

BAB V

SIMPULAN DAN SARAN

5.1 Simpulan

1. Diketahui bahwa faktor utama yang paling mempengaruhi risiko terjadinya diabetes adalah kadar gula darah yang tinggi. Faktor lainya seperti insulin, BMI dan umur juga mempengaruhi resiko terjadinya diabetes
2. Pengujian yang dilakukan dengan perhitungan manual dan menggunakan python pada *dataset* yang digunakan menghasilkan nilai akurasi. Nilai akurasi ini menjadi indikator untuk menentukan seberapa baik algoritma dalam mengklasifikasikan individu yang terindikasi mengidap diabetes serta menentukan kinerja algoritma yang digunakan.
3. Berdasarkan hasil pengujian menggunakan python terhadap *dataset* yang digunakan. Algoritma *ID3* menghasilkan akurasi sebesar 81.4% dan *naïve bayes* akurasi sebesar 76.2% dengan menggunakan metode evaluasi *10 fold cross validation*. Berdasarkan hasil tersebut, dapat disimpulkan bahwa algoritma *ID3* terbukti lebih baik dalam mengklasifikasikan penyakit diabetes.

5.2 Saran

1. Perlu dilakukan uji coba dengan algoritma klasifikasi lainnya, seperti *Neural Network*, *K-NN*, *K-Means*, dan lainnya, untuk membandingkan tingkat akurasi serta menentukan algoritma yang paling optimal dalam mengklasifikasikan individu yang terindikasi diabetes.
2. Penelitian ini perlu dikembangkan lebih lanjut, mengingat *dataset* diabetes yang digunakan hanya mencakup data perempuan. Oleh karena itu, penelitian

selanjutnya perlu dilakukan dengan menggunakan *dataset* yang mencakup data pria agar analisis dapat memberikan hasil yang lebih akurat.

3. Untuk meningkatkan akurasi dan ketepatan prediksi, perlu dilakukan perluasan *dataset* dengan mengumpulkan lebih banyak data yang relevan, sehingga model yang digunakan dapat memberikan hasil yang lebih akurat.



DAFTAR PUSTAKA

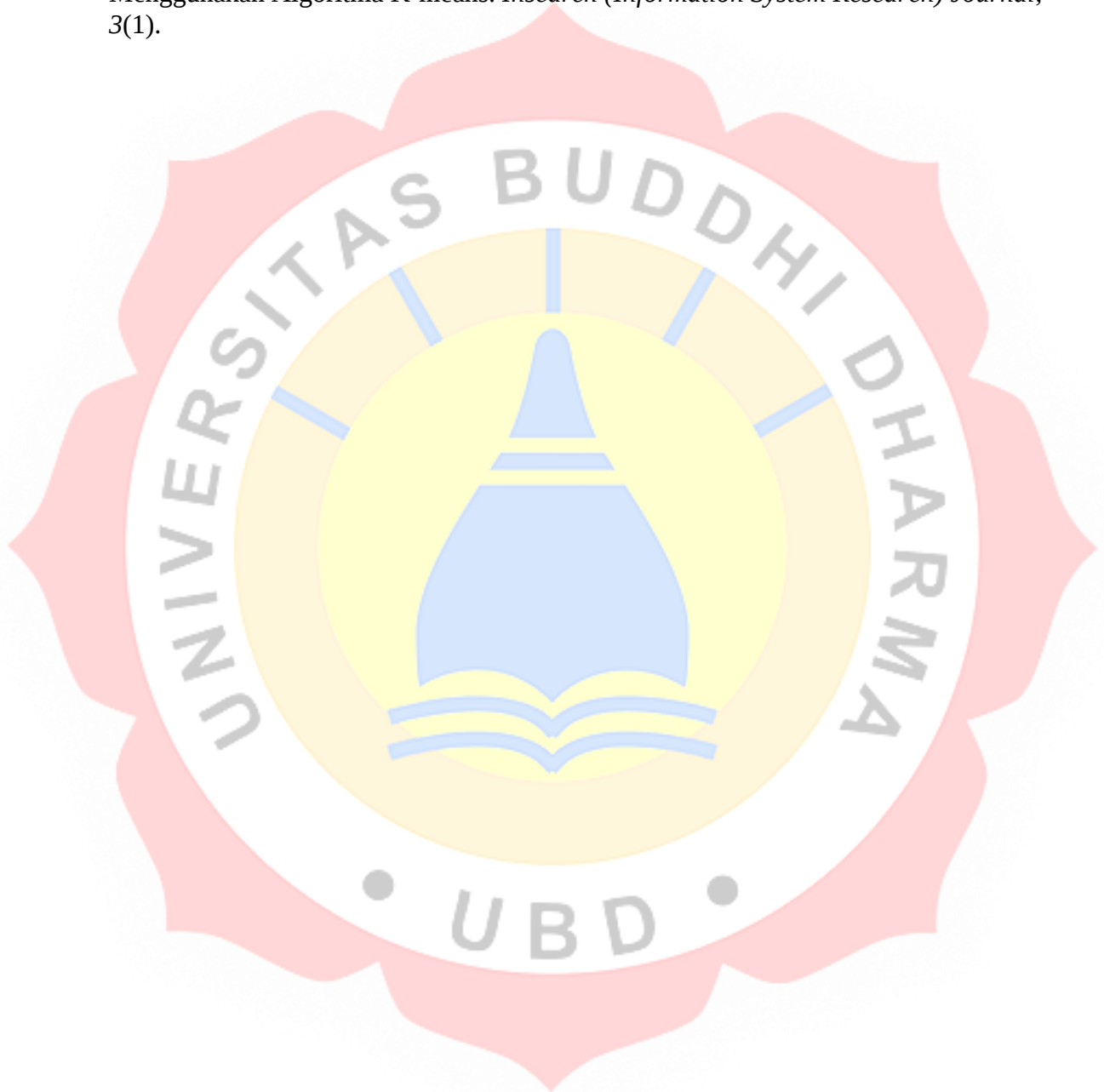
- Adrianus, H., & Lasut, D. (2023). Penerapan *Data mining* Untuk Memprediksi Penyakit Diabetes Menggunakan Algoritma *Naive bayes*. *Algor*, 4(2), 75–85.
- Argina, A. M. (2020). Penerapan Metode Klasifikasi K-Nearest Neighbor pada *Dataset* Penderita Penyakit Diabetes. *Indonesian Journal of Data and Science*, 1(2), 29–33. <https://doi.org/10.33096/ijodas.v1i2.11>
- Arifin, O., Saputra, K., & Fathoni, H. (2021). Implementation of *Data mining* using *NAIVE BAYES* Classifier Method in Food Crop Prediction. *Scientific Journal of Informatics*, 8(1), 43–50. <https://doi.org/10.15294/sji.v8i1.28354>
- Benarbia, M. (2022). *A Machine Learning Approach to Predicting the Onset of Type II A Machine Learning Approach to Predicting the Onset of Type II Diabetes in a Sample of Pima Indian Women Diabetes in a Sample of Pima Indian Women*. https://academicworks.cuny.edu/gc_etds/4895Discoveradditionalworksat:https://academicworks.cuny.edu
- Budiarto, E., Rino, R., Hariyanto, S., & Susilawati, D. (2022). Penerapan *Data mining* Untuk Rekomendasi Beasiswa Pada SD Maria Mediatrix Menggunakan Algoritma C4.5. *Algor*, 3(2), 23–34. <https://doi.org/10.31253/algor.v3i2.1019>
- Bustomi, Y., Nugraha, A., Juliane, C., & Rahayu, S. (2023). *Data mining* Selection of Prospective Government Employees with Employment Agreements using *Naive bayes* Classifier. *Sinkron*, 8(1), 1–8. <https://doi.org/10.33395/sinkron.v8i1.11968>
- Fadli, A., Rahmatulloh, A., Atmaja, G., & ... (2022). Pengenalan Dasar Pembuatan Website HTML Dengan Software Visual Studio Code. *Jurnal Pengabdian Masyarakat*, 1(11), 1325–1331. <https://journal.mediapublikasi.id/index.php/amma/article/view/1338%0Ahttps://journal.mediapublikasi.id/index.php/amma/article/download/1338/628>
- Hardiyanti, T. O., Wurjanto, A., Kusariana, N., Hestningsih, R., Epidemiologi dan Penyakit Tropik, P., Kesehatan Masyarakat Universitas Diponegoro, F., Epidemiologi dan Penyakit Tropik, B., & Kesehatan Masyarakat, F. (2021). *HUBUNGAN JENIS KELAMIN DAN BIDANG STUDI DENGAN PRAKTIK PENCEGAHAN DIABETES MELLITUS TIPE 2 PADA MAHASISWA (Studi Pada Mahasiswa Universitas Diponegoro Semarang)*. 9(2). <http://ejournal3.undip.ac.id/index.php/jkm>
- Jain, R., Shrivastava, V., Pandey, A., & Sharma, A. (2024). Modern Web Development using CSS & HTML. *International Journal of Emerging Science and Engineering*, 12(6), 13–16. <https://doi.org/10.35940/ijese.g2574.12060524>
- Jauhari, A., Rosa Anamisa, D., & Ayu Mufarroha, F. (2020). *Pengantar Teknologi Informasi: Model, Siklus, Desain, Sistem Pendukung Keputusan*. Media Nusa Creative. <https://books.google.co.id/books?id=EH5JEAQAQBAJ>
- Lundén, N., Bekar, E. T., Skoogh, A., & Bokrantz, J. (2023). Domain Knowledge in CRISP-DM: An Application Case in Manufacturing. *ScienceDirect*, 56(2), 7603–7608. <https://doi.org/10.1016/j.ifacol.2023.10.1156>
- Maisa Hana, F. (2023). Penerapan Algoritma *ID3 Decision tree* Pada Klasifikasi Penyakit Diabetes. *Jurnal Dinamika Informatika*, 12(2), 1–11.

