

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian-penelitian sebelumnya terkait *fuzzy logic* dan metode yang digunakan dapat dilihat pada tabel.

Tabel 2.1 Perbandingan Terhadap Penelitian Sebelumnya.

No	Nama	Judul Penelitian	Metode	Hasil
1	Supina Batubara (2017)	Analisis Perbandingan Metode Fuzzy Mamdani Dan Fuzzy Sugeno Untuk Penentuan Kualitas Cor Beton Instan	Metode Takagi Sugeno & Metode Mamdani	Lebih direkomendasikan metode Mamdani karena metode Mamdani hasil perhitungannya lebih mendekati hasil yang sebenarnya dibandingkan dengan sugeno. Adapun Inputannya berupa agregat halus kadar lumpur, agregat halus kadar air, agregat kasar kadar lumpur, dan agregat.
2	Slamet Budirahardjo dan Setyoningsih Wibowo (2017)	Data Mining dalam Kajian Kualitas Aspal Beton Menggunakan <i>Forward Selection</i> Berbasis Naïve Bayes	Forward Selection berbasis Naïve Bayes	Menerapkan ilmu komputer khususnya data mining didalam pengujian kualitas aspal beton sehingga dapat menentukan atribut atau parameter yang berpengaruh dalam uji kualitas aspal beton dan akurat dalam klasifikasi menentukan kualitas aspal beton

3	Gulam Abdul Malik dan Agus Maman Abadi (2017)	Menentukan Harga Beras Sesuai Mutu Kualitas Beras Dengan Logika Fuzzy Mamdani	Metode Mamdani	Membantu Penjual maupun pembeli dalam menentukan harga beras, dimana aplikasi logika fuzzy ini mengubah input kadar air, jumlah butir patah, dan jumlah butir menir sehingga menghasilkan output harga yang sesuai dengan kualitas beras.
4	Boy Fechera, Jaja Kustija, dan Siscka Elvyanti (2012)	Optimasi Penggunaan Membership Function Logika Fuzzy Pada Kasus Identifikasi Kualitas Minyak Transformator	Metode Mamdani	Hasil yang diperoleh dalam penelitian ini, bisa menjadi alternatif dalam pengecekan kualitas minyak transformator dan menggunakan fungsi keanggotaan triangular merupakan pilihan yang paling optimal. Dimana terdapat tiga inputan yaitu kadar asam, warna minyak, dan tegangan tembus.

Penelitian ini merujuk pada penelitian sebelumnya yang dilakukan oleh Slamet Budirahardjo dan Setyoningsih Wibowo (2017) yang membahas tentang penerapan ilmu komputer khususnya *data mining* yang menentukan atribut atau parameter yang berpengaruh dalam uji kualitas aspal beton dan akurat dalam klasifikasi menentukan kualitas aspal beton dengan metode *Forward Selection* berbasis Naïve Bayes. Sedangkan pada penelitian ini dirancang sebuah aplikasi untuk menentukan banyaknya campuran pada pengaspalan jalan raya sehingga memberikan kualitas yang baik menggunakan metode Mamdani dan PHP sebagai bahasa pemrogramannya.

Dalam penelitian ini akan dilakukan penelitian secara langsung dan melakukan identifikasi permasalahan yang terjadi sesuai dengan masalah

yang telah diuraikan sebelumnya. Berdasarkan identifikasi masalah diambil sebuah solusi yakni pembuatan aplikasi menentukan kualitas pengaspalan.

2.2 Gambaran Umum Objek Penelitian

CV Bumi Indah merupakan sebuah CV yang bergerak di bidang pembangunan infrastruktur jalan raya yang berpusat di pulau Sumba. Salah satu cabang dari CV ini terdapat di Kelurahan Penfui Kota Kupang, NTT yang merupakan cabang kelima yang telah beroperasi sejak juni 2016. CV ini menangani proyek di Pulau Timor. Kegiatan yang dilakukan sebelum melakukan pengaspalan jalan ialah melakukan pengujian pada material berupa agregat, aspal dan mencampur material tersebut dalam suhu panas tertentu. Pengujian dan pencampuran material tersebut dilakukan pada laboratorium milik CV Bumi Indah. Campuran material dihitung berdasarkan ketentuan menggunakan mesin AMP. Ketentuan material dalam campuran untuk pengaspalan menjadi kriteria yang mempengaruhi kualitas pengaspalan jalan. Proses yang terjadi dilapangan, perhitungan material yang dicampurkan kadang melebihi atau kurang dari kriteria yang ditentukan dikarenakan pekerja tidak dapat menentukan secara pasti kriteria perhitungan material campuran. Akibatnya pengaspalan tidak bertahan lama, menyebabkan kerusakan dan kualitasnya pun kurang baik.

