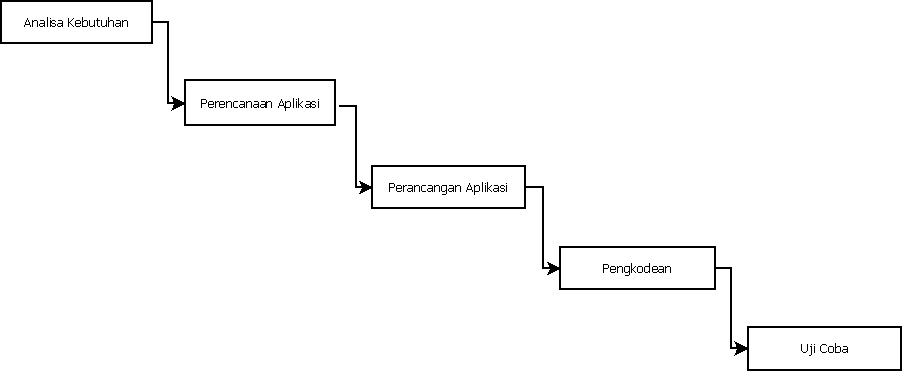
**BAB III**

**ANALISA DAN PERANCANGAN**

Metode perancangan aplikasi untuk masalah diabetes detection application menggunakan metodologi yaitu *Sytem Development Life Cycle* (SDLC) yang merupakan siklus hidup pengembangan sistem. Pada tahap ini memahami masalah yang ada pada perencanaan untuk proses klasifikasi menggunakan *Binary Logistic Regression,* kemudian merancanng struktur dan tampilan system yang dilakukan untuk memberi gambaran awal mengenai sistem yang akan di bangun. Selanjutnya pada tahap pembuatan perencanaan yang tekah dilakukan lalu di implementasikan dengan melakukan penulisan *source code* menggukanan Jupyter Notebook dengan *stack*  *library* yang ditentukan menggunakan bahasa pemrograman python sehingga menjadi sebuah sistem dengan dasar bantuan literature yang sudah dikumpulkan sebelumnya, Setelah itu dilakukan pengujian terhadap sistem yang telah jadi dengan menggunakan data-data yang telah ada. Tahapan metode dapat di-lihat seperti pada gambar 3.1.



Gambar 3.1: Tahapan Metode Aplikasi

**3.1 Perencanaan Aplikasi**

Dalam penelitian ini berisi perencanaan aplikasi yang dapat melakukan implementasi metode *Binary Logistic Regression,* karena algoritma *Binary Logistic Regression* mempunyai kemampuan mengklasifikasikan data dalam jumlah yanh cukup besar dengan waktu komputasi yan relatif cepat dan efisien. Sehingga dapat digunakan dalam deteksi penyakit diabetes dan pemilihan variable independen adalah salah satu factor yang paling penting dapat mempengaruhi kinerja klasifikasi. Data yang digunakan adalah Diabetes UCI dataset yang mengklasifikasi-kan 2 kondisi yaitu “0” untuk hasil output negatif dan “1” untuk hasil output positif yang menggunakan bahasa pemrograman python.

**3.2 Analisa Kebutuhan**

Analisa kebutuhan sangat diperlukan dalam pembuatan program aplikasi, agar saat program dijalankan dapat bekerja dengan baik. Berikut beberapa kebutuhan perangkat keras (*hardware*), perangkat lunak (*software*) dan *dataset*.

**3.2.1 Perangkat Keras**

Perangkat keras(*hardware*) adalah komponen pada komputer yang dapat terlihat dan disentuh secara fisik. Selain itu, perangkat keras juga digunakan untuk menguji coba aplikasi. Bentuk dari komputer atau laptop dapat kita sebut sebagai hardware atau perangkat keras, berikut spesifikasi yang digunakan dalam pembuatan program aplikasi ini.

* Laptop Lenovo B475, dengan spesifikasi ditabel 3.1.

Tabel 3.1 : Spesifikasi Perangkat Keras

|  |  |
| --- | --- |
| Prosesor | AMD A6-3400M APU (Llano) |
| Memori | DDR3 4 GB |
| Hardisk | 500 GB |
| Graphics | AMD Radeon HD 6520G 512 MB. |

**3.2.2 Perangkat Lunak**

Perangkat lunak (*software*) adalah data-data yang terdapat pada sebuah komputer yang diformat kemudian disimpan secara digital. Bisa dibilang bahwa software merupakan komponen yang tidak terlihat secara fisik, tetapi terdapat dalam sebuah komputer. Perangkat lunak yang digunakan dalam pembuatan aplikasi ini.