- Nazanah, J. T. M. A., & Jambak, M. I. (2023). Pemanfaatan Algoritma *Decision tree ID3* Bagi Manajemen Bimbel Untuk Menentukan Faktor Kelulusan Pada Sekolah Kedinasan. *KLIK: Kajian Ilmiah Informatika Dan Komputer*, 3(6), 915–924. <https://doi.org/10.30865/klik.v3i6.791>
- Noija, H. C., Wemaf, P. A., Nurdianty, O. A., Sohilait, W., Haumahu, S., Yusuf, H., Tomagola, K., Rindi, W., Ana, L., Salaiswa, T. S., Akuntansi, J., Ekonomi, F., & Bisnis, D. (2023). Perancangan Sistem Informasi Akuntansi Untuk Analisis Siklus Pendapatan Pada Orantata Celullar Menggunakan DFD Dan *Flowchart*. *Jurnal Bisnis Dan Manajemen (JURBISMAN)*, 1(2), 577–592. <https://ejournal.lapad.id/index.php/jurbisman/article/view/188>
- Pahlevi, R., Fredlina, K. Q., & Utami, N. W. (2021). Penerapan Algoritma *ID3* dan SVM Pada Klasifikasi Penyakit Diabetes Melitus Tipe 2. *Prosiding Seminar Nasional Aplikasi Sains & Teknologi (SNAST)*, 2, A64–A75. <https://ejournal.akprind.ac.id/index.php/snast/article/view/3340>
- Pangestu, R. A., & Noris, S. (2023). Analisa *Data mining* Prediksi Lelang Suku Cadang Dengan Metode K-NearestNeighbor (Studi Kasus PT. Parmud Jaya Perkasa). *Jurnal Informatika Multi*, 1(4), 285–295.
- Prasetia, O., Machfud, S., Pamulang, U., Selatan, T., Bayes, N., & Miner, R. (2024). *Analisis Prediksi Penyakit Demensia Alzheimer Menggunakan Metode Decision tree C4.5 Dan Naive*. 5, 14–25.
- Rahman, R., & Sutanto, F. A. (2023). *Data mining* to Predict Gojek's Consumer Satisfaction Level Using *Naive bayes* Algorithm. *International Journal of Information System & Technology Akreditasi*, 6(158), 590–602.
- Rawat, B., Purnama, S., & Mulyati, M. (2021). MySQL Database Management System (DBMS) On FTP Site LAPAN Bandung. *International Journal of Cyber and IT Service Management*, 1(2), 173–179. <https://doi.org/10.34306/ijcitsm.v1i2.47>
- Renaldy, & Putra, D. S. D. (2023). Aplikasi Prediksi Harga Ayam Dengan Metode Naives Bayes Pada Supplier Ayam Potong. *Jurnal Algor*, 4(2), 141–148. <http://repository.buddhidharma.ac.id/id/eprint/1609>
- Ridho, M. R. (2024). *Pengertian Flowchart: Fungsi, Jenis, Simbol, dan Contohnya*. Telkom University. <https://bee.telkomuniversity.ac.id/pengertian-Flowchart-fungsi-jenis-simbol-dan-contohnya/#:~:text=Flowchart adalah alat visual yang,menyelesaikan suatu tugas atau masalah>
- Ridwan, A. (2020). Penerapan Algoritma *NAIVE BAYES* Untuk Klasifikasi Penyakit Diabetes Mellitus. *Jurnal SISKOM-KB (Sistem Komputer Dan Kecerdasan Buatan)*, 4(1), 15–21. <https://doi.org/10.47970/siskom-kb.v4i1.169>
- Rino, R. (2021). Comparison of *Data mining* Methods Using *C4.5* Algorithm and *Naive bayes* in Predicting Heart Disease. *Tech-E*, 4(2), 44–51. <https://doi.org/10.31253/te.v4i2.543>
- Schrauf, M. F., de los Campos, G., & Munilla, S. (2021). Comparing Genomic Prediction Models by Means of Cross Validation. *Frontiers in Plant Science*, 12(November), 1–11. <https://doi.org/10.3389/fpls.2021.734512>
- Syafrudin, M. (2020). *Deep Neural Network for Predicting Diabetic Retinopathy from Risk*