2.3 Kecerdasan Buatan

Kecerdasan buatan berasal dari bahasa inggris "*Artificial Intelligence*" atau disingkat AI, yaitu *intelligence* adalah kata sifat yang berarti cerdas sedangkan *artificial* artinya buatan. Kecerdasan buatan yang dimaksud disini merujuk pada mesin yang mampu berpikir, menimbang tindakan yang akan diambil, dan mampu mengambil keputusan yang dilakukan oleh manusia (Mulyanto, et al., 2010).

2.4 Logika Fuzzy

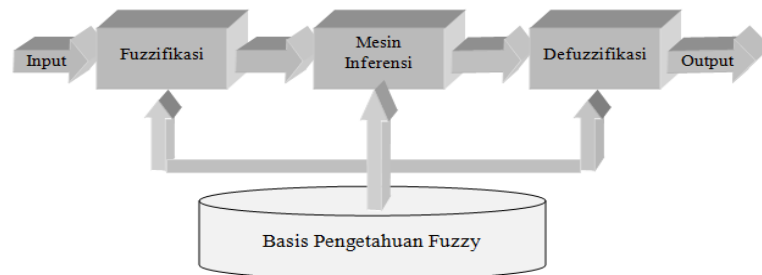
Sebelum munculnya teori logika fuzzy, dikenal sebuah logika tes (*Crisp Logic*) yang memiliki nilai benar atau salah secara tegas. Sebaliknya logika fuzzy merupakan sebuah logika yang memiliki kekaburan atau kesamaran (*fuzzynes*) antara benar dan salah. Dalam teori logika fuzzy sebuah nilai bisa bernilai benar dan salah secara bersamaan maupun berapa besar kebenaran dan kesalahan suatu nilai tergantung kepada bobot keanggotaan yang dimilikinya (Kusumadewi, 2004). Logika fuzzy merupakan salah satu komponen pembentuk *soft computing*. Teori logika fuzzy dikembangkan oleh Prof. Lotfi Zadeh sekitar tahun 1960-an dengan penentuan himpunan fuzzy. Dasar logika fuzzy ada teori himpunan fuzzy, pada teori himpunan fuzzy peranan derajat keanggotaan sebagai penentu keberadaan elemen dalam suatu himpunan sangatlah penting. Nilai keanggotaan atau derajat keanggotaan atau *membership function* menjadi ciri utama dari penalaran dengan logika fuzzy. Kelebihan dari teori logika fuzzy adalah kemampuan dalam proses penalaran secara bahasa (*linguistic reasoning*), sehingga dalam perancangannya tidak memerlukan persamaan matematik dari objek yang akan dikendalikan.

Alasan menggunakan logika fuzzy antara lain yaitu :

1. Konsep logika fuzzy mudah dimengerti. Konsep matematis yang mendasari penalaran fuzzy sangat sederhana dan mudah dimengerti.
2. Logika fuzzy sangat fleksibel.
3. Logika fuzzy mampu memodelakan fungsi – fungsi nonlinear yang sangat kompleks
4. Logika fuzzy dapat membangun dan mengaplikasikan pengalaman – pengalaman para pakar secara langsung tanpa melalui proses pelatihan.
5. Logika fuzzy dapat bekerjasama dengan teknik – teknik secara konvensional.
6. Logika fuzzy didasarkan pada bahasa alami.

