

**PENERAPAN METODE GAUSS-NEWTON DAN ITERASI
JACOBI PADA OPTIMASI REGRESI LOGISTIK BINER**

SKRIPSI

**OLEH
RIENKE ANASTASYA WYANTI
NIM. 210601110103**



**PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2025**

**PENERAPAN METODE GAUSS-NEWTON DAN ITERASI
JACOBI PADA OPTIMASI REGRESI LOGISTIK BINER**

SKRIPSI

**Diajukan Kepada
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Matematika (S.Mat)**

**Oleh
Rienke Anastasya Wyanti
NIM. 210601110103**

**PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2025**

PENERAPAN METODE GAUSS-NEWTON DAN ITERASI JACOBI PADA OPTIMASI REGRESI LOGISTIK BINER

SKRIPSI

Oleh
Rienke Anastasya Wyanti
NIM. 210601110103

Telah Disetujui Untuk Diuji

Malang, 09 Desember 2025

Dosen Pembimbing I



Dr. Mohammad Jamhuri, M.Si
NIP. 19810502 200501 1 004

Dosen Pembimbing II



Erna Herawati, M.Pd
NIPPPK. 19760723 202321 2 006

Mengetahui,
Ketua Program Studi Matematika



Dr. Fachrur Rozi, M.Si
NIP. 19800527 200801 1 012

PENERAPAN METODE GAUSS-NEWTON DAN ITERASI JACOBI PADA OPTIMASI REGRESI LOGISTIK BINER

SKRIPSI

Oleh
Rienke Anastasya Wyanti
NIM. 210601110103

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima sebagai Salah satu Persyaratan
untuk Memperoleh Gelar Sarjana Matematika (S.Mat)

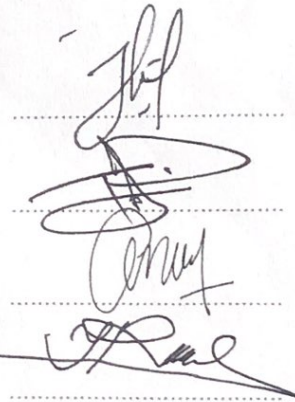
Tanggal 23 Desember 2025

Ketua Penguji : Juhari, M.Si

Anggota Penguji 1 : Hisyam Fahmi, M.Kom

Anggota Penguji 2 : Dr. Mohammad Jamhuri, M.Si

Anggota Penguji 3 : Erna Herawati, M.Pd



Mengetahui,
Ketua Program Studi Matematika



Dr. Fachrur Rozi, M.Si
NIP. 19800527 200801 1 012

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini

Nama : Rienke Anastasya Wyanti

NIM : 210601110103

Program Studi : Matematika

Fakultas : Sains dan Teknologi

Judul Skripsi : Penerapan Metode Gauss-Newton dan Iterasi Jacobi pada
Optimasi Regresi Logistik Biner

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini merupakan hasil karya sendiri, bukan pengambilan tulisan atau pemikiran orang lain yang saya akui sebagai pemikiran saya, kecuali dengan mencantumkan sumber cuplikan pada daftar rujukan di halaman terakhir. Apabila dikemudian hari terbukti skripsi ini adalah hasil jiplakan atau tiruan, maka saya bersedia menerima sanksi yang berlaku atas perbuatan tersebut.

Malang, 23 Desember 2025



Rienke Anastasya Wyanti

NIM. 210601110103

MOTO

“Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya.”

(QS. Al-Baqarah: 286)

“Apa yang ditakdirkan untukmu tidak akan meleset darimu, dan apa yang meleset darimu tidak akan pernah mengenai dirimu.”

(HR. Muslim)

PERSEMBAHAN

Dengan segala kerendahan hati dan penuh rasa syukur, karya ini saya persembahkan kepada:

Ayah Wintoko dan Ibu Siti Fatimah

Kedua orang tuaku, yang tidak pernah berhenti memberikan doa, kasih sayang, dan pengorbanan yang tak terhitung sejak awal hingga akhir perjalanan ini. Setiap pengorbanan dan nasihat yang kalian berikan menjadi kekuatan terbesar dalam perjalanan saya menyelesaikan studi ini. Semoga dengan karya sederhana ini bisa memberikan kebanggaan tersendiri untuk kalian berdua.