Factors. ResearchGate. https://www.researchgate.net/figure/Model-validation-a-10-fold-cross-validation-CV-and-b-stratified-10-fold-CV_fig1_344335921

Timotius, A., & Fenriana, I. (2024). PERANCANGAN APLIKASI PREDIKSI PENYAKIT JANTUNG MENGGUNAKAN METODE NAÏVE BAYES. *JURNAL ALGOR*, 5(2), 54–65.

Yoliadi, D. N. (2023). *Data mining* Dalam Analisis Tingkat Penjualan Barang Elektronik Menggunakan Algoritma K-means. *Insearch (Information System Research) Journal*, 3(1).



LAMPIRAN

Lampiran 1 : Kartu Bimbingan Skripsi



UNIVERSITAS BUDDHI DHARMA

Jl. Imam Bonjol No. 41 Karawaci Ilir, Tangerang
021 5517853 / 021 5586822 admin@buddhidharma.ac.id

KARTU BIMBINGAN TA/SKRIPSI

NIM : 20211000041
Nama Mahasiswa : CALVIN TANUJAYA
Fakultas : Sains dan Teknologi
Program Studi : Teknik Informatika
Jenjang : Strata Satu
Tahun Akademik/Semester : 2024/2025 Genap
Dosen Pembimbing : Desiyanna Lasut, S.Kom., M.Kom
Judul Skripsi : PERBANDINGAN ALGORITMA NAIVE BAYES DAN DECISION TREE ID3 DALAM MENGLASIFIKASIKAN PENYAKIT DIABETES



Tanggal	Catatan	Paraf
2025-03-04	Pengajuan judul	
2025-03-18	Acc judul dan jurnal referensi	
2025-04-08	Bab 1	
2025-04-22	Bab 2	
2025-04-29	Bab 3	
2025-05-13	Bab 4	
2025-05-27	Bab 5	
2025-06-03	Final Program	
2025-06-10	Persentasi	

Mengetahui
Ketua Program Studi



Hartana Wijaya, M.Kom

Tangerang, 11 June 2025
Pembimbing

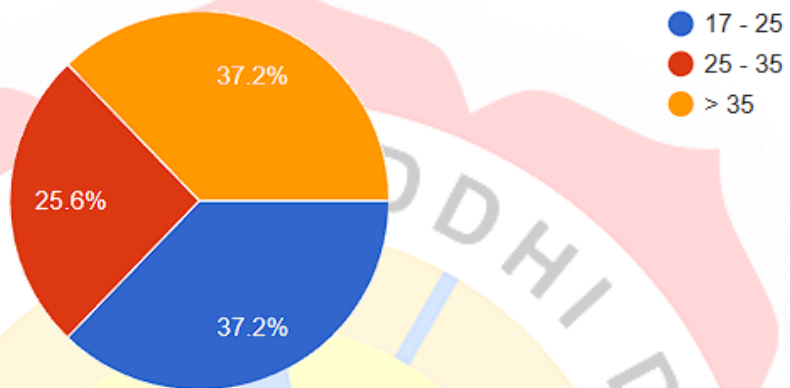


Desiyanna Lasut, S.Kom., M.Kom

Lampiran 2 : Hasil Kuesioner *User Acceptance Testing* (UAT)

