**BAB II**

**TINJAUAN PUSTAKA**

1. **Penyakit Diabetes**

Penyakit diabetes atau biasa disebut diabetes mellitus (DM) merupakan penyakit gangguan metabolic menahun akibat pancreas tidak memproduksi cukup insulin atau tubuh tidak dapat menggunakan insulin yang di produksi secara efektif. Insulin sendiri adalah hormone yang mengatur keseimbangan kadar gula darah.[1] Berikut adalah penyebab pasien dapat mengidap atau mengalami diabetes :[1]

1. Disfungsi pankreas

Disfungsi pangkreas biasanya terjadi pada orang yang memiliki diabetes . Kondisi ini terjadi saat pankreas tidak mampu memproduksi [insulin](https://hellosehat.com/pusat-kesehatan/diabetes-kencing-manis/fakta-hormon-insulin) yang diperlukan untuk memecah makanan dan mengubahnya menjadi energi.

1. Keturunan

[Diabetes bersifat turun-temurun.](https://hellosehat.com/pusat-kesehatan/diabetes-kencing-manis/yang-harus-dilakukan-jika-anda-memiliki-keturunan-diabetes/) Seseorang berpeluang lebih besar untuk terkena penyakit tersebut jika orang tersebut memiliki orang tua, saudara kandung, atau kakek dan nenek yang mengalami atau mengidap penyakit diabetes.

1. Infeksi virus

Infeksi virus dapat menghancurkan atau merusak sel-sel dalam tubuh orang yang berusia muda. Hal ini juga dapat memicu respon autoimun yang berdampak buruk pada fungsi pankreas sehingga bisa menyebabkan diabetes. Beberapa infeksi virus yang dapat menyebabkan diabetes meliputi, coxsackievirus B, cytomegalovirus, adenovirus, rubella dan [gondong](https://hellosehat.com/hidup-sehat/beda-gondok-dan-gondongan/).

1. Obesitas

Obesitas merupakan faktor utama dapat menyebabkan orang yang berusia muda rentan terkena diabetes adalah[obesitas.](https://hellosehat.com/hidup-sehat/6-jenis-obesitas-yang-manakah-anda/) Obesitas atau berat badan berlebih akan mencegah tubuh merespon insulin.

1. Pola makan yang buruk

Pola makan yang buruk tidak hanya akan meningkatkan risiko terkena diabetes saja, namun masalah kesehatan lainnya seperti kenaikan berat badan, tekanan darah tinggi, dan bahkan penyakit jantung.

1. Rokok dan alkohol

Kecanduan alkohol dan kebiasaan merokok juga bisa jadi faktor yang menyebabkan anak muda berisiko tinggi menderita diabetes. Kebiasaan yang tidak sehat ini bisa mengganggu aktivitas metabolisme tubuh dan mempengaruhi produksi insulin.

1. Kurang Istirahat

Tidur atau istirahat yang cukup sangat penting untuk menjaga kesehatan. Pasalnya, kurang tidur membuat[ritme sirkandian](https://hellosehat.com/hidup-sehat/jam-biologis-manusia/) terganggu sehingga memicu perubahan hormon tubuh, termasuk insulin yang mengatur gula darah. Hal ini membuat kurang tidur akut bisa meningkatkan risiko seseorang terkena diabetes.

1. Males Ber-aktivitas

Kurangnya aktivitas fisik dapat menyebabkan perlambatan metabolisme dalam tubuh, sehingga mereka berisiko tinggi terkena diabetes di usia muda.

**2.2 *Data Mining***

Penambangan Data (*Data Mining*) adalah kegiatan mengekstraksi atau menambang pengetahuan dari data yang berjumlah besar, informasi dari data inilah yang nantinya sangat berguna untuk pengembangan. Data Mining adalah proses yang menggunakan teknik statistik, matematika, kecerdasan buatan,  *machine learning*  untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai database besar.[6]

Menurut Gartner Group, *Data Mining* adalah suatu proses menemukan hubungan yang berarti, pola, dan kecenderungan dengan memeriksa dalam sekumpulan besar data yang tersimpan dalam penyimpanan dengan menggunakan teknik pengenalan pola seperti teknik statistic dan matematika.[6]

*Data mining* memiliki tugas utama salah satunya yaitu klasifikasi (*classification*), klasifikasi bertujuan untuk mengklasifikasikan item data menjadi satu dari beberapa kelas standar. Sebagai contoh, suatu program email dapat mengklasifikasikan email yang sah dengan email spam. Beberapa algoritma klasifikasi antara lain pohon keputusan, algoritma Naïve Bayes dan logistik regresi. Data mining adalah suatu proses ekstraksi atau penggalian data yang belum diketahui sebelumnya, namun dapat dipahami dan berguna dari database yang besar serta digunakan untuk membuat suatu keputusan bisnis yang sangat penting.[2]

Berdasarkan definisi di atas dapat disimpulkan bahwa *data mining* adalah suatu proses mengekstrasi dan identifikasi dari suatu data dengan menggunakan teknik statistika, matematika, kecerdasan buatan dan *machine learning* yang dapat menjelaskan pola data dan membuat model prediksi dari data tersebut.