2.4.1 Cara Kerja Logika Fuzzy

Untuk memahami cara kerja logika fuzzy, perhatikan struktur elemen dasar sistem *inferensi fuzzy* berikut :



Gambar 2.1 Cara kerja logika fuzzy (Mulyanto, 2011)

Keterangan :

- Basis pengetahuan *Fuzzy* adalah kumpulan *rule-rule fuzzy* dalam bentuk pernyataan *IF...THEN*.
- Fuzzifikasi adalah proses untuk mengubah *input* sistem yang mempunyai nilai tegas menjadi variabel *linguistik* menggunakan fungsi keanggotaan yang disimpan dalam basis pengetahuan fuzzy.
- Mesin inferensi adalah proses untuk mengubah *input fuzzy* menjadi *output fuzzy* dengan cara mengikuti aturan-aturan (*IF-THEN Rules*) yang telah ditetapkan pada basis pengetahuan fuzzy.
- Defuzzifikasi adalah mengubah *output fuzzy* yang diperoleh dari mesin inferensi menjadi nilai tegas menggunakan fungsi keanggotaan yang sesuai dengan saat dilakukan fuzzifikasi.

2.4.2 Variabel Linguistik Fuzzy

Variabel linguistik adalah sebuah variabel yang memiliki nilai berupa kata-kata dalam bahasa alamiah bukan angka.

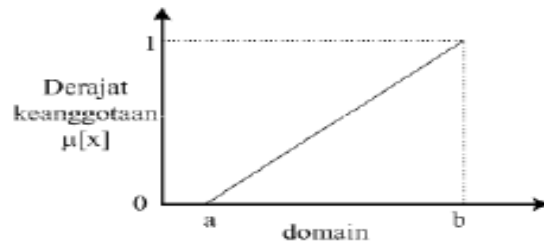
2.4.3 Fungsi Keanggotaan Fuzzy

Fungsi keanggotaan adalah suatu kurva yang menunjukkan pemetaan titik-titik *input* data ke dalam nilai keanggotaannya (derajat keanggotaannya) yang memiliki *interval* antara 0 sampai 1.

Ada beberapa fungsi yang bisa digunakan :

1. Representasi *Linear* ada dua kemungkinan himpunan *fuzzy linear* yaitu :

- a) Representasi *linear* naik, kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol (0) bergerak ke kanan menuju nilai domain yang memiliki derajat keanggotaan lebih tinggi.

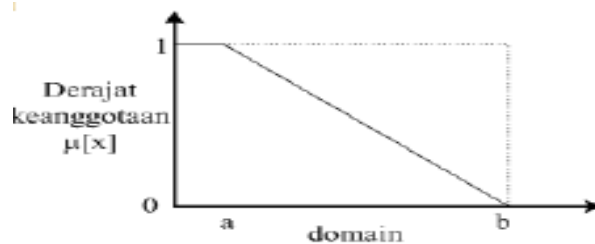


Gambar 2.2 Representasi *Linear* Naik

Fungsi keanggotaan :

$$\mu [x] = \begin{cases} 0; & x \leq a \\ (x-a) / (b-a) & a \leq x \leq b \dots\dots\dots [1] \\ 1; & x \geq b \end{cases}$$

- b) Representasi *linear* turun, merupakan kebalikan yang pertama. Garis lurus dimulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai domain yang memiliki derajat keanggotaan lebih rendah.

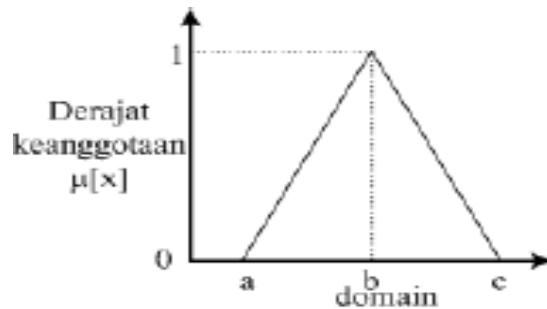


Gambar 2.3 Representasi *Linear* turun

Fungsi keanggotaan :

$$\mu[x] = \begin{cases} (b-x) / (b-a) & a \leq x \leq b \dots\dots\dots [2] \\ 0; & x \geq b \end{cases}$$