Umur

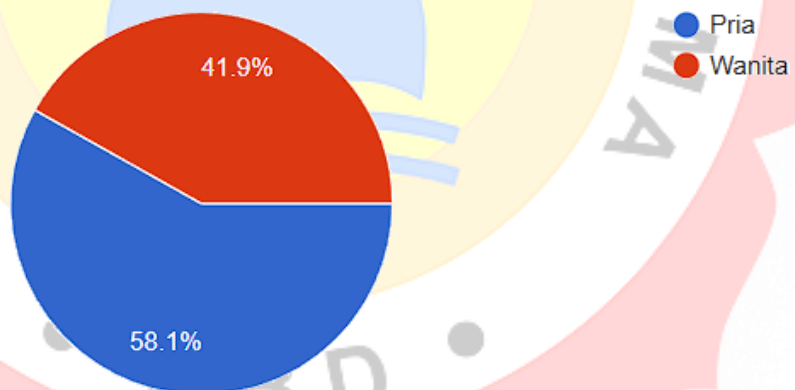
43 responses



Lampiran 2.1 Kuisisioner Umur

Jenis Kelamin

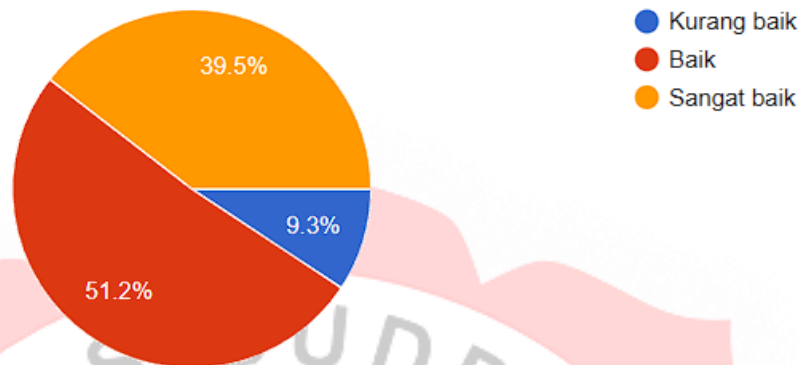
43 responses



Lampiran 2.2 Kuesioner jenis kelamin

1. Apakah hasil prediksi ditampilkan dengan jelas?

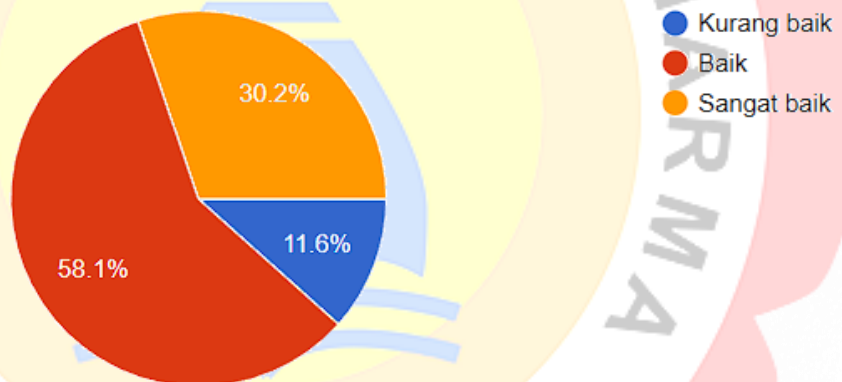
43 responses



Lampiran 2.3 kuesioner 1

2. Apakah Anda merasa terbantu dengan fitur prediksi yang disediakan?

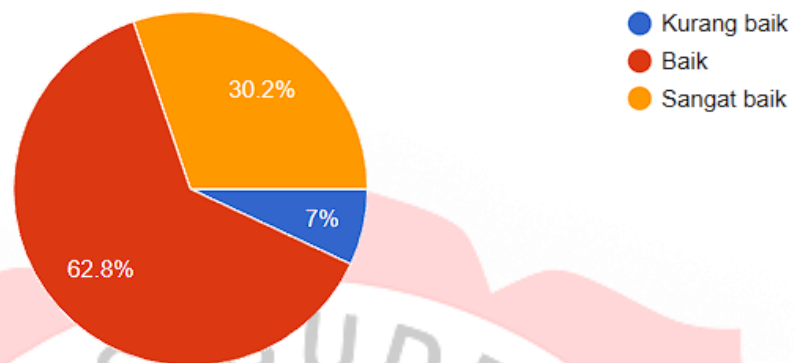
43 responses



Lampiran 2.4 Kuisisioner 2

3. Apakah fitur riwayat prediksi membantu dalam memantau kondisi kesehatan?

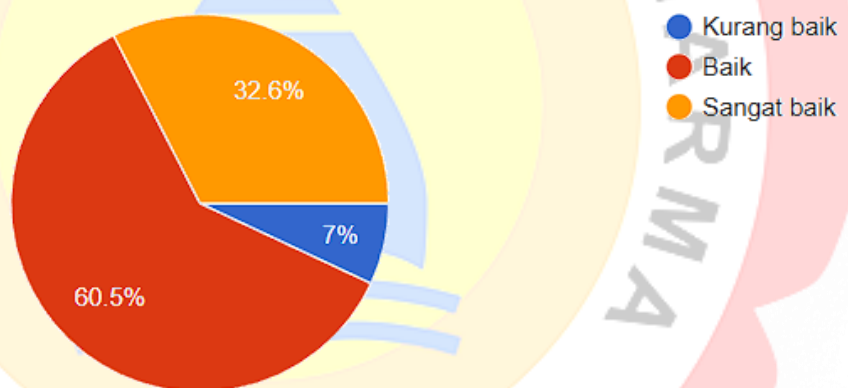
43 responses



Lampiran 2.5 kuesioner 3

4. Apakah tampilan website mudah digunakan dan dipahami?

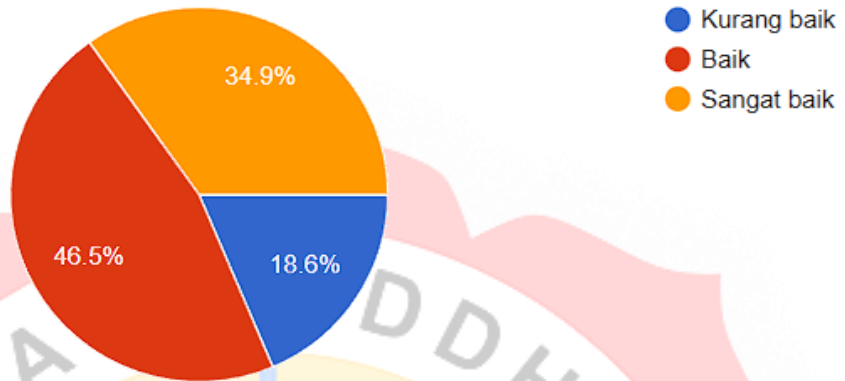
43 responses



Lampiran 2.6 kuesioner 4

5. Apakah input data kesehatan mudah dimasukkan tanpa kebingungan?

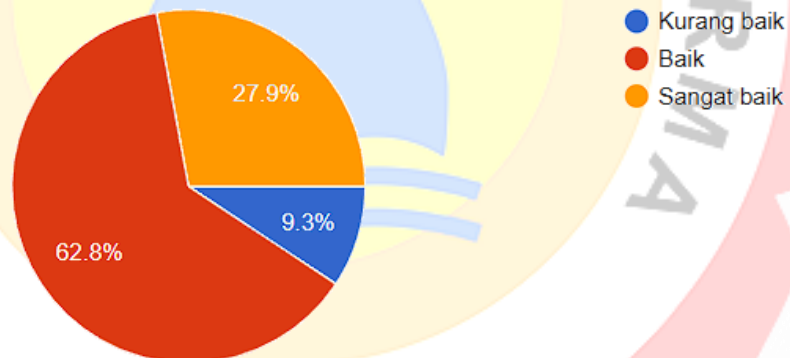
43 responses



Lampiran 2.7 kuesioner 5

6. Apakah website ini dapat membantu pengguna merasa lebih sadar akan risiko diabetes?

43 responses



Lampiran 2.8 kuesioner 6

Lampiran 3 : Requirement elicitation

PERBANDINGAN ALGORITMA NAIVE BAYES DAN DECISION TREE ID3 DALAM MENGLASIFIKASIKAN PENYAKIT DIABETES

Nama : Akmal Sulistyanto

Umur : 18

Pekerjaan : Karyawan

Requirement Elicitation

No	Analisis Kebutuhan Aplikasi	Keterangan
	Saya ingin sistem dapat :	
1	harus akurat	✓
2	ada hasil	✓
3	tidak bisa bisa aplikasi buat hp	X
4	bisa cek tgl pengcekkan	✓
5	mudah dipakai	✓
6	bisa tlpn dokter sekitar sekitar	X
7		
8		