Data yang akan digunakan dalam *data mining* dibagi menjadi dua yaitu data *training* dan data *testing*. Data *training* adalah data yang digunakan untuk menghasilkan model klasifikasi. Sedangkan data *testing* adalah data yang digunakan untuk mengevaluasi model. Untuk menentukan banyak komposisi data *training* dan data *testing* digunakan teknik *cross-validation* atau yang sering disebut estimasi rotasi.

Istilah *data mining* dan *knowledge discovery in databases* (KDD) sering kali digunakan secara bergantian untuk menjelaskan proses penggalian informasi tersembunyi dalam suatu basis data yang besar. Sebenernya kedua istilah tersebut memiliki konsep yang berbeda, tetapi berkaitan satu sama lain. Dan salah satu tahapan dalam keseluruhan proses KDD adalah *data mining.* Proses *data mining* dapat dijelaskan sebagai berikut:[12]

1. Data Selection

Pemilihan (seleksi) data dari sekumpulan operasional perlu dilakukan sebelum tahap penggalian informasi.

1. Pre-Processing/ Cleaning

Sebelum proses *data mining* dapat dilaksanakan, perlu dilakukan proses *cleaning* pada data yang menjadi focus KDD.

1. Transformation

*Coding* adalah proses transformasi pada data yang telah dipilih, sehingga data tersebut sesuai untuk proses *data mining.*

1. Data Mining

*Data mining* adalah proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan algoritma atau metode tertentu.

1. Interpretation/ Evaluation

Pola informasi yang dihasilkan dari proses *data mining* perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan. Tahap ini mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesis yang ada sebelumnya.

Dalam *cross-validation* proporsi data training dan data testing tergantung pada analisis misalnya data dibagi menjadi 50% : 50% atau 2/3 untuk training dan 1/3 untuk testing. 2:1 (Han dan Kamber, 2012). *Data mining* adalah sebuah proses, sehingga dalam melakukan prosesnya harus sesuai prosedur yaitu proses CRISP-DM (Cross-Industry Standard Process for Data Mining). CRISP-DM memiliki tahapan *data mining* sebagai berikut:[6]

1. *Business Understanding*

Menentukan tujuan proyek dan kebutuhan secara detail dalam lingkup bisnis atau unit penelitian secara keseluruhan.

1. Data *Understanding*

Dilakukan pengumpulan data, kemudian menganalisa data serta evaluasi kualitas data.

1. Data *Preparation*

Persiapkan data mentah kemudian di seting untuk data akhir yang akan digunakan untuk fase selanjutnya, pilih kasus dan variabel yang diinginkan yang digunakan untuk menganalisa sesuai analisa masalah, lakukan transformasi pada variabel tertentu jika diinginkan, bersihkan data untuk alat pemodelan.

1. *Modelling*

Pada tahap ini, pilih dan terapkan teknik pemodelan yang tepat, lakukan pengaturan model untuk mengoptimalkan hasil, jika diperlukan lakukan ulang ke tahap persiapan sesuai dengan persyaratan spesifikasi dari teknik data mining tertentu

1. Evaluasi

Melakukan evaluasi satu atau lebih model, tentukan apakah model sudah mencapai tujuan yang diterapkan dalam tahap pertama, mengambil keputusan mengenai penggunaan hasil data mining.

1. *Deployment*

Memanfaatkan model yang telah dibuat, deployment yang sederhana adalah sampai menghasilkan laporan sedangkan deployment yang kompleks adalah melaksanakan model untuk proses data mining paralel pada departemen lain.