2. Representasi Kurva Segitiga



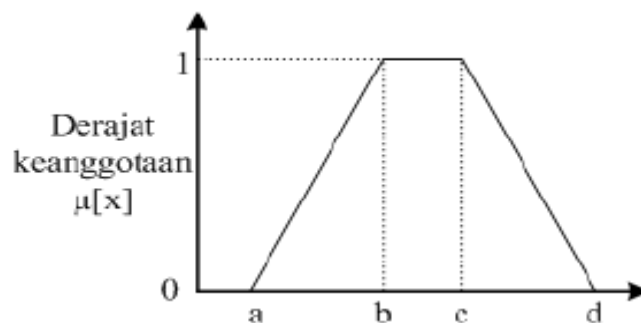
Gambar 2.4 Representasi kurva segitiga

Fungsi keanggotaan :

$$\mu(x) = \begin{cases} 0; & x \leq a \text{ atau } x \geq c \\ (x-a) / (b-a) & a \leq x \leq b \\ (b-x) / (c-b) & b \leq x \leq c \end{cases} \dots\dots\dots [3]$$

3. Representasi Kurva Trapezium

Kurva trapezium pada dasarnya sama seperti bentuk segitiga, hanya saja ada beberapa titik yang memiliki nilai keanggotaan satu.



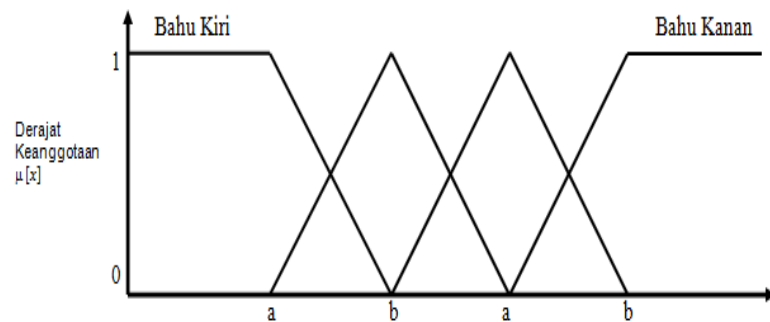
Gambar 2.5 Representasi Kurva Trapezium

Fungsi keanggotaan :

$$\mu[x] = \begin{cases} 0; & x \leq a \text{ atau } x \geq d \\ (x-a) / (b-a) & a \leq x \leq b \\ (d-x) / (d-c) & c \leq x \leq d \end{cases} \dots\dots\dots [4]$$

4. Representasi Kurva bentuk Bahu

Daerah yang terletak di tengah-tengah suatu variabel yang direpresentasikan dalam bentuk segitiga, pada sisi kanan dan kirinya akan naik dan turun. Himpunan *fuzzy* ‘bahu’, bukan segitiga digunakan untuk mengakhiri variabel suatu daerah *fuzzy*. Bahu kiri bergerak dari benar ke salah, demikian juga bahu kanan bergerak dari salah ke benar.



Gambar 2.6 Representasi Kurva Bahu.

Fungsi keanggotaan kurva bahu kiri :

$$\mu(x) = \begin{cases} 1; & x \leq a \\ (b-a) / (b-a) & a \leq x \leq b \\ 0; & x \geq b \end{cases} \dots\dots\dots [5]$$

Fungsi keanggotaan kurva bahu kiri :

$$\mu(x) = \begin{cases} 0; & x \leq a \\ \dots\dots\dots [6] \\ (x-a) / (b-a) & a \leq x \leq b \end{cases}$$

2.4.4 Sistem Inferensi *Fuzzy*

Sistem inferensi *fuzzy* merupakan sebuah kerangka kerja perhitungan berdasarkan pada konsep teori himpunan *fuzzy*, aturan *fuzzy if-then*, dan pemikiran *fuzzy* yang digunakan dalam penelitian ini.

2.4.4.1 Metode Mamdani

Metode mamdani paling sering digunakan dalam aplikasi-aplikasi karena strukturnya yang sederhana, yaitu menggunakan operasi *min-max* atau *max-product*. Metode ini diperkenalkan oleh Ebrahim Mamdani pada tahun 1975. Untuk mendapatkan *output*, diperlukan 4 tahapan berikut :

1) Fuzzifikasi

Pada Metode Mamdani, baik variabel input maupun variabel *output* dibagi menjadi satu atau lebih himpunan *fuzzy*

2) Pembentukan basis pengetahuan *fuzzy* (rule dalam bentuk *IF...THEN*).