Tangerang, 8 Juli 2025

Pembimbing

Responden

Mahasiswa

Desiyanna Lasut S.Kom., M.Kom

NUPTK. 1534764665230253

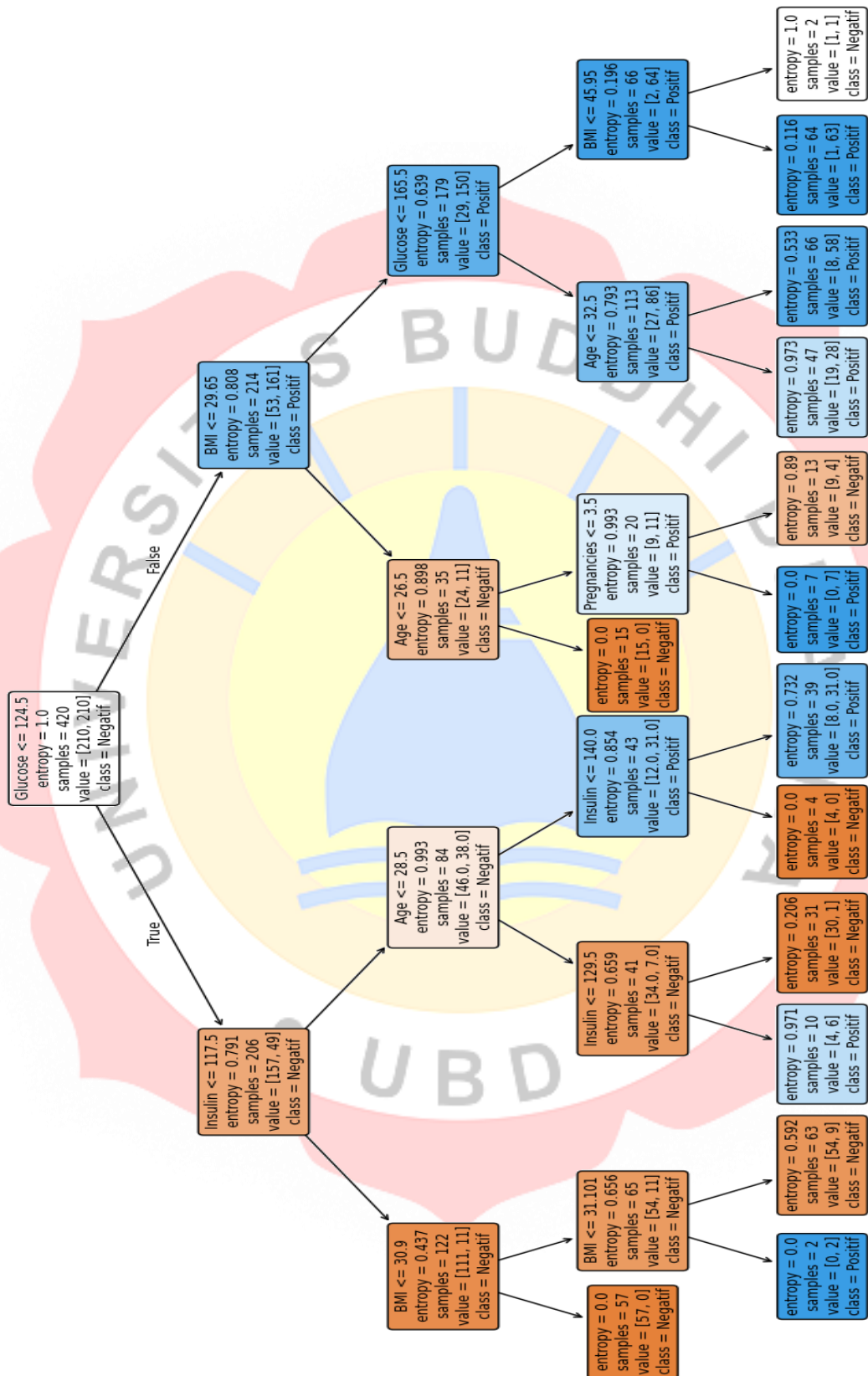
Akmal Sulistyanto

Calvin Tanujaya

20211000041

Lampiran 3.1 Requirement Elicitation

Lampiran 4 : Decision tree ID3



Lampiran 4.1 Decision tree ID3

Lampiran 5 : Hitung manual

Kehamilan	Gula Darah	Tekanan Darah	Ketebalan Kuli	Insulin	BMI	Riwayat Diabetes	Umur	Hasil	
0	137	40	35	168	43.1	2.288	33	Positif	
3	78	50	32	88	31	0.248	26	Positif	
5	166	72	19	175	25.8	0.587	51	Positif	
10	125	70	26	115	31.1	0.205	41	Positif	
3	88	58	11	54	24.8	0.267	22	Negatif	
4	103	60	33	192	24	0.966	33	Negatif	
4	111	72	47	207	37.1	1.39	56	Positif	
1	101	50	15	36	24.2	0.526	26	Negatif	
5	88	66	21	23	24.4	0.342	30	Negatif	
0	100	88	60	110	46.8	0.962	31	Negatif	
Average	Positif	Kehamilan	Gula Darah	Tekanan Darah	Ketebalan Kuli	Insulin	BMI	Riwayat Diabetes	Umur
	Negatif	4.4	123.4	60.8	31.8	150.6	33.62	0.9436	41.4
Var	Positif	13.3	1054.3	221.2	109.7	2316.3	44.087	0.7908093	153.3
	Negatif	4.3	54.5	206.8	389	4815	100.888	0.1117838	19.3
StdDev	Positif	3.646916506	32.46998614	14.87279395	10.47377678	48.12795445	6.639804214	0.889274592	12.3814377
	Negatif	2.073644135	7.38241153	14.38054241	19.72308292	69.39020104	10.04430187	0.334340844	4.39317653
Data testing	Kehamilan	Gula Darah	Tekanan Darah	Ketebalan Kuli	Insulin	BMI	Riwayat Diabetes	Umur	
	3	110	70	30	120	35	0.5	35	
Probabilitas	Positif	Kehamilan	Gula Darah	Tekanan Darah	Ketebalan Kuli	Insulin	BMI	Riwayat Diabetes	Umur
	Negatif	0.1016468	0.011286396	0.022158398	0.037540789	0.006773941	0.05881457	0.396232852	0.02819875
		0.188888733	0.008951354	0.025722656	0.02012855	0.004988647	0.032917576	1.127720395	0.02938606
Positif	0.5								
Negatif	0.5								
Hasil	Positif	2.12406E-12	Confidence	0.471376203					
	Negatif	2.38203E-12		0.528623797					

Lampiran 5.1 Hitung manual naive bayes

Gula Da	Tekanan Da	Ketebalan K	Insulin	BMI	Riwayat Diab	Umur	Hasil		
>120	<=60	>30	>100	>35	>1	<=45	Positif		
<=120	<=60	>30	<=100	<=35	<=1	<=45	Positif		
>120	>60	<=30	>100	<=35	<=1	>45	Positif		