Adik Jaessy Dayyana Maulidiyah

Dengan perannya yang memberi dukunganmu dengan caramu sendiri.

Kakung Sumardi dan Uti Winarsih

Yang senantiasa merawatku, mendoakan dan memberi nasihat. Doa kalian adalah penyemangat yang tidak pernah putus.

Untuk diriku sendiri, Rienke Anastasya Wyanti

Terima kasih kepada diriku sendiri telah bertahan hingga mencapai tahap ini yang tetap berjuang di tengah keraguan, meskipun tidak secepat yang lain, namun tetap berusaha memberikan yang terbaik. Tidak menyerah meski pernah diremehkan, dan terus melangkah hingga akhirnya sampai di tahap ini.

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Segala puji syukur penulis panjatkan kehadirat Allah SWT. atas segala rahmat, karunia, serta hidayah-Nya, tidak lupa shalawat serta salam senantiasa tercurahkan kepada Nabi Muhammad SAW., sehingga penulis mampu menyelesaikan skripsi dengan baik.

Hasil dengan judul “Penerapan Metode Gauss-Newton dan Iterasi Jacobi pada Optimasi Regresi Logistik Biner” ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Matematika pada Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Dalam penyusunan skripsi ini, penulis telah banyak menerima bantuan, bimbingan serta arahan, dan dukungan dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan terima kasih yang tulus kepada:

1. Prof. Dr. Hj. Ilfi Nur Diana, M.Si., CAHRM., CRMP., selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim.
2. Dr. Agus Mulyono, M.Kes, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim.
3. Dr. Fachrur Rozi, M.Si, selaku Ketua Program Studi Matematika, Universitas Islam Negeri Maulana Malik Ibrahim.
4. Dr. Mohammad Jamhuri, M.Si, selaku Dosen Pembimbing I, yang dengan sabar memberikan bimbingan, arahan, saran, serta kritik yang sangat membantu penulis dalam Menyusun dan menyelesaikan skripsi ini.
5. Erna Herawati, M.Pd, selaku Dosen Pembimbing II, yang telah memberikan arahan dan masukan yang sangat bermanfaat, khususnya dalam mengintegrasikan nilai-nilai agama dengan penelitian penulis.
6. Juhari, M.Si, selaku Ketua Penguji, yang dengan penuh perhatian memberikan penilaian, masukan, serta rekomendasi yang sangat berharga demi penyempurnaan skripsi ini.
7. Hisyam Fahmi, M.Kom, selaku Anggota Penguji I, yang telah memberikan saran serta masukan guna memperbaiki dan menyempurnakan skripsi ini.

8. Seluruh dosen Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim.
9. Superhero, Ayah Wintoko, terimakasih selalu berjuang untuk kehidupan penulis, beliau memang tidak sempat merasakan pendidikan sampai bangku perkuliahan, namun beliau mampu mendidik penulis, memotivasi, memberikan dukungan dan semangat serta selalu mengajarkan kebaikan dalam hidup penulis sehingga penulis mampu menyelesaikan studinya sampai sarjana. Sehat selalu dan Panjang umur karena ayah harus selalu ada disetiap perjuangan dan pencapaian hidup penulis.
10. Pintu surgaku, Ibu Siti Fatimah yang selalu menjadi penyemangat penulis dan menjadi sandaran terkuat dari kerasnya dunia. Yang tidak henti-hentinya memberikan kasih sayang dengan penuh cinta dan selalu memeberikan motivasi yang luar biasa. Terimakasih untuk doa- doa yang selalu diberikan untuk penulis, terimakasih selalu berjuang untuk penulis, berkat doa serta dukungannya sehingga penulis bisa berada dititik ini. Sehat selalu dan Panjang umur karena ibu harus selalu ada disetiap perjuangan dan pencapaian hidup penulis.
11. Seluruh keluarga tercinta kakek, nenek, paman, dan tante yang selalu memberi doa dan menyayangi penulis dengan tulus.
12. Kepada saudara Jaessy Dayyana Maulidiyah. Terimakasih telah memberi dukungan dan semangat selama ini serta doa yang terbaik buat penulis.
13. Teruntuk sahabat dan teman terkasih penulis Nisak, Dea, Hana, dan Adil terima kasih untuk menjadi partner bertumbuh di segala ko yang terkadang tidak terduga, menjadi pendengar yang baik untuk penulis serta menja orang yang selalu memberikan semangat dan meyakinkan penulis bahwa segala masalah yang dihadapi selama proses skripsi akan berakhir.
14. Kepada pemilik tanggal lahir 12-desember-2002 (AD). Terimakasih telah menjadi sosok pendamping setia dalam segala hal, yang menemani penulis pada masa perkuliahan, tempat berbagi suka duka, memberi dukungan, kasih sayang dan semangat, serta perhatian kepada penulis dalam penyusunan skripsi dari awal hingga selesainya skripsi ini.

