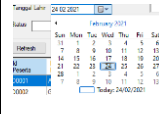
No	Item Uji	Input	Proses	Yang Diharapkan	Output	Hasil Pengujian
1	Form Login	Memilih Button Keluar	Keluar dari Aplikasi	Keluar Dari Aplikasi SPK	Aplikasi Behernti	Berhasil
		Memilih Button Login	Jika Username terisi benar dan password terisi salah	Tampil Pasan Box Gagal Login !!!		Berhasil
			Jika Username terisi salah dan password terisi benar	Tampil Pasan Box Gagal Login !!!		Berhasil
			Jika Username terisi Benar dan password terisi benar	Masuk ke from Menu Utama		Berhasil
		TextBox Password TextChar = *	Jika memasukan password text menjadi *	Password text menjadi *		Berhasil
2	From Menu	Memilih Menu File	Tampil list menu strip	Tampil list menu strip file		Berhasil

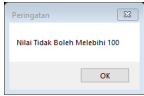
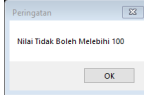
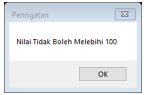
		Memilih Menu Keluar	Keluar dari Aplikasi	Keluar Dari Aplikasi SPK	Aplikasi Berhenti	Berhasil
		Memilih Menu Log Out	Melakukan logout Aplikasi	Tampil form login		Berhasil
		Memilih Menu Peserta	Tampil list menu strip	Tampil list menu strip peserta		Berhasil
		Memilih Menu Buat Data Baru	Menampilkan form buat data baru	Tampil form Buat Data Baru		Berhasil
		Memilih Menu Lihat Data Peserta	Menampilkan form lihat data peserta	Tampil form lihat data peserta		Berhasil
		Memilih Menu Sistem	Tampil list menu strip	Tampil list menu strip sistem		Berhasil
		Memilih Menu Input Nilai	Menampilkan form input nilai	Tampil from lihat hasil fuzzy		Berhasil
		Memilih Menu Sistem Fuzzy	Menampilkan form sistem fuzzy	Tampil from lihat hasil fuzzy		Berhasil
		Memilih Menu Lihat Hasil	Menampilkan from lihat hasil fuzzy	Tampil from lihat hasil fuzzy		Berhasil

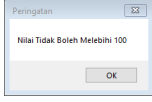


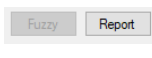

3	Form Buat Data Baru	Memilih Button Buat ID	Membuat id peserta secara otomatis	Tampil Id di textbox Id Peserta		Berhasil
		Memilih Button Refresh	Merifresh Tampilan datagridview agar sesuai dengan data terbaru	Refresh datagridview	Refresh datagridvi ew	Berhasil
		Memilih Button Simpan	Menyimpan data ke database dan menampilkan data yang terdapat di database ke datagridview	Data Tersimpan ke database dan data di database tampil ke datagridview	Data Tersimpan ke database dan data di database tampil ke datagridvi ew	Berhasil
		Memilih Button Clear	Menghapus seluruh isi textbox	Seluruh isi textbox terhapus	Seluruh isi textbox terhapus	Berhasil
		Memilih Button Keluar	Keluar dari form Buat data Baru	Kembali ke from menu	Kembali ke from menu	Berhasil
		Text Box Nama	Jika TextBox di masukan angka	Angka Tdk muncul	Angka Tidak Muncul	Berhasil
			Jika TextBox di masuk huruf	Huruf Muncul	Huruf Muncul	Berhasil

		TextBox NIK	Jika TextBox di masukan angka	Angka muncul	Angka Muncul	Berhasil
			Jika TextBox di masukan huruf	Huruf Tdk Muncul	Huruf Tidak Muncul	Berhasil
		TextBox Tanggal Lahir	Jika TextBox di klik	Tampil Kalender Untuk Memilih Tanggal		Berhasil
		ComboBo x Status	Jika TextBox di klik muncul pilihan kawin dan belum kawin	Angka Tdk muncul	Angka Tidak Muncul	Berhasil
			TextBox No Telepon hanya angka saja	Jika TextBox di masukan angka	Angka muncul	Angka Muncul
			Jika TextBox di masukan huruf	Huruf Tdk Muncul	Huruf Tidak Muncul	Berhasil
4	From Lihat Data Peserta	Memilih Button Refresh	Merfresh Tampilan datagridview agar sesuai dengan data terbaru	Refresh datagridview	Refresh datagridvi ew	Berhasil
		Memilih Button Hapus	Menghapus data yang berada di	Menghapus data di database serta datagridview	Menghapus s data di database serta	Berhasil