**2.3 *Binary Logistic Regression***

*Binary Logistic Regression* adalah sebuah pendekatan untuk membuat model prediksi seperti halnya regresi linear atau yang biasa disebut dengan istilah Ordinary Least Squares (OLS) regression. Perbedaannya adalah pada *Binary Logistic Regression*, peneliti memprediksi variabel terikat yang berskala dikotomi. Skala dikotomi yang dimaksud adalah skala data nominal dengan dua kategori, misalnya: Ya dan Tidak, Baik dan Buruk, Tinggi dan Rendah, atau Cepat dan Lambat. Model regresi logistik biner digunakan jika variabel responnya menghasilkan dua kategori bernilai 0 dan 1. [3]

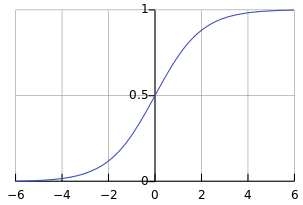
Regresi logistik biner merupakan suatu metode analisis data yang digunakan untuk mencari hubungan antara variabel respon (y) yang bersifat biner atau dikotomus dengan variabel prediktor (x) yang bersifat polikotomus. Keluaran dari variabel respon y terdiri dari 2 kategori yang biasanya dinotasikan dengan y = 1 (sukses) dan y = 0 (gagal). Model regresi logistik dibentuk dengan menyatakan nilai P(Y = 1│x) sebagai π (x) yang dinotasikan sebagai berikut: [9]

Suatu fungsi dari π(x) dicari dengan menggunakan transformasi logit, yaitu g(x) yang dapat dinyatakan sebagai berikut: [9]

Regresi logistik biner bisa diartikan sebagai suatu model pendekatan matematika yang digunakan untuk menggambarkan hubungan antara beberapa vaiabel penjelas dengan suatu variabel dikotomi. Variabel dikotomi mempunyai dua kemungkinan yang biasanya dinyatakan dengan 0 (gagal) dan 1 (sukses). Diberikan model sebagai berikut:[5]

Jika Y diberi 0 dan 1, maka

Persamaan umum untuk regresi logistik biner adalah



Gambar 2.1 Fungsi logistik regresi biner

Dengan model prediksi sebagai berikut:

Asumsi – asumsi dalam membangun model *logistic regression binary*:[8]

1. Sampel yang diperlukan dalam jumlah relatif besar, minimum dibutuhkan hingga 50 sampel data untuk sebuah variabel prediktor (bebas).
2. Variabel dependen harus bersifat dikotomi (2 variabel)
3. Kategori dalam variabel independent harus terpisah satu sama lain atau bersifat eksklusif
4. Tidak mengasumsikan hubungan linier antar variabel dependen dan independent..
5. Variabel independent tidak harus memiliki keragaman yang sama antar kelompok variabel.

**2.4 Python**

Python dikembangkan oleh Guido van Rossum pada tahun 1990 di CWI, Amsterdam sebagai kelanjutan dari bahasa pemrograman ABC. Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Python diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif.[13]

Menurut pengertian dari Python Software Foundation (2016), Python adalah bahasa pemrograman interpretatif, berorinetasi objek dan semantik yang dinamis. Python memiliki *high-level* struktur data, *dynamic typing* dan *dynamic binding*. Python memiliki sintaks sederhana dan mudah dipelajari untuk penekanan pada kemudahan membaca dan mengurangi biaya perbaikan program. Python mendukung modul dan paket untuk mendorong kemodularan program dan *code reuse*.[13]

Interpreter Python dan standard library-nya tersedia secara gratis untuk semua platform dan dapat secara bebas disebarkan. Bahasa yang digunakan dalam pengontrolan adalah bahasa Python. Sedangkan menurut Dr. Andrew N Harrington (2009), bahasa python adalah bahasa pemrograman yang memiliki banyak fungsi, interaktif, berorientasi objek dan merupakan bahasa pemrograman tingkat tinggi. Bahasa python adalah bahasa pemrograman formal dengan aturan-aturan dan format spesifiknya sendiri. Berikut beberapa fitur yang dimiliki bahasa pemrograman python:[13]

1. Memiliki kepustakaan yang luas, dalam distribusi python telah disediakan modul-modul siap pakai untuk berbagai keperluan.
2. Memiliki tata bahasa yang jernih dan mudah dipelajari.
3. Memiliki aturan layout kode sumber yang memudahkan pengecekan, pembacaan kembali, dan penulisan ulang kode sumber berorientasi objek.
4. Memiliki sistem pengelolaan memori otomatis (garbage collection, seperti java).
5. Modular, mudah dikembangkan dengan menciptakan modul-modul baru. Modul-modul tersebut dapat dibangun dengan bahasa python maupun C/C++.
6. Memiliki fasilitas pengumpulan sampah otomatis. Seperti halnya pada bahasa pemrograman java.