* Windows 7 Ultimate 64 Bit

Sistem Operasi pada komputer yang digunakan

* Python 3.6

Bahasa Pemrograman yang digunakan

* Jupyter Notebook

Editor penulisan code program Python 3.6

**3.2.3 Dataset**

Data penelitian yang digunakan bersumber dari portal <https://www.kaggle.com/> atau *Machine Learning UCI*. Sebanyak 768 baris dan 9 kolom yang disebut atribut *( pregnancies, Glucose, Diastolic blood pressure (mm Hg), Triceps skin fold thickness (mm), Insulin, BMI, Diabetes pedigree function, Age, Outcome )*.

Tabel 3.2 Dataset

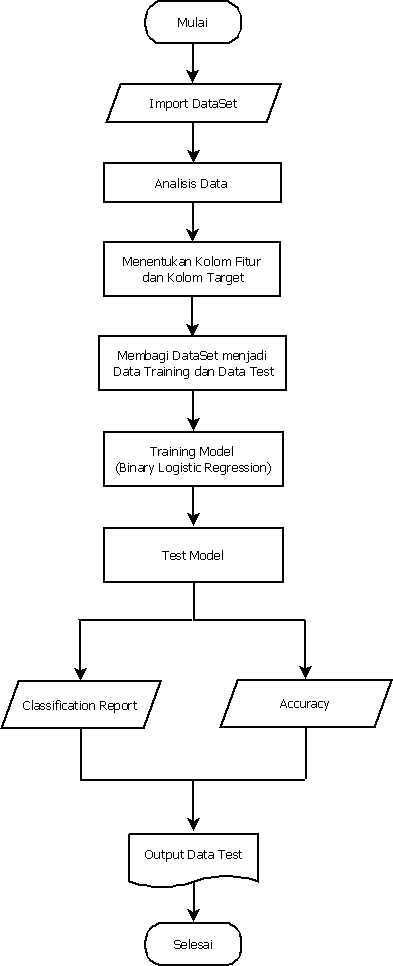
|  |  |  |  |
| --- | --- | --- | --- |
| **Label** | **Makna Label** | **Singkatan** | **Nilai** |
| Pregnancies | Jumlah kehamilan yang pernah di alami (khusus wanita) | Pregncies | Integer |
| Glucose | Kadar Glukosa | Gc | Integer |
| Diastolic blood pressure (mm Hg) | Tekanan Darah Diastolik  (mm 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 |  |

**3.3 Perancangan Aplikasi**

Tahap ini merancang struktur dan tampilan system yang dilakukan untuk memberi gambaran awal mengenai system yang dibuat.

**3.3.1 Flowchart**

Tahap pertama yang dilakukan adalah mengumpulkan dataset yang terdiri dari data training dan data test. Setelah mengumpulkan dataset, kemudian dilakukan pembagian atribut. Atribut terbagi menjadi, yaitu terdapat fitur dan target. Seperti pada gambar 3.2.



Gambar 3.2 Flowchart

Gambar 3.2 diatas menjelaskan tentang *flowchart* untuk membuat klasifikasi penyakit diabetes yang terdiri dari 9 tahap.

Tahap pertama, dimulai dengan mengidentifikasi masalah, menetapkan tujuan masalah, mengumpulkan dataset mengenai penyakit diabetes dari kaggle.com dan di import kedalam program.

Tahap kedua, menganalisa dataset yang akan digunakan dengan cara memilih data yang lengkap atau tidak mengandung *missing value* dan *noise.*

Tahap ketiga, data klasifikasi dibagi menjadi dua data yang terdiri dari data *training* dan data *testing* yang masing masing 80% dan 20% dengan jumlah data *training* sebanyak 614 data dan data *testing* sebanyak 154 data*.*

Tahap keempat, melakukan proses *training* pada data yang telah dibagi menjadi data khusus *training.*