3) Aplikasi fungsi implikasi menggunakan fungsi Min. Sedangkan komposisi aturan, apabila sistem terdiri dari beberapa aturan, maka inferensi diperoleh dari kumpulan dan korelasi antar aturan. Ada 3 metode yang digunakan dalam melakukan inferensi sistem *fuzzy*, yaitu:

a. Metode *Max (Maximum)*

Pada metode ini, solusi himpunan *fuzzy* diperoleh dengan cara mengambil nilai maksimum aturan, kemudian menggunakannya untuk memodifikasi daerah *fuzzy*, dan mengaplikasikannya ke *output* dengan menggunakan operator *OR (union)*. Jika semua proposisi telah dievaluasi, maka *output* akan berisi suatu himpunan *fuzzy* yang merefleksikan kontribusi dari tiap-tiap proposisi. Secara umum dapat dituliskan :

$$\mu_{sf}[xi] = \max (\mu_{sf}[xi] , \mu_{kf}[xi]) [7]$$

dengan :

$\mu_{sf}[xi]$ =nilai keanggotaan solusi *fuzzy* sampai aturan ke-i;

$\mu_{kf}[xi]$ =nilai keanggotaan konsekuen *fuzzy* aturan ke-i;

b. Metode *additive (Sum)*

Pada metode ini, solusi himpunan *fuzzy* diperoleh dengan cara melakukan *bounded-sum* terhadap semua *output* daerah *fuzzy*. Secara umum dituliskan:

$$\mu_{sf}[xi] = \min (1, \mu_{sf}[xi] + \mu_{kf}[xi]) \dots\dots\dots [8]$$

dengan :

$\mu_{sf}[xi]$ =nilai keanggotaan solusi *fuzzy* sampai aturan ke-i;

$\mu_{kf}[xi]$ =nilai keanggotaan konsekuen *fuzzy* aturan ke-i;

c. Metode Probabilistik OR (*Probor*)

Pada metode ini, solusi himpunan *fuzzy* diperoleh dengan cara melakukan product terhadap semua *output* daerah *fuzzy*. Secara umum dituliskan :

$$\mu_{sf}[xi] = (\mu_{sf}[xi] + \mu_{kf}[xi]) - (\mu_{sf}[xi] * \mu_{kf}[xi]) \dots\dots\dots [9]$$

dengan :

$\mu_{sf}[xi]$ =nilai keanggotaan solusi *fuzzy* sampai aturan ke-i;

$\mu_{kf}[xi]$ =nilai keanggotaan konsekuen *fuzzy* aturan ke-i;

4) Defuzzifikasi (Penegasan)

Input dari proses defuzzifikasi adalah suatu himpunan *fuzzy* yang diperoleh dari komposisi aturan-aturan *fuzzy*, sedangkan *output* yang dihasilkan merupakan suatu bilangan pada domain himpunan *fuzzy* tersebut. Sehingga jika diberikan suatu himpunan *fuzzy* dalam range tertentu, maka harus dapat diambil suatu nilai *crisp* tertentu sebagai *output*. Ada beberapa metode yang dipakai dalam defuzzifikasi:

a. Metode *Centroid*.

Pada metode ini penetapan nilai *crisp* dengan cara mengambil titik pusat daerah *fuzzy*.

b. Metode Bisektor.

Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai pada domain *fuzzy* yang memiliki nilai keanggotaan seperti dari jumlah total nilai keanggotaan pada daerah *fuzzy*.

c. Metode *Mean of Maximum* (MOM).

Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai rata-rata domain yang memiliki nilai keanggotaan maksimum.

d. Metode *Largest of Maximum* (LOM)

Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai terbesar dari domain yang memiliki nilai keanggotaan maksimum.

e. Metode *Smallest of Maksimum* (SOM).

Solusi *crisp* diperoleh dengan cara mengambil nilai terkecil dari domain yang memiliki nilai keanggotaan maksimum.

2.5 Pengertian Aplikasi

Menurut Jogiyanto (2005), perangkat lunak aplikasi adalah program yang ditulis dan ditulis dan diterjemahkan oleh *language software* untuk menyelesaikan suatu aplikasi tertentu.