**2.5 Tkinter**

Menurut Python Software Foundation (2016), Tkinter (Tk *Interface*) adalah suatu pustaka GUI (alat penghubung pemakai grafis) *widget* standar pembuatan *Interface* python untuk TkGUI *Toolkit*. Tkinter merupakan antarmuka grafisnya dari TCL(*Tool Command Language*), yang memberikan kemudahan bagi para programmer dalam pembuatan programnya. Tkinter juga merupakan pustaka grafis yang dapat memberikan kemudahan dalam pembuatan program berbasis grafis. Setiap GUI Toolkit menyediakan *widget,* yaitu objek user interface seperti *button, scrollbar, listbox, checkbutton, radiobutton, label text* dan lain sebagainya. *Widget* mengkapsulasi detail implementasi dan untuk setiap *widget* telah di definisikan perilaku *defaultya* sehingga mempermudah pemrograman GUI.[16] Pustaka Tkinter merupakan pustaka standar yang digunakan untuk membuat aplikasi berbasis GUI pada bahasa pemrograman python. Di dalam membuat sebuah GUI biasanya terdapat berbagai objek – objek atau *widget*. *Widget* adalah objek yang ditampilkan dalam sebuah GUI untuk beriteraksi dengan user. Tkinter menyediakan suatu tombol *widget* (*Button*), suatu label *widget* (*Label*), dan sebagainya. Kebanyakan *widget* diletakan di suatu *root widget*. *Toplevel widget* adalah suatu *widget* yang khusus tanpa adanya induk. *Toplevel widget* diciptakan ketika memanggil Tkinter(). Untuk memeriksa pustaka Tkinter ini tersedia atau tidaknya dalam bahasa pemrograman python yang telah terpasang dalam komputer, dalam IDLE python tinggal mengetikan script sebagai berikut:

>>> **from** Tkinter import \*

>>>

Jika pustaka Tkinter tidak tersedia di dalam python maka akan terjadi error pada proses pemanggilan modul, agar Tkinter pada python tersedia maka python harus di pasang ulang atau di install. Jika modul Tkinter dipanggil namun belum tersedia di python maka akan menghasilkan output sebagai berikut :

>>> **from** Tkinter import \* Traceback (innermost last): File "<stdin>", line 1, in ?

File "/usr/lib/python1.5/lib-tk/Tkinter.py", line 8, in ?

**import** \_tkinter # If this fails your Python may not

be configured for Tk

ImportError: No module named \_tkinter

Di dalam pemrograman GUI, suatu objek di dalam sebuah tampilan merupakan bagian yang terpenting untuk memperindah tampilan, misalnya dalam sebuah tampilan tedapat label, teks, tombol, *list box*, *combo box*, dan lain – lain.

**2.6 *Dataset***

Penelitian ini menggunakan dataset mengenai penyakit diabetes yang diperoleh dari situs *Kaggle* atau *Machine Learning UCI* (Ronit, 2018)sebanyak 768 *record*.

Tabel 2.1. Deskripsi *dataset*

|  |  |  |  |
| --- | --- | --- | --- |
| **Label** | **Makna Label** | **Singkatan** | **Nilai** |
| Pregnancies | Jumlah kehamilan yang pernah di alami (khusus wanita) | Pregncies | Integer |
| Glucose | Kadar Glukosa | Gc | Integer |
| Diastolic blood pressure (mm Hg) | Tekanan Darah Diastolik  (mm Hg) | Dbp | Integer |
| Triceps skin fold thickness (mm) | Ketebalan lipatan kulit trisep (mm) | Sfold | Integer |
| Insulin | Insulin serum dalam 2 jam  (mu U / ml) | ISul | Integer |
| BMI | Indeks massa tubuh (berat dalam kg / (tinggi dalam m) ^ 2) | Bmi | Float |
|  |  |  |  |
| Diabetes pedigree function | Keturunan Diabetes | DPedigree | Float |
| Age (years) | Umur | Age | integer |
| Outcome | Output penyakit Diabetes | Result |  |

**2.7 Jupyter Notebook**

Menurut pengertian dari Jupyter Notebook App (2014) adalah aplikasi server-klien yang memungkinkan pengeditan dan menjalankan dokumen notebook melalui browser web. Aplikasi Notebook Jupyter dapat dijalankan pada desktop lokal yang tidak memerlukan akses internet (seperti yang dijelaskan dalam dokumen ini) atau dapat diinstal pada server jarak jauh dan diakses melalui internet. Selain menampilkan / mengedit / menjalankan dokumen notebook, Notebook Jupyter memiliki "*Dashboard*", "panel kontrol" yang memperlihatkan file-file lokal dan memungkinkan untuk membuka dokumen notebook atau mematikan kernel mereka.

* Kernel

Kernel notebook adalah "mesin komputasi" yang mengeksekusi kode yang terkandung dalam dokumen Notebook. Kernel ipython, dirujuk dalam panduan ini, mengeksekusi kode python. Kernel untuk banyak bahasa lain ada (kernel resmi). Saat Anda membuka dokumen Notebook, kernel yang terkait diluncurkan secara otomatis. Ketika notebook dijalankan (baik sel per sel atau dengan menu Cell -> Run All), kernel melakukan perhitungan dan menghasilkan hasilnya. Bergantung pada jenis perhitungan, kernel dapat mengkonsumsi CPU dan RAM yang signifikan.