>120	>60	<=30	>100	<=35	<=1	<=45	Positif		
<=120	<=60	<=30	<=100	<=35	<=1	<=45	Negatif		
<=120	<=60	>30	>100	<=35	<=1	<=45	Negatif		
<=120	>60	>30	>100	>35	>1	>45	Positif		
<=120	<=60	<=30	<=100	<=35	<=1	<=45	Negatif		
<=120	>60	<=30	<=100	<=35	<=1	<=45	Negatif		
<=120	>60	>30	>100	>35	<=1	<=45	Negatif		
1 Gula darah									
	Jumlah_Kasus	Positif	Negatif	Entropy	Gain				
	10	5	5	1					
<=3	5	2	3	0.97095					
>3	5	3	2	0.97095	0.029049406				
<=120	7	2	5	0.86312					
>120	3	3	0	0	0.395815602				
<=60	5	2	3	0.97095					
>60	5	3	2	0.97095	0.029049406				
<=30	5	2	3	0.97095					
>30	5	3	2	0.97095	0.029049406				
<=100	4	1	3	0.81128					
>100	6	4	2	0.9183	0.12451125				
<=35	7	3	4	0.98523					
>35	3	2	1	0.9183	0.034851555				
<=1	8	3	5	0.95443					
>1	2	2	0	0	0.236452798				
<=45	8	3	5	0.95443					
>45	2	2	0	0	0.236452798				
2 Riwayat diabetes									
	Jumlah_Kasus	Positif	Negatif	Entropy	Gain				
Total	7	2	5	0.86312					
Kehamilan									
<=3	4	1	3	0.81128					
>3	3	1	2	0.9183	0.00598				
Tekanan Darah									
<=60	4	1	3	0.81128					
>60	3	1	2	0.9183	0.00598				
Ketebalan Kulit									
<=30	3	0	3	0					
>30	4	2	2	1	0.29169				
Insulin									
<=100	4	1	3	0.81128					
>100	3	1	2	0.9183	0.00598				
BMI									
<=35	5	1	4	0.72193					
>35	2	1	1	1	0.06174				
Riwayat Diabetes									
<=1	6	1	5	0.65002					
>1	1	1	0	0	0.30596				
Umur									
<=45	6	1	5	0.65002					
>45	1	1	0	0	0.30596				

Lampiran 5.2 Hitung manual ID3

3 Ketebalan kulit						4 Insulin					
	Jumlah_Kasus	Positif	Negatif	Entropy	Gain	Total	Jumlah_Kasus	Positif	Negatif	Entropy	Gain
	6	1	5	0.65002		Kehamilan	3	1	2	0.9183	
<=3	4	1	3	0.81128		<=3	2	1	1	1	0.25163
>3	2	0	2	0	0.109170339	>3	1	0	1	0	
<=60	4	1	3	0.81128		Tekanan Darah					
>60	2	0	2	0	0.109170339	<=60	2	1	1	1	0.25163
<=30	3	0	3	0		>60	1	0	1	0	
>30	3	1	2	0.9183	0.190874505	Insulin					
<=100	4	1	3	0.81128		<=100	1	1	0	0	0.9183
>100	2	0	2	0	0.109170339	>100	2	0	2	0	
<=35	5	1	4	0.72193		BMI					
>35	1	0	1	0	0.048415676	<=35	2	1	1	1	0.25163
<=45	6	1	5	0.65002		>35	1	0	1	0	
>45	0	0	0	0	0	Umur					
						<=45	3	1	2	0.9183	0
						>45	0	0	0	0	