2.6 Bahasa Pemrograman PHP

PHP (*Hypertext Preprocessor*) merupakan salah satu bahasa pemrograman yang paling banyak digunakan saat ini. Sudah jadi rahasia umum kalau PHP mampu membuat halaman dinamis, memanipulasi form, dan dapat dihubungkan dengan *database*. PHP merupakan salah satu *open source software*, yang dapat diartikan bahwa PHP dapat dimodifikasi, didistribusikan dan diintegrasikan dengan produk lain oleh penggunanya. PHP juga dapat berjalan dalam *web server* yang berbeda dan dalam sistem informasi yang berbeda pula (Sidik, 2012).

PHP (*Hypertext Preprocessor*) merupakan salah satu bahasa pemrograman *web* yang paling banyak digunakan saat ini. Sudah jadi rahasia umum kalau PHP mampu membuat halaman dinamis, memanipulasi form, dan dapat dihubungkan dengan *database*. PHP merupakan salah satu *open source software*, yang dapat diartikan bahwa PHP dapat dimodifikasi, didistribusikan, dan diintegrasikan dengan produk lain oleh penggunanya. PHP juga dapat berjalan dalam *web server* yang berbeda dan dalam sistem informasi yang berbeda pula (Sutarman, 2003).

Beberapa kelebihan PHP dari bahasa pemrograman *web*, antara lain:

- 1) Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.

- 2) *Web Server* yang mendukung PHP dapat ditemukan dimana-mana dari mulai apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah.
- 3) Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan *developer* yang siap membantu dalam pengembangan.
- 4) Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.

PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem.

2.7 Database MySQL

Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan menggunakan perangkat lunak untuk memanipulasinya. Sedangkan MySQL (*My Structure Query Language*) adalah pemrograman membuat *database* yang berbasis DOS bersifat *Open Source*, artinya siapa saja boleh menggunakan dan tidak dicekal. MySQL juga merupakan program pengakses *database* yang bersifat jaringan sehingga dapat digunakan untuk aplikasi *Multi User* (banyak pengguna) (Faisal, 2011).

Menurut Nugroho (2004), MySQL merupakan pemrograman atau sistem manajemen database (kumpulan data yang terstruktur) yang menggunakan basis bahasa SQL (*Structure Query Language*). MySQL merupakan sistem manajemen database yang dapat diandalkan dan penggunaannya mudah untuk dipahami. MySQL didesain untuk menangani database yang besar dengan cepat, memiliki tingkat keamanan dan konektivitas yang tinggi.

2.8 Material Pengaspalan

Proses pengerjaan pada konstruksi jalan raya terdiri dari campuran aspal keras dan agregat yang bergradasi menerus (*well graded*) dicampur, dihamparkan dan dipadatkan dalam keadaan panas pada suhu tertentu yang mampu memberikan sumbangan daya dukung yang terukur serta berfungsi sebagai lapisan kedap air yang dapat melindungi konstruksi dibawahnya.

Bahan utama penyusun perkerasan jalan adalah agregat dan aspal. Untuk mendapatkan hasil yang baik dan berkualitas dalam menghasilkan perkerasan jalan, maka bahan-bahan tersebut harus memiliki nilai yang tepat sehingga memberikan kualitas yang baik pula. Dari material tersebut akan dicampur dan dipanaskan pada sebuah mesin AMP dalam suhu yang telah ditentukan.

a) Agregat

Mutu dari agregat akan sangat menentukan mutu dari perkerasan yang akan dihasilkan. Pengawasan terhadap mutu agregat dapat dilakukan dengan pengujian di laboratorium. Agregat didefinisikan sebagai batu pecah, kerikil, pasir atau komposisi mineral lainnya, baik yang berupa hasil pengolahan (penyaringan, pemecahan) yang merupakan bahan baku utama konstruksi perkerasan jalan. Banyaknya agregat pada campuran mengisi 85 - 95% berat campuran atau 75 - 85% volume campuran (Bina Marga, 2010). Oleh karena itu perlu diperhatikan dengan baik kualitas agregat yang akan dipakai, yaitu memperhatikan sifat-sifat dari agregat tersebut seperti gradasi dan ukuran butir, kebersihan, bentuk dan tekstur permukaan, kekuatan dan porositas. Diperlukan pemeriksaan laboratorium mengenai mutu dari agregat itu sendiri. Dengan demikian agregat yang akan dipakai dalam penelitian dapat memenuhi sesuai dengan syarat yang ditentukan.