* *Dashboard* Notebook

*Dashboard* Notebook adalah komponen yang ditampilkan pertama kali ketika Anda meluncurkan Notebook Jupyter. *Dashboard* Notebook terutama digunakan untuk membuka dokumen notebook, dan untuk mengelola kernel yang berjalan (memvisualisasikan dan mematikan). *Dashboard* Notebook memiliki fitur lain yang mirip dengan manajer file, yaitu menavigasi folder dan mengganti nama / menghapus file.

**2.8 *System Development Life Cycle***

*System development life cycle* adalah metode-metode dalam mengembangkan sebuah perangkat lunak. Proses tersebut terbagi menjadi analisa kebutuhan, desain, coding, pengujian, instalasi, serta pemeliharaan. Semua aktifitas tersebut dijalankan dengan cara yang berbeda-beda sesuai dengan kebutuhan klien. Cara-cara yang berbeda itu disebut sebagai *software* *development life cycle model*. SDLC (System Development Life Cycle) merupakan siklus yang menggambarkan perangkat lunak yang dibangun. SDLC juga merupakan pola yang diambil untuk mengembangkan sistem perangkat lunak, yang terdiri dari tahap-tahap:[10]

1. Perencanaan (*planning*)
2. Analisis (*analysis*)
3. Desain (*design*)
4. Implementasi / Koding (*implementation*),
5. Uji coba (*testing*)
6. Pengelolaan (*maintenance*).

Metode perancangan *software* berdasarkan teori model *waterfall* menurut Sommerville (2011) adalah tahapan utama yang langsungmencerminkan dasar pembangunan kegiatan, berikut ini adalahtahapannya:

1. *Requirements analysis and definition*, mengumpulkan apa yang dibutuhkan secara lengkap untuk kemudian dianalisis guna mendefinisiskan kebutuhan yang harus dipenuhi oleh program yang akan dibangun. Fase ini harus dikerjakan dengan lengkap untuk menghasilkan desain yang lengkap.
2. *System and software design*, setelah apa yang dibutuhkan telah selesai dikumpulkan dan sudah lengkap maka desain kemudian di kerjakan.
3. *Implementation and unit testing*, desain program diterjemahkan dalam kode-kode dengan menggunakan bahasa pemerograman yang sudah di tentukan. Program yang dibangun langsung diuji secara unit, apakah sudah bekerja dengan baik.
4. *Integration and system testing*, penyatuan unit-unit program untuk kemudian di uji secara keseluruhan (*system testing*).
5. *Operating and maintenance,* mengoperasikan program dilingkungannya dan melakukan pemeliharaan, seperti penyesuaian atau perubahan untuk adaptasi dengan situasi yang sebenarnya.

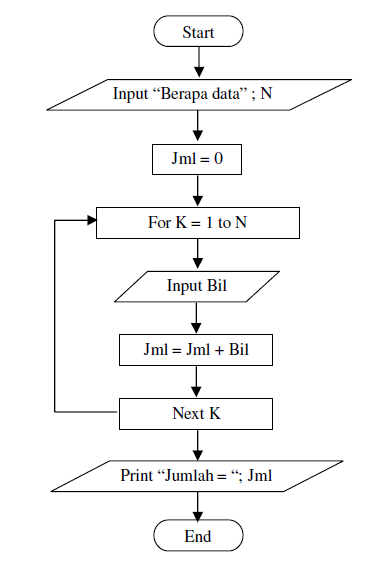
**2.9 *Flowchart***

*Flowchart* merupakan penggambaran secara grafik dari langkah-langkah dan urutan prosedur suatu program. Biasanya mempengaruhi penyelesaian masalah yang khusunya perlu dipelajari dan dievaluasi lebih lanjut.[4]

Tujuan Membuat *Flowchart* adalah :

* Menggambarkan suatu tahapan penyelesaian masalah
* Secara sederhana, terurai, rapi dan jelas
* Menggunakan simbol-simbol standar.

*Flowchart* di bedakan menjadi 5 jenis *flowchart,* antara lain *system flowchart, document flowchart, schematic flowchart, program flowchart, process flowchart*. Jenis *flowchart* yang digunakan *Program Flowchart*,bagan alir program (*program flowchart*) merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program. Bagan alir program dibuat dari derivikasi bagan alir sistem. Bagan alir program dapat terdiri dari dua macam, yaitu bagan alir logika program (*program logic flowhart*) dan bagan alir program komputer terinci (*detailed computer program flowchart*). Bagan alir logika program digunakan untuk menggambarkan tiap-tiap langkah di dalam program komputer secara logika. Bagan alat logika program ini dipersiapkan oleh analis sistem. Gambar berikut menunjukkan bagan alir logika program.