Lampiran 5.3 Hitung manual ID3



Lampiran 6 : Listing Program

```
1 from matplotlib import pyplot as plt
2 import pandas as pd
3 from flask import Flask, render_template, request, redirect, url_for, session
4 from sqlalchemy import create_engine, MetaData, Table, insert, text
5 from sklearn.model_selection import cross_val_score, train_test_split
6 from sklearn.preprocessing import StandardScaler
7 from sklearn.tree import DecisionTreeClassifier, plot_tree
8 from sqlalchemy.orm import sessionmaker
9 from datetime import datetime
10 import bcrypt
11 # Konfigurasi Flask
12 app = Flask(__name__)
13 app.secret_key = 'key' # Secret key untuk sesi pengguna
14
15 # Konfigurasi database
16 db_user = 'root'
17 db_password = ''
18 db_host = 'localhost'
19 db_name = 'diabetes'
20 engine = create_engine(f'mysql+pymysql://{db_user}:{db_password}@{db_host}/{db_name}')
21
22 # Membuat sesi untuk melakukan operasi database
23 Session = sessionmaker(bind=engine)
24 session_db = Session()
25
26 # Menyiapkan tabel inputdata dan dataset
27 metadata = MetaData()
28 metadata.reflect(bind=engine)
29 inputdata_table = metadata.tables.get('inputdata')
30 dataset_table = metadata.tables.get('dataset')
31
32 query = "SELECT glucose, insulin, bmi, age, outcome FROM dataset"
33 df = pd.read_sql(query, engine)
34
35 X = df.drop('outcome', axis=1)
36 y = df['outcome']
37 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
38
39 id3_model = DecisionTreeClassifier(criterion='entropy', random_state=42, max_depth=3)
40 id3_model.fit(X, y)
41
42 # Halaman Login
43 @app.route('/', methods=['GET', 'POST'])
44 def login():
45     if request.method == 'POST':
46         username = request.form['username']
47         password = request.form['password']
48
49         # Ambil data pengguna berdasarkan username
50         query = text("SELECT * FROM user WHERE username=:username")
51         with engine.connect() as conn:
52             result = conn.execute(query, {"username": username}).mappings().fetchone()
53
54         if result:
55             stored_password = result['password'] # Hasil sudah berupa dict
56             if bcrypt.checkpw(password.encode('utf-8'), stored_password.encode('utf-8')):
57                 session['username'] = username
58                 return redirect(url_for('welcome'))
59             else:
60                 error = "Password salah!"
61         else:
62             error = "Username tidak ditemukan!"
63
64         return render_template('Login.html', error=error)
65
66     return render_template('Login.html')
```

Lampiran 6.1 Listing program python

```

69 @app.route('/register', methods=['GET', 'POST'])
70 def register():
71     if request.method == 'POST':
72         username = request.form['username']
73         password = request.form['password']
74         confirm_password = request.form['confirm_password']
75
76         #Konfirmasi password
77         if password != confirm_password:
78             error = "Password dan konfirmasi password tidak cocok!"
79             return render_template('register.html', error=error)
80
81         # Periksa apakah username sudah ada di database
82         query = text("SELECT * FROM user WHERE username=:username")
83         with engine.connect() as conn:
84             result = conn.execute(query, {"username": username}).fetchone()
85
86         if result:
87             error = "Username sudah digunakan, silakan pilih username lain."
88             return render_template('register.html', error=error)
89
90         # Hash password sebelum disimpan
91         hashed_password = bcrypt.hashpw(password.encode('utf-8'), bcrypt.gensalt())
92
93         # Simpan pengguna baru ke database
94         stmt = text("INSERT INTO user (username, password) VALUES (:username, :password)")
95         with engine.connect() as conn:
96             conn.execute(stmt, {"username": username, "password": hashed_password.decode('utf-8')})
97             conn.commit()
98
99         return redirect(url_for('Login'))
100     return render_template('register.html')
101
102
103 @app.route('/Logout')
104 def logout():
105     session.pop('username', None) # Hapus username dari sesi
106     return redirect(url_for('Login')) # Arahkan kembali ke halaman login
107
108 # Halaman Welcome setelah login
109 @app.route('/welcome')
110 def welcome():
111     if 'username' not in session:
112         return redirect(url_for('Login')) # Jika belum login, arahkan ke halaman login
113
114     username = session['username'] # Ambil username dari sesi
115     return render_template('welcome.html', username=username) # Render halaman datang.html

```

Lampiran 6.2 Listing program python

```

118 @app.route('/index', methods=['GET', 'POST'])
119 def index():
120     if 'username' not in session:
121         return redirect(url_for('login'))
122
123     username = session['username']
124     result = None
125     saran = None
126
127     if request.method == 'POST':
128         try:
129             tahu_glucose = request.form['tahu_glucose']
130
131             if tahu_glucose == 'ya':
132                 glucose = int(request.form['glucose'])
133             else:
134                 gejala = request.form.getlist('gejala')
135                 skor = len(gejala)
136
137                 if skor >= 2:
138                     glucose = 140 # diasumsikan tinggi
139                 elif skor == 1:
140                     glucose = 120 # sedang
141                 else:
142                     glucose = 100 # normal
143
144             tahu_insulin = request.form['tahu_insulin']
145
146             if tahu_insulin == 'ya':
147                 insulin = int(request.form['insulin'])
148             else:
149                 gejala = request.form.getlist('gejalainsulin')
150                 skor = len(gejala)
151
152                 if skor >= 2:
153                     insulin = 200 # diasumsikan tinggi
154                 elif skor == 1:
155                     insulin = 100 # sedang
156                 else:
157                     insulin = 50 # normal
158
159             height_cm = float(request.form['height'])
160             weight_kg = float(request.form['weight'])
161             bmi = weight_kg / (height_cm ** 2)
162             age = int(request.form['age'])
163
164             input_data = [glucose, insulin, bmi, age]
165             prediction = idf_model.predict(input_data)
166
167             outcome_str = "Diabetes" if prediction == 1 else "Negatif"
168             current_time = datetime.now()
169
170             stmt = insert(insert_data_table).values(
171                 glucose=glucose,
172                 insulin=insulin,
173                 weight=weight,
174                 height=height_cm,
175                 bmi=bmi,
176                 age=age,
177                 outcome=outcome_str,
178                 username=username,
179                 waktu=current_time
180             )
181
182             with engine.connect() as conn:
183                 conn.execute(stmt)
184                 conn.commit()
185
186             # Prediction results and suggestions
187             if prediction == 1:
188                 result = outcome_str
189                 saran = """
190                 <h3>Sangat memerlukan pemeriksaan ke dokter</h3>
191                 <ul>
192                 <li><b>Diet</b>: Perhatikan tingkat gula dan karbohidrat sederhana. Konsumsi lebih banyak sayuran, buah rendah gula, protein tanpa lemak, dan biji-bijian utuh.</li>
193                 <li><b>Aktivitas Fisik</b>: Lakukan olahraga rutin minimal 30 menit per hari, seperti berjalan cepat, berenang, atau yoga.</li>
194                 <li><b>Pemeriksaan Rutin</b>: Lakukan pemeriksaan ke dokter serta periksa gula darah secara teratur untuk memantau kadar gula dan konsultasikan perkembangan dengan dokter spesialis.</li>
195                 <li><b>Pengelolaan Stres</b>: Kurangi stres melalui teknik relaksasi seperti meditasi, yoga, atau terapi konseling.</li>
196                 <li><b>Obat-obatan</b>: Minum obat atau insulin sesuai resep dokter untuk mengontrol kadar gula darah.</li>
197                 <li><b>Hidrasi</b>: Konsumsi air putih yang cukup untuk menjaga metabolisme tubuh.</li>
198                 </ul>
199                 """
200             else:
201                 result = outcome_str
202                 saran = """
203                 <h3>H3</h3>
204                 <ul>
205                 <li><b>Diet</b>: Perhatikan asupan gula dengan asupan sayuran, buah, dan serat yang cukup.</li>
206                 <li><b>Aktivitas Fisik</b>: Tetap aktif secara fisik untuk menjaga berat badan ideal dan kesehatan jantung.</li>
207                 <li><b>Pemeriksaan Berkala</b>: Lakukan tes kesehatan secara rutin untuk mendeteksi risiko diabetes atau penyakit lainnya lebih awal.</li>
208                 <li><b>Hindari Gula Berlebih</b>: Kurangi konsumsi gula tambahan dan makanan olahan untuk mencegah risiko di masa depan.</li>
209                 <li><b>Hindari Stres Berlebih</b>: Manajemen stres melalui hobi atau aktivitas santai dapat membantu menjaga kesehatan mental dan fisik.</li>
210                 </ul>
211                 """
212
213             return render_template("index.html", result=outcome_str, saran=saran)
214
215         except Exception as e:
216             print(f"Error: {e}")
217             return render_template("index.html", result="Error ketika melakukan prediksi", saran=None)
218
219     return render_template("index.html", result=None, saran=None)