15. Last but not least. I wanna thank me. I wanna thank me for believing in me. I wanna thank me for doing hard work. I wanna thank me for having no day off. I wanna thank me for never quitting

16. Seluruh mahasiswa angkatan 2021.

Penulis menyadari bahwa karya tulis ini masih jauh dari kata sempurna dan masih terdapat berbagai kekurangan. Oleh karena itu, segala bentuk kritik dan saran yang bersifat membangun sangat penulis harapkan demi perbaikan dan penyempurnaan di masa mendatang. Semoga karya ini dapat memberikan manfaat bagi para pembaca serta menjadi kontribusi positif dalam pengembangan ilmu pengetahuan.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Malang, 23 Desember 2025

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN	ii
HALAMAN PERSETUJUAN	iii
PERNYATAAN KEASLIAN TULISAN	v
MOTO	vi
PERSEMBAHAN.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	xi
DAFTAR TABEL	xiii
DAFTAR GAMBAR	xiv
DAFTAR SIMBOL	xv
DAFTAR LAMPIRAN	xvi
ABSTRAK	xvii
ABSTRACT	xviii
مستخلص البحث.....	xix
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	5
1.3 Tujuan Penelitian	5
1.4 Manfaat Penelitian	5
1.5 Batasan Masalah	6
1.6 Definisi Istilah	6
BAB II KAJIAN TEORI	9
2.1 Teori Pendukung.....	9
2.1.1 Regresi Logistik Biner.....	9
2.1.2 Metode Gauss-Newton	13
2.1.3 Metode Iterasi Jacobi.....	16
2.2 Integrasi Al-Qur'an dalam Teori Optimasi.....	18
2.3 Kajian Topik dengan Teori Pendukung	22
BAB III METODE PENELITIAN	24
3.1 Jenis Penelitian	24
3.2 Data dan Sumber Data	24
3.3 Langkah-langkah Penyelesaian	25
3.3.1 Praproses Data	27
3.3.2 Penerapan Regresi Logistik Biner	28
3.3.3 Penerapan Metode Gauss-Newton.....	29
3.3.4 Penerapan Metode Iterasi Jacobi	30
3.3.5 Evaluasi Model	30
3.3.6 Analisis Hasil.....	32
BAB IV HASIL DAN PEMBAHASAN.....	34
4.1 Deskripsi Data	34
4.2 Implementasi Metode secara Manual	36
4.2.1 Pra-pemrosesan Data Secara Manual	36
4.2.2 Inisialisasi Bobot	40
4.2.3 Menghitung Fungsi Sigmoid	41