			database serta datagridview		datagridview	
		Memilih Button Update	Mengrubah data yang berada di database	Data di database Berubah	Data di database Berubah	Berhasil
		Memilih Button Kembali	Keluar dari form Lihat Data peserta	Kembali ke from menu	Kembali ke from menu	Berhasil
		Memilih Menu Edit	Menampilkan data di Datagridview ke textbox	Data tampil ke textbox	Data tampil ke textbox	Berhasil
		Memilih Menu Hapus	Menghapus data yang berada di database serta datagridview	Menghapus data di database serta datagridview	Menghapus data di database serta datagridview	Berhasil
		TextBox Nama hanya huruf saja	Jika TextBox di masukan angka	Angka Tdk muncul	Angka Tidak Muncul	Berhasil
			Jika TextBox di masuk huruf	Huruf Muncul	Huruf Muncul	Berhasil
		TextBox NIK hanya angka saja	Jika TextBox di masukan angka	Angka muncul	Angka Muncul	Berhasil
			Jika TextBox di masukan huruf	Huruf Tdk Muncul	Huruf Tidak Muncul	Berhasil

		TextBox Tanggal Lahir	Jika TextBox di klik	Tampil Kalender Untuk Memilih Tanggal		Berhasil
		ComboBo x Status	Jika TextBox di klik muncul pilihan kawin dan belum kawin	Angka Tdk muncul	Angka Tidak Muncul	Berhasil
		TextBox No Telepon hanya angka saja	Jika TextBox di masukan angka	Angka muncul	Angka Muncul	Berhasil
			Jika TextBox di masukan huruf	Huruf Tdk Muncul	Huruf Tidak Muncul	Berhasil
5	From Input Nilai	Memilih Button Refresh	Merfresh Tampilan datagridview agar sesuai dengan data terbaru	Refresh datagridview	Refresh datagridvi ew	Berhasil
		Memilih Button Simpan	Menyimpan data ke database dan menampilkan data yang terdapat di database ke datagridview	Data Tersimpan ke database dan data di database tampil ke datagridview	Data Tersimpan ke database dan data di database tampil ke datagridvi ew	Berhasil

		Memilih Button Kembali	Keluar dari form Input Nilai	Kembali ke from menu	Kembali ke from menu	Berhasil
		Memilih Menu Pilih Data	Menampilkan data di Datagridview ke textbox	Data tampil ke textbox	Data tampil ke textbox	Berhasil
		TextBox Matematika	Jika TextBox Nilai Matematika di isi >100	Tampil Massage Box Nilai Tidak Boleh Lebih dari 100		Berhasil
			Jika TextBox Nilai Matematika di isi <100	Barhasil memasukan nilai	Barhasil memasukan nilai	Berhasil
		TextBox Pengetahuan Umum	Jika TextBox Nilai Pengetahuan Umum di isi >100	Tampil Massage Box Nilai Tidak Boleh Lebih dari 100		Berhasil
			Jika TextBox Nilai Pengetahuan Umum di isi <100	Barhasil memasukan nilai	Barhasil memasukan nilai	Berhasil
		TextBox Kepribadian	Jika TextBox Nilai Kepribadian di isi >100	Tampil Massage Box Nilai Tidak Boleh Lebih dari 100		Berhasil
			Jika TextBox Nilai	Barhasil memasukan nilai	Barhasil memasukan nilai	Berhasil

			Kepribadian di isi <100			
		TextBox Peraktik	Jika TextBox Nilai Peraktik di isi >100	Tampil Massage Box Nilai Tidak Boleh Lebih dari 100		Berhasil
			Jika TextBox Nilai Peraktik di isi <100	Barhasil memasukan nilai	Barhasil memasukan nilai	Berhasil
6	From Sistem Fuzzy	Memilih Button Fuzzy	Menampilkan Hasil Perengkingan	Menampilkan Hasil Perengkingan		Berhasil
			Mengaktifkan Button Report			Berhasil
		Memilih Button Simpan	Menyimpan data ke database dan menampilkan data yang terdapat di database ke datagridview	Data Tersimpan ke database dan data di database tampil ke datagridview	Data Tersimpan ke database dan data di database tampil ke datagridview	Berhasil
		Memilih Button Report	Menampilkan laporan hasil perengkingan	Menampilkan laporan hasil perengkingan		Berhasil
		Memilih Button Kembali	Keluar dari form Sistem Fuzzy	Kembali ke from menu	Kembali ke from menu	Berhasil