```

Lampiran 6.3 Listing program python

```

222 @app.route('/history')
223 def history():
224     if 'username' not in session:
225         return redirect(url_for('login'))
226
227     username = session['username']
228     query = f"SELECT * FROM inputdata WHERE username='{username}'"
229     user_data = pd.read_sql(query, engine)
230
231     return render_template("history.html", username=username, user_data=user_data)
232
233 # Route untuk update data pada history
234 @app.route('/update/cint:record_id', methods=['GET', 'POST'])
235 def update(record_id):
236     if 'username' not in session:
237         return redirect(url_for('login'))
238
239     username = session['username']
240
241     # Ambil data berdasarkan ID
242     query = f"SELECT * FROM inputdata WHERE id={record_id} AND username='{username}'"
243     with engine.connect() as conn:
244         record = conn.execute(text(query)).fetchone()
245
246     if request.method == 'POST':
247         try:
248             tahu_glucose = request.form['tahu_glucose']
249
250             if tahu_glucose == 'ya':
251                 glucose = int(request.form['glucose'])
252             else:
253                 gejala = request.form.getlist('gejala')
254                 skor = len(gejala)
255
256                 if skor >= 2:
257                     glucose = 140 # diasumsikan tinggi
258                 elif skor == 1:
259                     glucose = 120 # sedang
260                 else:
261                     glucose = 100 # normal
262
263             tahu_insulin = request.form['tahu_insulin']
264
265             if tahu_insulin == 'ya':
266                 insulin = int(request.form['insulin'])
267             else:
268                 gejala = request.form.getlist('gejalaInsulin')
269                 skor = len(gejala)
270
271                 if skor >= 2:
272                     insulin = 200 # diasumsikan tinggi
273                 elif skor == 1:
274                     insulin = 100 # sedang
275                 else:
276                     insulin = 50 # normal
277
278             height_cm = float(request.form['height'])
279             weight = float(request.form['weight'])
280             height_m = height_cm / 100
281             bmi = weight / (height_m ** 2)
282             age = int(request.form['age'])
283
284             input_data = [[glucose, insulin, bmi, age]]
285             prediction = id3_model.predict(input_data)
286
287             outcome = "Diabetes" if prediction == 1 else "Negatif"
288
289             # Update data di database
290             update_query = text("""
291             UPDATE inputdata
292             SET glucose=:glucose, insulin=:insulin,
293             height=:height, weight=:weight, bmi=:bmi, age=:age,
294             outcome=:outcome, waktu=:waktu
295             WHERE id=:record_id AND username=:username
296             """)
297
298             current_time = datetime.now()
299
300             with engine.connect() as conn:
301                 conn.execute(update_query, {
302                     'glucose': glucose,
303                     'insulin': insulin,
304                     'height': height_cm,
305                     'weight': weight,
306                     'bmi': bmi,
307                     'age': age,
308                     'waktu': current_time,
309                     'outcome': outcome,
310                     'record_id': record_id,
311                     'username': username
312                 })
313                 conn.commit()
314
315             return redirect(url_for('history'))
316
317         except Exception as e:
318             print(f"Error: {e}")
319             return "Terjadi kesalahan saat memperbarui data."
320
321     return render_template("update.html", record=record)
322
323 # Route untuk menghapus data dari history
324 @app.route('/delete/cint:record_id', methods=['GET', 'POST'])
325 def delete(record_id):
326     if 'username' not in session:
327         return redirect(url_for('login'))
328
329     username = session['username'] # Ambil username dari sesi
330
331     # Hapus data dari database
332     delete_query = text("""DELETE FROM inputdata WHERE id=:record_id AND username=:username""")
333     with engine.connect() as conn:
334         conn.execute(delete_query, {'record_id': record_id, 'username': username})
335         conn.commit()
336
337     return redirect(url_for('history'))
338
339 if __name__ == '__main__':
340     app.run(debug=True)

```

Lampiran 6.4 Listing program python

DAFTAR RIWAYAT HIDUP



Data diri

NIM : 20211000041
Nama Lengkap : Calvin Tanujaya
Tempat, Tanggal Lahir : Tangerang, 01 Juli 2003
Jenis kelamin : Laki - Laki
Alamat : Villa Tomang Baru Blok A-4/10
Agama : Kristen
Email : cvin239@gmail.com
Telepon : 0811-8890-220

Pendidikan formal

2010 – 2015 : SD Kasih Bangsa
2015 – 2018 : SMP Maria Mediatrix
2018 – 2021 : SMA Maria Mediatrix
2021 – 2025 : Universitas Buddhi Dharma

Tangerang, 4 Agustus 2025

Calvin Tanujaya