4.2.4	Menghitung Fungsi <i>Loss</i>	43
4.2.5	Formulasi Gauss-Newton	45
4.2.6	Menyusun matriks Jacobian A dan matriks RHS	47
4.2.7	Penyelesaian SPL menggunakan iterasi Jacobi	49
4.2.8	Perhitungan Prediksi.....	52
4.2.9	Evaluasi Model	56
4.3	Hasil Implementasi Secara Komputasional	58
4.3.1	Pra-Pemrosesan Data Secara Komputasional.....	58
4.3.2	Penanganan Masalah Numerik dengan Clipping.....	60
4.3.3	Hasil pelatihan model	63
4.3.4	Evaluasi pada data uji	65
4.4	Analisis Hasil.....	67
4.4.1	Perbandingan dengan Metode Dasar	67
4.4.2	Perbandingan dengan Penelitian Sebelumnya.....	72
4.5	Analisis Integrasi Keislaman	74
BAB V PENUTUP		77
5.1	Kesimpulan.....	77
5.2	Saran	78
DAFTAR PUSTAKA.....		79
LAMPIRAN.....		82
RIWAYAT HIDUP.....		90

DAFTAR TABEL

Tabel 4.1	Contoh Sampel Data 6 Baris Pertama dengan Seluruh Fitur	35
Tabel 4.2	Subset 6 Sample Pertama dengan 2 Fitur Berkorelasi Tinggi.....	36
Tabel 4.3	Tabel Hasil Standarisasi 6 Data Awal dengan 2 Fitur	39
Tabel 4.4	Ringkasan Hasil Prediksi	56
Tabel 4.5	Perhitungan Proporsi Pembagian Dataset	59
Tabel 4.6	Hasil Data Sebelum Standarisasi.....	60
Tabel 4.7	Hasil Data Setelah Standarisasi.....	60
Tabel 4.8	Perkembangan Loss dan Akurasi Selama Proses Konvergensi...	64
Tabel 4.9	Perbandingan Hasil Performa Gauss-Newton +Iterasi Jacobi, SGDM, RMSProp	69
Tabel 4.10	Perbandingan Epoch Konvergensi ($\text{acc-val} < 0.75$) dan Waktu Eksekusi	71
Tabel 4.11	Perbandingan dengan Penelitian Sebelumnya.....	72

DAFTAR GAMBAR

Gambar 3.1	Flowchart Metode Gauss-Newton dan Iterasi Jacobi.....	26
Gambar 4.1	Presentase Tidak Diabetes dan Diabetes pada Data Pima Indian Diabetes	35
Gambar 4.2	Grafik Loss Sebelum Clipping.....	62
Gambar 4.3	Grafik Loss Sesudah Clipping.....	62
Gambar 4.4	Grafik Akurasi Sebelum Clipping	63
Gambar 4.5	Grafik Akurasi Sesudah Clipping.....	63
Gambar 4.6	Confusion Matrix Hasil Prediksi Metode Gauss-Newton + iterasi Jacobi	66
Gambar 4.7	Perbandingan Grafik Loss pada Data Training dan Validasi antara Gauss-Newton + iterasi Jacobi, SGDM, RMSProp.....	68
Gambar 4.8	Perbandingan Grafik Akurasi pada Data Training dan Validasi antara Gauss-Newton + iterasi Jacobi, SGDM, RMSProp	68
Gambar 4.9	Grafik Perbandingan Loss dengan Pemberhentian Iterasi pada <i>Validation Accuracy</i> >0.75 antara Gauss-Newton + iterasi Jacobi, SGDM, RMSProp.....	70
Gambar 4.10	Grafik Perbandingan Akurasi dengan Pemberhentian Iterasi pada <i>Validation Accuracy</i> >0.75 antara Gauss-Newton + iterasi Jacobi, SGDM, RMSProp.....	70