		Memilih Menu Pilih Data	Menampilkan data di Datagridview ke textbox	Data tampil ke textbox	Data tampil ke textbox	Berhasil
7	Form Lihat Hasil	Memilih Button Kembali	Keluar dari form lihat hasil	Kembali ke from menu	Kembali ke from menu	Berhasil
8	Cari Data by Name	Memilih Button Cari	Mencari data di database sort by nama	Pencarian data sort by nama	Pencarian data sort by nama	Berhasil

4. KESIMPULAN

Berdasarkan hasil pembahasan beserta penelitian yang telah dilakukan maka dapat diambil kesimpulan, diantaranya sebagai berikut :

1. Dengan adanya aplikasi pembantu sistem pendukung keputusan penerimaan karyawan dengan metode *fuzzy logic* diharapkan dapat membantu pengguna dalam membuat keputusan yang berkualitas.
2. Sistem pendukung keputusan penerimaan karyawan menggunakan metode *fuzzy logic ini* dirancang dan dibangun untuk memperbaiki masalah dalam pengambilan keputusan dalam menerima karyawan yang lebih berkualitas.

5. REFERENSI

Agus Naba (2009) Tutorial Cepat & Mudah Fuzzy Logic dengan Matlab. CV ANDI OFFSET, Yogyakarta

Anharku (2003-2009): Flowchart. IlmuKomputer.com

Aron Naldo Ritonga, Sri Lestari (2012): Perancangan Sistem Pendukung Keputusan Pengangkatan Calon Kepala Sekolah Negeri Bandar Lampung Dengan Metode SAW. Jurnal Informatika, Vol. 12 No.2. Fakultas Ilmu Komputer Informatics & Business Institute Darmajaya Bandar Lampung.

Ayuni Putri Astari, Rachman Komarudin (2018): Sistem Pendukung Keputusan Pemilihan Karyawan Terbaik Dengan Metode Fuzzy Tahani. Jurnal Penelitian Ilmu Komputer, System Embedded & Logic STMIK Nusa Mandiri Jakarta

Denny Hermanto (2003-2006). Tutorial Pemrograman Fuzzy Logic. IlmuKomputer.Com.

Diana (2018): Metode & Aplikasi Sistem Pendukung Keputusan. Deepublish (Group Penerbit CV Budi Utama). Yogyakarta

Eliasta Kataren (2015): Pemanfaatan *Fuzzy Logic* Dalam Sistem Penerimaan Pegawai Baru. Vol. IV No 2 : 57-60 Jurnal Universitas Sumatera Utara. Medan.

en.wikipedia.org (2019): System context diagram

Hanif Al Fatta (2009): Rekayasa Sistem Pengenalan Wajah. CV ANDI OFFSET, Yogyakarta.

Jubilee Enterprise (2015): Pengenalan *Visual Studio* 2013. PT Elex Media Komputindo Kelompok Gramedia. Jakarta.

xvii

K Praditya (2016): Rancang Bangun Aplikasi Pengkajian Kariyawan Pada Perusahaan. Institut Bisnis & Informatika STIKOM. Surabaya.

Mainah, Dini Hamdani (2017): Analisis dan Perancangan Sistem Informasi Pembahasan SecaraPeraktis Dengan Contoh Kasus. CV BUDI UTAMA. Yoyakarta

Muhammad Wali (2017): Membangun Aplikasi Windows dengan Visual Basic.NET 2015 Teori dan Praktikum. Go Print. Banda Aceh.

Nidhra dan Donethi (2012): BlackBox And Whitebox Testing Techniques A-Literature Review.

Putra Candraguna (2019): Aplikasi Android Media Pembelajaran Tata Surya *Markerless Augmented Reality*. Skripsi Sekolah Tinggi Manajemen Informatika Indonesia Mandiri. Bandung.

Quadi dan Farooq. (2010): Evaluating Effectiveness of Software Testing Techniques With Emphasis on Enchancing Software Reliability. Vol 2.

Sri Mulyani (2016): Metode Analisis dan Perancangan Sistem. Abdi Sistematika. Bandung.

Tata Sutabari (2012): Analisa Sistem Informasi. CV ANDI OFFSET. Yogyakarta.

Wahana Komputer (2010): Panduan Belajar MySQL Database Server. PT TransMedia. Jakarta Selatan.