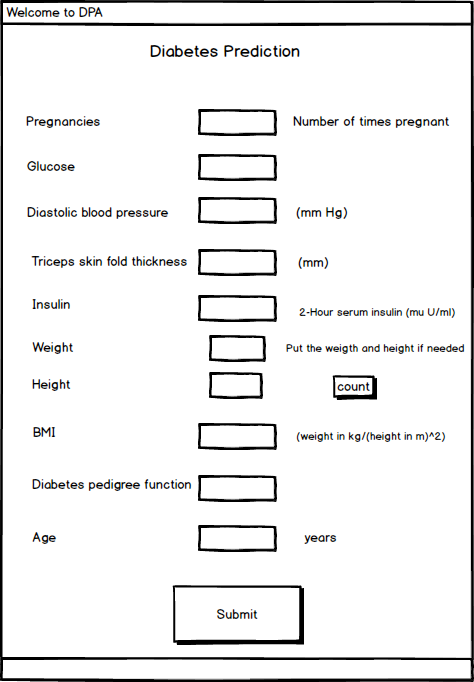
Tahap kelima, melakukan *test model* yang mana akan terbagi menjadi dua proses yaitu proses *classification report* dan *accuracy.* Dan tahap keenam, proses output *testing* atau mencoba klasifikasi penyakit diabetes dengan data *testing* lainnya.

**3.3.2 Perancangan Tampilan**

Rancangan tampilan aplikasi ini menjelaskan bagaimana suatu aplikasi akan tampil pada layar, tata letak, element dan fitur yang ada pada aplikasi. Rancangan tampilan halaman tersebut menjadi konsep dari aplikasi sehingga akan terlihat unsur yang saling terkait.

**3.3.2.1 Rancangan Tampilan Halaman Utama**

Pada rancangan ini adalah tampilan halaman utama dari aplikasi yang tampak setelah masuk ke dalam aplikasi. Rancangan tampilan dapat dilihat pada Gambar 3.3.

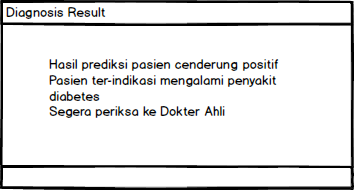


Gambar 3.3 Rancangan Tampilan Utama

Pada rancangan halaman utama langsung diarahkan pada form. Pada aplikasi ini terdapat satu buah button untuk mengolah data dari beberapa inputan yang disediakan. Pada baris pertama, diperlukan data input berupa jumlah kehamilan yang pernah dialami oleh pasien perempuan. Pada baris kedua, diperlukan data input berupa jumlah kadar glukosa yang berasa dari hasil laboratorium. Pada baris ketiga, diperlukan data inputan berupa tekanan darah atau tensi. Pada baris keempat, diperlukan data input berupa ketebalan lipatan kulit trisep dalam ukuran millimeter. Pada baris kelima, diperlukan data input berupa insulin yang mana berasal dari hasil suntikan insulin pada tubuh pasien selama dua jam setelah dilakukan penyuntikan insulin. Pada baris keenam, diperlukan data input berupa indeks massa tubuh yang diberikan rumus umumnya pada bab dua. Pada baris ketujuh, diperlukan data inputan berupa data angka biner 0 atau 1 yang memiliki arti 0 yaitu mempunyai keturunan diabetes dan untuk 1 memiliki arti sebaliknya. Dan pada baris terakhir, diperlukan data inputan berupa umur dalam hitungan tahun.

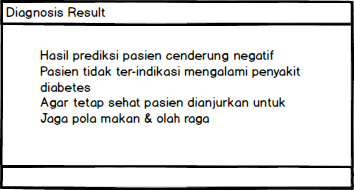
**3.2.2.2 Rancangan Tampilan Hasil Output**

Pada rancangan ini adalah tampilan hasil diagnosa dari aplikasi yang tampak setelah proses pengisian *form*. Rancangan tampilan dapat dilihat pada 3.4 untuk hasil positif dan Gambar 3.5 untuk hasil negatif.



Gambar 3.4 Rancangan tampilan hasil positif

Pada rancangan halaman hasil akan menampilkan informasi pada Gambar 3.4, jika pasien dalam kondisi tidak sehat atau ter-indikasi.



Gambar 3.5 Rancangan tampilan hasil negatif

Pada rancangan halaman hasil akan menampilkan informasi pada Gambar 3.4 jika pasien dalam kondisi tidak ter-indikasi.

**3.4 PENGKODEAN**

**3.4.1 Import Library**

*Library* adalah koleksi dari rutin-rutin program yang digunakan untuk membangun dan mengembangkan perangkat lunak. *Library,* umumnya mengandung kode program dan data pembantu (banyak programmer menyebut-nya sebagai *helper*), yang menyediakan layanan-layanan kepada program-program independen.