b) Aspal

Sulaksono (2001), aspal adalah sejenis mineral yang banyak digunakan untuk konstruksi jalan, khusus perkerasan lentur. Aspal merupakan material organik (*hydrocarbon*) yang kompleks yang dapat

diperoleh langsung dari alam atau dengan proses tertentu (*artificial*). Aspal adalah material penting dalam perkerasan lentur karena dapat merekatkan (bersifat sebagai perekat), mengisi rongga (sebagai *filler*) dan memiliki sifat kedap air (*waterproof*). Umumnya aspal terbagi atas bentuk cair, semipadat, dan padat pada suhu ruang (25°C). Aspal adalah material yang pada temperatur ruang berbentuk padat sampai agak padat, dan bersifat termoplastis (Sukirman, 2003). Jadi aspal akan mencair jika dipanaskan sampai temperatur tertentu, dan kembali membeku jika temperatur turun. Bersama dengan agregat, aspal merupakan material pembentuk campuran perkerasan jalan. Banyaknya aspal dalam campuran perkerasan berkisar antara 4-10% berdasarkan berat campuran, atau 10-15% berdasarkan volume campuran (Bina Marga, 2010).

c) Suhu

Temperatur suhu pemadatan yang digunakan pada viskositas kinematik 100 ± 10 centistokes, sehingga campuran antara agregat dan aspal tersebut dimasukkan ke dalam cetakan untuk dicampur 100°C - 170°C (Bina Marga, 2010).

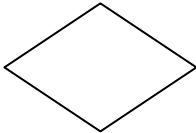
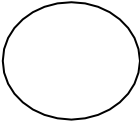
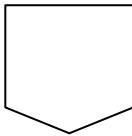
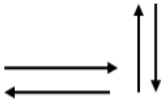
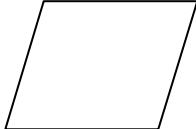
2.9 Diagram - Diagram Perancangan Sistem

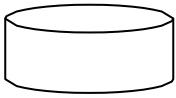
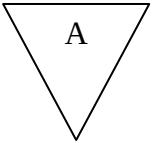
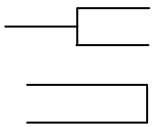
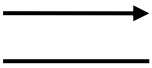
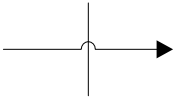
Diagram-diagram yang digunakan dalam perancangan sistem yang akan dibuat adalah sebagai berikut :

2.9.1 Diagram Alir (*Flowchart*)

Flowchart dibuat untuk menunjukkan alur kerja dari suatu sistem secara umum. Dengan adanya *flowchart*, dapat mengidentifikasi permasalahan yang ada pada sistem yang lama dan menganalisa kebutuhan dari sistem tersebut sebagai langkah awal dalam rancangan sistem baru yang baru yang akan dibentuk. Simbol-simbol dalam *flowchart* dapat dilihat pada tabel berikut :

Tabel 2.2 Simbol *Flowchart*

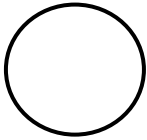
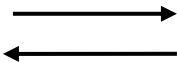
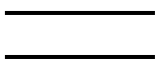
No	Simbol	Nama	Keterangan
1		Dokumen (<i>Document</i>)	Menunjukkan dokumen sebagai yang digunakan untuk merekam data terjadinya suatu transaksi.
2		Operasi Manual (<i>Manual Operation</i>)	Menunjukkan proses yang dikerjakan secara manual.
3		Decision	Menunjukkan pilihan yang akan dikerjakan atau keputusan yang harus dibuat dalam proses pengolahan data.
4		Conector (On-page connector)	Digunakan untuk penghubung dalam satu halaman.
5		Conector (Off-page connector)	Digunakan untuk penghubung berbeda halaman.
6		Garis aliran (<i>flow line</i>)	Menunjukkan arus data antar simbol/proses.
7		Catatan	Digunakan untuk menggambarkan catatan akuntansi yang digunakan untuk mencatat data yang direkam sebelumnya didalam dokumen atau formulir.