Gambar 2.2 Flowchart [4]

1. *Process Flowchart*

Bagan alir proses (process *flowchart*) merupakan bagan alir yang banyak digunakan di teknik industry. Bagan alir ini juga berguna bagi analis sistem untuk menggambarkan proses dalam suatu prosedur. Berikut ini merupakan notasi atau symbol-simbol yang digunakan dapat dibagi menjadi 3 (tiga) kelompok yaitu :[4]

1. *Flow Direction Symbols* (Simbol Penghubung/alur)

Simbol yang digunakan untuk menghubungkan antara symbol yang satu dengan yang lainnya. Simbol ini juga disebut connecting line, sebagai contoh:

Tabel 2.2 *Flow Direction Symbol* [4]

|  |  |  |  |
| --- | --- | --- | --- |
| No | Simbol | Nama | Keterangan |
| 1 |  | Arus / *Flow* | Untuk menyatakan arus suatu proses |

1. *Processing Symbols* (Simbol Proses)

Simbol yang menunjukkan jenis operasi pengolahan dalam suatu proses / prosedur. Simbol – simbol tersebut adalah :

Tabel 2.3 *Processing symbols* [4]

|  |  |  |  |
| --- | --- | --- | --- |
| No | Simbol | Nama | Keterangan |
| 1 |  | Proses | Sebuah fungsi pemro-sesan yang dilaksana-kan oleh komputer bia-sanya menghasilkan p-erubahan terhadap data atau informasi. |
| 2 |  | Terminal | Sebuah fungsi yang menyatakan permulaan atau akhir suatu progr-am. |

1. Input / Output Symbols (Simbol Input – output)

Symbol yang menunjukkan jenis peralatan yang digunakan sebagai media input atau output. Symbol – symbol tersebut adalah :

Tabel 2.4 *Input* – *Ouput Symbols* [4]

|  |  |  |  |
| --- | --- | --- | --- |
| No | Simbol | Nama | Keterangan |
| 1 |  | *Input* / *Output* | Untuk menyatakan pr-oses *input* dan *output* tanpa tergantung deng-an jenis peralatannya. |

**2.10 Aplikasi**

Aplikasi adalah satu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas dan bisa disebut sebagai sistem lengkap yang mengerjakan tugas spesifik. Aplikasi basis data terdiri atas sekumpulan menu, formulir, laporan dan program yang memenuhi kebutuhan suatu fungsional unit bisnis / organisasi / instansi.[11]

Aplikasi dapat diartikan juga sebagai program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu. Aplikasi merupakan software yang dirancang untuk penggunaan praktisi khusus, klasifikasi luas ini dapat dibagi menjadi 2 (dua) yaitu:[11]

1. Aplikasi paket, suatu program dengan dokumentasi tergabung yang dirancang untuk jenis masalah tertentu.
2. Aplikasi *software* spesialis, program dengan dokumentasi tergabung yang dirancang untuk menjalankan tugas tertentu.

**2.11 Confussion Matrix**

Untuk masalah klasifikasi biner, evaluasi diskriminasi dari solusi terbaik (optimal) selama pelatihan klasifikasi dapat di definisikan berdasarkan *confusion matrix* seperti yang akan di gambarkan pada tabel 2.5. Baris tabel mewakili kelas yang di prediksi, sedangkan kolom mewakili kelas yang sebenarnya. Dari *confusion matrix* ini menjelaskan bahwa tp dan tn menunjukkan jumlah *instance* positif dan negative yang di klasifikasikan dengan benar. Sementara itu, fp dan fn masing-masing menyatakan jumlah *instance* negatif dan positif yang salah di klasifikasi-kan. Berikut beberapa matriks yang umum digunakan : [14]

Tabel 2.5 Confusion Matrix dan Klasifikasi Biner

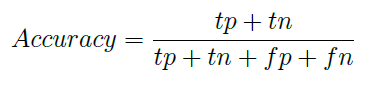


Penjelasan tabel 2.5 :

1. *True Positive* (TP) merupakan jumlah data positif yang terklasifikasi dengan benar oleh sistem.
2. *True Negative* (TN) merupakan jumlah data negatif yang terklasifikasi dengan benar oleh sistem.
3. False Positive (FP) merupakan jumlah data positif yang terklasifikasi salah oleh sistem.
4. False Negatives (FN) merupakan jumlah data negative namun terklasifikasi salah oleh sistem.