DAFTAR SIMBOL

\mathbb{R}^n	: Ruang fitur berdimensi n
X	: Matriks fitur berdimensi $r \times (n + 1)$ yang berisi seluruh nilai fitur
Y	: Matriks label berdimensi $r \times 1$ yang berisi seluruh nilai label
\vec{x}_i	: Vektor prediktor ke- i
y_i	: Variabel respon biner ke- i (0 atau 1)
r	: Banyaknya data pelatihan
n	: Banyaknya fitur
w_0, w_1, \dots, w_n	: Koefisien regresi
i	: Indeks data ke- i
$h(z_i)$: Fungsi sigmoid
$L(\vec{w})$: Fungsi <i>loss</i>
$\Delta \vec{w}$: Perubahan vektor bobot
A	: Matriks ruas kiri yang berdimensi $r \times (n + 1)$
B	: Matriks ruas kanan yang berdimensi $(r \times 1)$
H	: Matriks Hessian aproksimasi yang berdimensi $(n + 1) \times (n + 1)$
C	: Vektor ruas kanan dalam SPL yang berdimensi $(n + 1) \times 1$
D	: Matriks diagonal dari H
L	: Matriks segitiga bawah dari H
U	: Matriks segitiga atas dari H
w^k	: Bobot pada iterasi ke- k
w^{k+1}	: Bobot pada iterasi ke $k+1$

DAFTAR LAMPIRAN

Lampiran 1	Data Pima Indian Diabetes	82
Lampiran 2	Script <i>Phyton</i> Metode Gauss-Newton	82

ABSTRAK

Wyanti, Rienke Anastasya. 2025. **Penerapan Metode Gauss-Newton dan Iterasi Jacobi dalam Optimasi Regresi Logistik Biner**. Skripsi. Program Studi Matematika Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (1) Dr. Mohammad Jamhuri, M.Si. (2) Erna Herawati, M.Pd.

Kata Kunci: Gauss–Newton, Iterasi Jacobi, Regresi Logistik Biner, Optimasi, Nilai-Nilai Islam.

Regresi logistik biner merupakan metode klasifikasi yang bergantung pada proses optimasi untuk memperoleh parameter model yang akurat. Penelitian ini menerapkan metode Gauss–Newton yang dikombinasikan dengan iterasi Jacobi untuk meningkatkan efisiensi dan stabilitas dalam estimasi parameter. Model dilinearisasi pada setiap iterasi sehingga dihasilkan sistem $HW = C$ dengan pendekatan $H \approx A^T A$ dan $C \approx A^T B$, kemudian diselesaikan menggunakan iterasi Jacobi tanpa invers matriks eksplisit. Pendekatan ini dirancang untuk mempercepat konvergensi sekaligus menjaga kestabilan numerik. Pengujian pada Pima Indians Diabetes Dataset menunjukkan bahwa kombinasi Gauss–Newton dan iterasi Jacobi memberikan performa baik dengan akurasi 77.39%, precision 68.42%, recall 65.00%, dan F1-score 66.67%. Kriteria henti validasi ≥ 0.75 tercapai pada epoch ke-8 dengan waktu komputasi 7.956 ms. Dibandingkan SGDM dan RMSProp, metode ini menunjukkan konvergensi lebih cepat dan pola penurunan loss yang lebih stabil.

ABSTRACT

Wyanti, Rienke Anastasya. 2025. **Application of the Gauss-Newton Method and Jacobi Iteration in Binary Logistic Regression Optimization**. Thesis. Mathematics Science and Technology Study Program, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Supervisors: (1) Dr. Mohammad Jamhuri, M.Si. (2) Erna Herawati, M.Pd.

Keywords: Gauss–Newton, Jacobi Iteration, Binary Logistic Regression, Optimization, Islamic Values.

Binary logistic regression is a classification method that relies on the optimization process to obtain accurate model parameters. This study applied the Gauss–Newton method combined with Jacobi iterations to improve efficiency and stability in parameter estimation. The model is standardized in each iteration so that an $HW = C$ system is generated with the $H \approx A^T A$ and $C \approx A^T B$ approaches, then solved using Jacobi iterations without explicit matrix inverses. This approach is designed to accelerate convergence while maintaining numerical stability. Testing on the Pima Indians Diabetes Dataset showed that the combination of Gauss–Newton and Jacobi iterations gave good performance with an accuracy of 77.39%, accuracy of 68.42%, recall of 65.00%, and an F1-score of 66.67%. The validation stop criterion ≥ 0.75 was achieved in the 8th epoch with a compute time of 7,956 ms. Compared to SGDM and RMSProp, this method shows faster convergence and a more stable loss reduction pattern.