* Import *library* pandas



Gambar 3.6 Import *library* pandas

Pandas adalah *library* berlisensi *open source* yang menyediakan struktur data dan analisis data yang mudah digunakan dan berkinerja tinggi untuk bahasa pemrograman python. Sangat kita perlukan untuk membersihkan data mentah ke dalam sebuah bentuk yang cocok untuk analisis (yaitu tabel seperti dataset). Penting untuk dicatat di sini bahwa pandas melakukan tugas penting seperti menyelaraskan data untuk perbandingan dan penggabungan set data, penanganan data yang hilang, dll, itu telah menjadi sebuah library nyata untuk pemrosesan data tingkat tinggi dalam python (yaitu statistik). Disini pandas didefinisikan sebagai pd sehingga pemanggilan *library* tersebut dapat dilakukan dengan memanggil pd lalu di ikutiperintahnya.

* Import *library* Matplotlib



Gambar 3.7 Import *library* matplotlib

Fungsi yang paling penting dalam matplotlib adalah plot, yang memungkinkan untuk mem-plot data 2D. Disini matplotlib di definisikan sebagai plt sehingga pemanggilan *library* dapat dilakukan plt kemudian diikuti perintahnya.

* Import *library* numpy



Gambar 3.8 Import *library* numpy

Numpy adalah *library* inti untuk komputasi ilmiah dengan python. Ini menyediakan objek array multidimensional berkinerja tinggi dan alat untuk bekerja dengan array ini. Disini numpy di definisikan sebagai np sehingga pemanggilan *library* dapat dilakukan np lalu diikuti perintahnya.

* Import *library* sklearn



Gambar 3.9 Import *library* sklearn

Scikit learn atau biasa disingkat menjadi sklearn, merupakan *library* yang paling sering digunakan membuat sebuah *data mining*, *machine learning* dll, karena di dalamnya terdapat model-model algoritma yang dapat di panggil sesuai kebutuhan. Berikut beberapa kebutuhan sklearn dalam pembuatan program sesuai fungsinya.

* Import *library* *metrics*



Gambar 3.10 import *library* metrics

Gambar 3.11 merupakan sebuah pengukuran kuantitatif terhadap sesuatu yang dijadikan sebagai atribut dalam menentukan kualitas yang dalam program dibutuhkan pengukuran kualitas suatu prediksi.

* Import *library train test split*



Gambar 3.11 import *train\_test\_split*

Gambar 3.12 adalah pemanggilan *library* untuk membagi data *training* dan data *test* secara acak, namun bisa juga peneliti yang menentukan berapabanyak persentase data *training* atau data *test*.

**3.4.2 Import Dataset**

Dataset merupakan data mentah yang sangat digunakan pada proses metode *data mining* yang akan digunakan sebagai bahan-bahan untuk membuat pemodelan. Berikut gambar berupa *listing code* untuk memanggil atau import dataset ( dataset diabetes ) pada program python :



Gambar 3.12 import atau memuat dataset

Dataframe adalah struktur data pandas yaitu sebuah data yang berisi koleksi kolom dengan nama dan jenisnya. Pemanggilan dataset diabetes diberi nama diabetes yang ber-ekstensi csv dan akan disimpan oleh pandas yang didefinisikan sebagai pd, dan ditampung oleh variabel df.

**3.4.3 Menentukan Variabel Kolom Fitur dan Target**

**3.4.3.1 Kolom Fitur**

Sebelum menentukan variabel, sebaiknya tentukan dulu fitur yang digunakan dalam pengelompokkan yang mempengaruhi penyakit diabetes agar mendapatkan hasil yang maksimal dengan metode Binary Logistic Regression. Penggunaan fitur pada program ini adalah fitur 1 – 8 kecuali fitur ke 9 (outcome) karena fitur tersebut merupakan output akhir yang tidak dapat dihitung menggunakan metode Binary Logistic Regression. Dengan adanya hal seperti itu total fitur yang digunakan peneliti hanya 8 fitur agar mencapai hasil yang maksimal.



Gambar 3.13 Kolom Fitur

Dalam proses implementasi metode Binary Logistic Regression untuk masalah diabetes detection menggunakan bahasa pemrograman python hanya menggunakan 8 fitur yang terdiri dari Gambar 3.13.

**3.4.3.2 Kolom Target**



Gambar 3.14 Kolom Target

Gambar 3.14 terdapat dua kategori utama yaitu 0 dan 1. untuk 0 tidak terindikasi diabetes dan 1 terindikasi penyakit diabetes. Untuk kategori 1 sebanyak 500 data dan kategori 0 sebanyak 268 data.

**3.4.3.3 Menentukan Variabel**

****

Gambar 3.15 definisi variabel x



Gambar 3.16 definisi variable y

Gambar 3.15 dan Gambar 3.16 ditentukan terlebih dahulu nama variable agar mempermudah saat pemanggilan fungsinya, biasanya variabel ditentukan berdasarkan jenis atau tipe datanya. Pada program ini data target diberi variabel y sebagai nilai data sakit dan data sehat, sedangkan data fitur diberi variabel x sebagai ciri-ciri dari sakit diabetes.