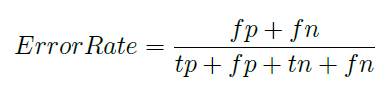
**2.11.1 Accuracy**

Secara umum, matriks akurasi mengukur rasio prediksi yang benar terhadap jumlah total *instance* yang dievaluasi.



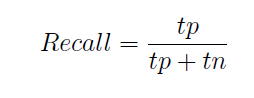
**2.11.2 Error Rate**

Kesalahan klarifikasi kesalahan mengukur rasio prediksi yang salah atas jumlah total kejadian yang dievaluasi.



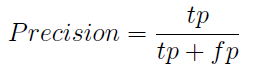
**2.11.3 Recall**

*True Positive Rate* biasa disebut nilai *recall* atau nilai sensitivitas. Perhitungan ini di awali dengan dari semua hasil positif seberapa sering hasil prediksi menghasilkan nilai benar. Recall digunakan untuk mengukur fraksi pola positif yang diklasifikasi-kan dengan benar.



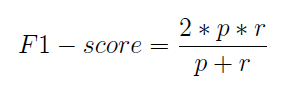
**2.11.4 Precision**

*Positive Predictive Value* atau biasa disebut nilai presisi atau nilai kepercayaan di dasarkan seberapa sering memprediksi secara besar dari semua kelas Presisi digunakan untuk mengukur pola positif yang diprediksi dengan benar dari total pola yang diprediksi di kelas positif.



**2.11.5 F1-Score**

Matriks ini mewakili rata-rata harmonic antara nilai recall dan presisi.



Keterangan pada tabel 2.11:

*p* = *Precision*

*r* = *Recall*

**2.12 Normalisasi Data**

Pada bidang analisis data, pengguna satuan ukuran dapat mempengaruhi hasil dari analisis yang didapatkan. Sebagai contoh, penggunaan ukuran meter pada pengukuran panjang akan memberikan hasil yang berbeda dengan pengukuran panjang menggunakan ukuran inci. Normalisasi data adalah metode statistika yang digunakan untuk mengatur nilai yang diukur pada suatu skala menjadi nilai pada skala berbeda yang lebih umum sehingga seluruh atribut dari data memiliki jangkauan nilai yang sama. Pada normalisasi, data ditransformasi sehingga memiliki jangkauan [-1,0,1,0] atau [0,0,1,0].[15]

Berikut penjelasan rumus yang dapat digunakan dalam normalisasi data : [15]

* *Min-max Normalization*

Normalisasi *min-max* pada awalnya mencari nilai maksimum dan minimum dari semua data yang tersedia atau ada. Normalisasi *min-max* melakukan transformasi linear terhadap data asli, dengan menggunakan persamaan berikut :

Penjelasan :

*v’i* = nilai hasil normalisasi dengan jangkauan [*new\_minA, new\_maxA*]

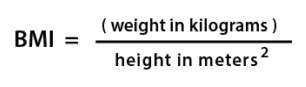
*vi* = nilai sebelum di normalisasi

*mixA* = nilai minimum dari data asli

*minA* = nilai maximum dari data asli

**2.13 BMI**

Pada dataset yang digunakan terdapat parameter yang akan dibutuhkan pada saat melakukan klasifikasi dan diperlukan perhitungan untuk mengisi variabel tersebut yaitu BMI. BMI adalah Indeks massa tubuh adalah cara yang baik untuk menilai apakah berat badan Anda sehat atau tidak. Variabel BMI ini wajib diisi dikarenakan menjadi salah satu variabel yang sangat diperlukan dalam proses klasifikasi. Berikut rumus umum untuk menghitung BMI yang akan digunakan pada pengisian kolom aplikasi pendeteksi diabetes :



Keterangan rumus 2.14:

Weight = Berat Badan ( kilogram )

Height = Tinggi Badan ( meter )

**2.14 Tabel Jurnal**

Tabel jurnal berisikan jurnal jurnal yang memiliki tema atau materi yang memiliki keterhubungan dengan materi data mining, klasifikasi, system prediksi, algoritma data mining, dan masalah mengenai penyakit.