مستخلص البحث

وينتي، رينكي أناستاسيا. ٢٠٢٥. تطبيق طريقة غاوس-نيوتن وتكرار جاكوبي في تحسين الانحدار اللوجستي الثنائي. البحث الجامعي. قسم الرياضيات، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية في مالانغ. المشرف: (1) الدكتور محمد جمحوري، الماجستير في العلوم (2) إيرنا هيراواني، الماجستير.

الكلمات الأساسية: غاوس-نيوتن، تكرار جاكوبي، الانحدار اللوجستي الثنائي، التحسين، القيم الإسلامية.

الانحدار اللوجستي الثنائي هو طريقة تصنيف تعتمد على عملية التحسين للحصول على معلمات دقيقة للنموذج. طبقت هذه الدراسة طريقة غاوس-نيوتن مع تكرارات جاكوبي لتحسين الكفاءة والاستقرار في تقدير المعلمات. تم توحيد النموذج في كل تكرار بحيث يتم توليد النظام $HW = C$ بنهج $H \approx A^T B$ و $C \approx A^T B$ ، ثم يحل باستخدام تكرارات جاكوبي بدون معكوسات مصفوفة صريحة. تم تصميم هذا النهج لتسريع التقارب مع الحفاظ على الاستقرار العددي. أظهرت الاختبارات على مجموعة بيانات السكري لدى بيما إنديانز أن الجمع بين نسخ غاوس-نيوتن وجاكوبي أعطى أداء جيداً بدقة 77.39%، ودقة 68.42%، واستدعاء بنسبة 65.00%، ودرجة F1 بلغت 66.67%. تم تحقيق معيار إيقاف التحقق ≤ 0.75 في الحقبة الثامنة بـ 7,956 مللي ثانية. مقارنة بـ SGDM و RMSProp، أظهرت هذه الطريقة تقارباً أسرع ونمط تقليل فقدان أكثر استقراراً.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Model regresi logistik biner adalah salah satu jenis regresi non-linear dengan variabel terikat yang berbentuk variabel dikotomi, yaitu bernilai nol atau satu. Variabel bebas dalam model ini dapat berupa variabel numerik maupun kategorik dan berfungsi sebagai *covariate*. Jika suatu percobaan memiliki dua kemungkinan hasil, seperti sukses dan gagal, maka percobaan tersebut akan mengikuti distribusi binomial (Roflin et al., 2023). Untuk memperoleh estimasi parameter yang tepat, diperlukan metode optimasi yang sesuai. Proses seleksi variabel juga penting dalam regresi logistik untuk meningkatkan akurasi model dalam menggambarkan data (Salim & Alfian, 2019). Oleh karena itu, diperlukan metode optimasi yang efisien untuk mempercepat konvergensi solusi. Pendekatan melalui metode Gauss–Newton dan iterasi Jacobi dapat digunakan dalam proses optimasi regresi logistik biner.

Dalam penerapan regresi logistik biner, metode optimasi sangat berperan dalam menentukan parameter model yang optimal. Salah satu metode yang digunakan adalah metode Gauss-Newton. Metode Gauss-Newton merupakan metode yang sederhana namun sangat efisien dalam menyelesaikan masalah kuadrat terkecil non-linier (Saleh, 2010) dan (Jamhuri & Subiono, 2021). Dalam menyelesaikan permasalahan non-linier, metode Gauss-Newton digunakan untuk meminimalkan nilai dari suatu fungsi, dengan pendekatan numerik yang lebih efektif melalui proses langsung atau iteratif (Siregar et al., 2018). Untuk meningkatkan stabilitas dan efisiensi dalam proses estimasi parameter, diperlukan

metode iteratif yang dapat digunakan untuk meningkatkan konvergensi solusi. Pada penelitian ini digunakan metode Jacobi.

Dalam kajian perspektif Islam, QS. Al-Ankabut ayat 69 (Kemenag, 2024), Allah SWT menjanjikan petunjuk kepada mereka yang bersungguh-sungguh dalam menjalankan usaha dengan niat yang benar, yaitu untuk mendekatkan diri kepada-Nya.