8		Penyimpanan/ <i>Storage</i>	Menunjukkan akses langsung perangkat penyimpanan/ <i>storage</i> pada disket.
9		<i>Off line storage</i>	Digunakan untuk menyimpan data secara manual dan sementara, jika “A” berarti disimpan menurut abjad, “N” berarti disimpan menurut nomor urut dan jika “T” berarti disimpan menurut kronologis atau menurut tanggal.
10		Keterangan atau komentar	Deskripsi proses atau komentar, untuk memperjelas pesan yang disampaikan dalam bagan alir.
11		Pertemuan garis alir	Menunjukkan dua garis alir bertemu dan salah satu garis mengikuti arus lainnya.
12		Persimpangan garis alir	Menunjukkan arah masing-masing garis, salah satu garis dibuat sedikit melengkung tepat pada persimpangan kedua garis tersebut.

2.9.2 Data Flow Diagram (DFD)

DFD merupakan suatu model logika atau proses yang dibuat untuk menggambarkan di mana asal data dan ke mana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut, dan interaksi antara data yang disimpan dan proses yang disimpan dan proses yang dikenakan pada data tersebut. Berikut tabel simbol-simbol yang digunakan di DFD:

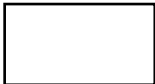
Tabel 2.3 Simbol *Data Flow Diagram* (DFD)

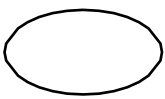
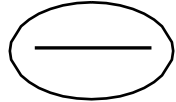
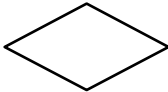
No	Simbol	Nama	Keterangan
1		Proses	Menunjukkan kegiatan atau kerja yang dilakukan oleh orang, mesin dan komputer.
2		<i>Eksternal Entity</i>	Menunjukkan entitas.
3		Simbol Arus Data (Data Flow)	Menunjukkan arus dari proses.
4		Data Store	Simpanan data pada komputer

2.9.3 Entity Relationship Diagram (ERD)

ERD merupakan diagram yang menggambarkan hubungan antara tabel dalam basis data yang direlasikan agar berfungsi optimal. Tujuan dari pembuatan ERD ini yaitu merancang suatu basis data untuk memperlihatkan hubungan atau relasi antar entitas atau objek yang terlihat beserta atributnya di dalam sistem secara keseluruhan. Berikut tabel simbol-simbol yang digunakan pada ERD:

Tabel 2.4 Simbol *Entity Relationship Diagram* (ERD)

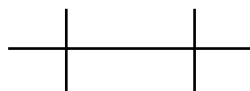
No	Simbol	Keterangan
1.		Entity Simbol yang menyatakan himpunan entitas ini bisa berupa : suatu elemen lingkungan, sumber daya, atau transaksi, yang begitu pentingnya bagi perusahaan sehingga didokumentasikan dengan data.

2.		Attribute Simbol terminal ini untuk menunjukkan nama-nama atribut yang ada pada entiti.
3.		Primary Key Attribute Simbol atribut yang digaris bawah, berfungsi sebagai <i>key</i> (kunci) di antara nama-nama atribut yang ada pada suatu <i>entity</i> .
4.		Relationship Simbol ini menyatakan relasi ini digunakan untuk menunjukkan hubungan yang ada antara entiti yang satu dengan entiti yang lainnya.
5.		Link Simbol berupa garis ini digunakan sebagai penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dengan atributnya.

2.9.4 Relasi

Relasi adalah hubungan antara satu tabel dengan tabel lainnya dalam basis data. Penggambaran relasi antar tabel menggunakan tanda panah. Berikut adalah jenis-jenis relasi antar tabel :

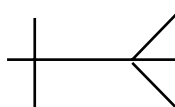
1. Relasi satu ke satu (*one to one*), satu *record* pada satu tabel hanya berhubungan dengan satu record pada tabel lainnya.



Atau 1: 1

Gambar 2.7 Relasi Satu Ke Satu

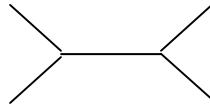
2. Relasi satu ke banyak (*one to many*), satu *record* pada satu tabel berhubungan dengan banyak record pada tabel lainnya.



Atau 1: n

Gambar 2.8 Relasi Satu Ke Banyak

3. Relasi banyak ke banyak (*many to many*), banyak *record* pada satu tabel berhubungan dengan banyak *record* pada tabel lainnya.



Atau n: m

Gambar 2.9 Relasi Banyak Ke Banyak