**3.4.4 Data Training dan Data Test**

Setelah menentukan variabel, selanjutnya menentukan pembagian data *training* dan data *testing*. Data *training* adalah sebuah data yang digunakan machine learning untuk mempelajari fitur dan target, sedangkan data *testing* adalah data yang digunakan machine learning untuk menguji sistem menentukan ter-indikasi atau tidak ter-indikasi sesuai dengan dataset.

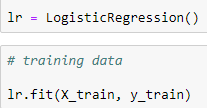


Gambar 3.17 Data Training dan Data Test

Pada gambar pembagian data training dan data test terjadi secara acak karena default pembagian terjadi secara random diantara lebih dari 0.0(0%) sampai kurang 1.0(100%), namun peneliti juga bisa menentukan berapa banyaknya data test dan data training yang ingin digunakan.“test\_size=0.2” dan “random\_state=0” yang di maksud disini adalah agar program saat di run data yang dibagi oleh train\_test\_split selalu sama(tidak berubah), sehingga nilai awal pembagian data diambil dari data ke-0.

**3.4.5 Metode Training Model dan Test Model**

Setelah menentukan data *training* dan *data testing,*  metode *training model* dan *testing model* akan dilakukan untuk melatih model dengan menggunakan algoritma yang akan dipilih. Berikut contoh pemanggilan *library* algoritma yang digunakan.



Gambar 3.18 Training Model dan Test Model

Metode yang dilakukan untuk pengelompokkan data training sebagai latihan machine learning dan pengelompokkan data test sebagai menguji machine learning adalah metode Binary Logistic Regression. Berikut cara penyelesaian metode Binary Logistic Regression pada persamaan 2.3.

Penjelasan :

Y = Dependent Variables

= Coefficient of Variables

= Independent Variables

= Constant

* Flowchart metode *Binary Logistic Regression*



Gambar 3.19 Flowchart metode *binary logistic regression*

* Pseudocode metode *binary logistic regression*



Gambar 3.20 Pseudocode algoritma *logistic regression*

* Proses algoritma *logistic regression binary*

1. Tentukan y sebagai data independen (fitur)
2. Tentukan a dan b sebagai data dependen (target)
3. Hitung regression koefisien dari data independen
4. Cari hubungan antara data independen dan data dependen
5. Ulangi langkah ke-3 sampai data dapat memprediksi data baru

**3.4.6 Menentukan Hasil prediksi**

Menentukan hasil prediksi dilakukan untuk melihat hasil prediksi dari model yang telah dibuat, dan untuk mengetahui tingkat akurasi yang dihasilkan setelah diterapkan algoritmayang digunakan. Berikut gambar 3.21 yang menjelaskan *lsiting code* untuk menentukan atau mengetahui hasil akurasi yang dicapai.

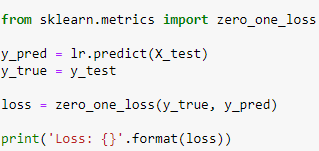


Gambar 3.21 Menentukan hasil akurasi

Gambar 3.21 ini merupakan penentuan hasil akurasi, hasil akurasi sangat dipengaruhi oleh dataset yang baik dan rapih, jumlah dataset pun bisa mempengaruhi hasil akurasi.

**3.4.7 Menentukan Hasil *Error/Loss***

Menentukan hasil *error/loss* berguna untuk membandingkan dengan hasil akurasi yang dicapai, sudah dipastikan hasil *error/loss* jika dijumlahkan dengan hasil akurasi akan berjumlah 100%. Berikut *listing code* untuk melihat hasil *error/loss* pada gambar 3.22.



Gambar 3.22 Menentukan *Loss/Error*

Gambar 3.22 merupakan penentuan hasil *error* atau *loss* pada model yang dibuat.

**3.4.8 Menentukan Hasil Klasifikasi**

Hasil klasifikasi diperlukan pada metode *data mining* untuk melihat hasil akurasi yang didapat secara rinci dengan ditampilkan data bisa berupa matriks. Berikut pada gambar 3.23 contoh *listing code* untuk import hasil klasifikasi menjadi matriks.

****

Gambar 3.23 Menentukan hasil klasifikasi

Gambar 3.23 merupakan penentuan hasil klasifikasi berdasarkan *precision, recall, dan* f1-*score.*