وَالَّذِينَ جَاهَدُوا فِينَا لَنَهْدِيَنَّهُمْ سُبُلَنَا وَإِنَّ اللَّهَ لَمَعَ الْمُحْسِنِينَ ﴿٦٩﴾

Artinya : "Dan orang-orang yang bersungguh-sungguh untuk (mencari keridhaan) Kami, benar-benar akan kami tunjukkan kepada mereka jalan-jalan Kami. Dan sesungguhnya Allah SWT benar-benar beserta orang-orang yang berbuat kebaikan."

Dalam konteks proses optimasi seperti metode Gauss-Newton dan iterasi Jacobi membutuhkan kesungguhan, pengulangan, dan perbaikan secara bertahap. Konsep ini sejalan dengan semangat ijtihad dalam Islam, yang menekankan upaya maksimal dalam mencari kebenaran melalui pemikiran dan analisis yang mendalam. Sebagaimana ijtihad menuntut kesabaran dan ketekunan dalam menggali hukum dan solusi terbaik dalam berbagai permasalahan, proses optimasi dalam matematika juga mengharuskan iterasi berulang hingga mencapai konvergensi yang diharapkan. Dengan kesungguhan yang terus-menerus, baik dalam ilmu agama maupun dalam pendekatan numerik, kemajuan menuju hasil yang optimal dapat tercapai.

Beberapa penelitian sebelumnya telah mengkaji penerapan regresi logistik maupun metode numerik terkait optimasi. (Afifah, 2020) menerapkan regresi logistik pada data Survei Sosial Ekonomi Nasional (SUSENAS) 2018 di Kabupaten Mojokerto, dan menemukan bahwa anggota rumah tangga, pembelian beras miskin,

serta penggunaan telepon genggam merupakan faktor signifikan dengan akurasi klasifikasi 94,49%. Selanjutnya, (Umaroh, 2020) membandingkan regresi logistik dengan *Classification And Regression Tree* (CART) pada data Survei Sosial Ekonomi Nasional (SUSENAS) 2018 di Kota Batu, dengan hasil bahwa regresi logistik memberikan akurasi klasifikasi 94,69%, dibandingkan CART 89,36%. Penelitian lain oleh, (Sutrisno, 2020) menyoroti penggunaan *Maximum Likelihood Estimation* (MLE) pada data medis, dan menunjukkan bahwa *pendekatan Quasi-Complete Separation* dapat meningkatkan akurasi estimasi parameter. Ketiga penelitian ini menegaskan relevansi regresi logistik sebagai metode klasifikasi biner yang andal, namun juga memperlihatkan bahwa kualitas hasil sangat dipengaruhi oleh metode optimasi yang digunakan.

Selain itu, beberapa penelitian mengkaji metode numerik alternatif. (Jamhuri et al., 2022) mengevaluasi metode Gauss-Newton pada dataset *car buyer* dan *ionosphere*, dan menemukan bahwa Gauss-Newton memiliki kecepatan konvergensi lebih tinggi dibandingkan gradient descent dengan akurasi yang hampir sama. Di sisi lain, metode iterasi jacobi juga telah banyak digunakan menyelesaikan sistem persamaan linier. (Nisa & Asih, 2020) memanfaatkannya untuk memprediksi penggunaan air di Pondok Pesantren Assalafy dan berhasil menghasilkan rekomendasi konservasi berbasis pola aktivitas santri. Penelitian lain oleh (Widya et al., 2014) dan (Marzuki & Herawati, 2015) juga menunjukkan bahwa iterasi Jacobi, dengan atau tanpa prekondisi, mampu mempercepat konvergensi dan menghasilkan solusi yang optimal pada berbagai bentuk sistem persamaan linear.