Berikut merupakan daftar jurnal – jurnal yang disajikan ke dalam table seperti berikut :

Tabel 2.6 Telaah Jurnal

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Judul Jurnal** | **Tujuan** | **Metode** | **Kekurangan** |
| 1 | Sistem Prediksi Penyakit Diabetes Berbasis Decision Tree (Anik Andriani, 2013) | Membuat klasifikasi data diabetes dan menerapkannya dalam pembangunan sistem prediksi | Decision Tree ( C4.5 ) | Hasil akurasi Prediksi masih tergolong rendah (72%) |
| **No** | **Judul Jurnal** | **Tujuan** | **Metode** | **Kekurangan** |
|  |  |  |  | Klasifikasi tidak di implementasikan ke dalam bentuk GUI |
| 2 | Klasifikasi Penyakit Diabetes Mellitus TIPE 2  Dengan Metode Algoritma C4.5 (Moh.Jasri, 2017) | Membuat klasifikasi data-data penyakit yang terhubung dengan penyakit diabetes dan membuat prediksi penyakit diabetes | Decision Tree ( C4.5 ) |
|  |  |  |  | Hasil akurasi mencapai 70,32% dan tidak diimplementasikan ke dalam GUI |
| 3 | Implementasi algoritma decision tree C4.5 untuk prediksi penyakit diabetes (Noviandi, 2018) | Membuat model prediksi penyakit diabetes menggunakan algortima *decision tree* | *Decision Tree* |

Penelitian yang dilakukan oleh Anik Andriani pada tahun 2013 bertujuan untuk membuat klasifikasi data diabetes. Metode yang digunakan adalah metode *decision tree*. Kekurangan pada hasil penelitian ini adalah klasifikasi yang telah dibuat memiliki hasil akurasi yang belum menyentuh 80% atau sekitar 72%.

Lalu pada tahun 2017, Mohammad Jasri memiliki penelitian yang sejenis, bertujuan membuat klasifikasi data diabetes yang dihubungkan dengan penyakit-penyakit lain yang memiliki keterhubungan dengan penyakit diabetes. Kekurangan pada penelitian ini adalah klasifikasi yang dibuat tidak diimplementasikan kedalam GUI.

Lalu pada tahun 2018, Noviandi melakukan penelitian yang sejenis, yang bertujuan untuk membuat model prediksi penyakit diabetes. Kekurangan pada penelitian ini adalah tingkat akurasi yang belum mencapai tingkat yang lebih dari 80% dan belum diimplementasikan kedalam GUI, hanya sebatas model prediksi penyakit diabetes.

Penelitian yang penulis lakukan bertujuan untuk mengimplementasikan metode *data mining* dengan algoritma *logistic regression binary*  dalam mendeteksi penyakit diabetes secara dini dan efisien. Fitur yang dilakukan dalam penelitian ini adalah membuat aplikasi deteksi penyakit diabetes yang akurat, peningkatan akurasi, dan meng-implmentasikan metode *data mining* dalam GUI.

**2.15 BlackBox**

Pengujian sistem adalah pengujian program perangkat lunak yang lengkap dan terintegrasi. Perangkat lunak atau yang sering dikenal dengan sebutan *software* hanyalah satuan elemen dari sistem berbasis komputer yang lebih besar.[17] Biasanya, perangkat lunak dihubungkan dengan perangkat lunak dan perangkat keras lainnya.

Salah satu metode pengujian system yaitu *Blackbox.* Black Box Testing atau yang sering dikenal dengan sebutan pengujian fungsional merupakan metode pengujian Perangkat Lunak yang digunakan untuk menguji perangkat lunak tanpa mengetahui struktur internal kode atau Program. Dalam pengujian ini, tester menyadari apa yang harus dilakukan oleh program tetapi tidak memiliki pengetahuan tentang bagaimana melakukannya.[17]

* **Kelebihan Black Box Testing yaitu:**

1. Efisien untuk segmen kode besar
2. Akses kode tidak diperlukan
3. Pemisahan antara perspektif pengguna dan pengembang

* **Kelemahan Black Box Testing yaitu:**

1. Cakupan terbatas karena hanya sebagian kecil dari skenario pengujian yang dilakukan

**2.16 Balsamiq Mockup**

Balsamiq Mockups adalah alat desain antarmuka pengguna untuk membuat gambar rangka (juga disebut *mockup* atau *prototipe low-fidelity*). Aplikasi ini dapat digunakan untuk menghasilkan sketsa digital dari ide produk anda untuk memfasilitasi diskusi dan pemahaman sebelum kode apapun ditulis. Setiap file Balsamiq Mockups (dengan ekstensi file .bmpr) mewakili satu proyek dan dapat berisi banyak mockup dan gambar, yang disimpan bersama dalam satu file .bmpr.[18] Mockup untuk proyek yang berbeda harus dibuat dalam file Balsamiq Mockup yang terpisah. Untuk membuat proyek baru, pilih Project > New Project dari menu. Antarmuka pengguna Balsamiq Mockups terdiri dari lima area utama: *toolbar*, UI *library*, canvas, *navigator panel*, dan *property panel*. Masing-masing dijelaskan pada Gambar 2.3.



Gambar 2.3 Balsamiq Mockup[18]