Meskipun metode optimisasi seperti *Gradient Descent*, *RMSProp*, dan *Adam* lebih umum digunakan dalam pelatihan model regresi logistik, metode Gauss–Newton relatif jarang diterapkan, khususnya dalam konteks regresi logistik biner. Hal ini disebabkan karena metode Gauss–Newton lebih dikenal dalam penyelesaian masalah kuadrat terkecil non-linear daripada dalam estimasi parameter probabilistik. Oleh karena itu, penerapan metode Gauss–Newton pada regresi logistik biner menjadi menarik untuk dikaji, karena menawarkan pendekatan berbeda yang berpotensi memberikan kecepatan konvergensi lebih tinggi serta kestabilan numerik yang lebih baik. Integrasi metode ini dengan iterasi Jacobi sebagai penyelesai sistem persamaan linear diharapkan dapat memberikan kontribusi baru terhadap efisiensi proses optimasi dan akurasi hasil klasifikasi.

Berdasarkan uraian di atas, diketahui bahwa regresi logistik terbukti efektif untuk klasifikasi biner, namun performanya sangat bergantung pada metode optimasi yang digunakan. Metode Gauss–Newton menawarkan kecepatan konvergensi yang tinggi, sedangkan iterasi Jacobi memberikan efisiensi dalam penyelesaian sistem persamaan linear. Oleh karena itu, penelitian ini mengusulkan integrasi metode Gauss–Newton sebagai teknik optimasi regresi logistik dengan iterasi Jacobi sebagai penyelesai sistem persamaan pada setiap langkah iterasinya. Pendekatan ini diharapkan dapat meningkatkan stabilitas dan efisiensi estimasi parameter, sekaligus menawarkan alternatif solusi dalam permasalahan optimasi regresi logistik.

1.2 Rumusan Masalah

Berdasarkan latar belakang, diperoleh rumusan masalah sebagai berikut:

1. Bagaimana penerapan metode Gauss-Newton dan metode iterasi Jacobi dalam optimasi regresi logistik biner?
2. Bagaimana efektivitas dan efisiensi gabungan antara metode Gauss-Newton dan metode iterasi Jacobi dalam konteks regresi logistik biner?

1.3 Tujuan Penelitian

Dengan adanya permasalahan yang muncul, maka tujuan dari penelitian ini adalah

1. Menerapkan metode Gauss-Newton dan metode iterasi Jacobi dalam optimasi regresi logistik biner.
2. Menganalisis efektivitas dan efisiensi kedua metode tersebut pada regresi logistik biner.

1.4 Manfaat Penelitian

Penelitian ini diharapkan memberikan manfaat sebagai berikut:

1. Manfaat Teoritis

Memberikan kontribusi terhadap pengembangan metode numerik, khususnya penerapan Gauss-Newton dan iterasi Jacobi dalam optimasi regresi logistik. Hasil penelitian ini dapat memperkaya literatur mengenai metode alternatif optimasi dalam statistik dan komputasi.

2. Manfaat Praktis

Menjadi referensi bagi peneliti atau praktisi yang membutuhkan pendekatan efisien untuk optimasi regresi logistik, khususnya pada aplikasi yang melibatkan klasifikasi biner. Selain itu, hasil penelitian ini dapat dimanfaatkan sebagai bahan ajar atau acuan dalam pembelajaran metode numerik dan statistika terapan.

1.5 Batasan Masalah

Agar pembahasan pada penulisan ini berfokus pada masalah yang diujikan, maka penelitian ini berfokus pada batasan-batasan masalah berikut:

1. Model regresi logistik biner dengan dua kelas (label 0 dan 1), tanpa mencakup regresi logistik multinomial atau bentuk lain.
2. Pada evaluasi model hanya menggunakan metrik akurasi, presisi, recall, dan F1_Score dan tidak membahas metrik evaluasi lain.

1.6 Definisi Istilah

1. Regresi logistik biner

Regresi logistik biner adalah metode statistik yang digunakan untuk memodelkan hubungan antara variabel independen dengan variabel dependen yang bersifat biner, yaitu bernilai 0 atau 1 (Hosmer Jr et al., 2013). Model ini menggunakan fungsi sigmoid untuk mengubah kombinasi linear variabel prediktor menjadi probabilitas suatu kejadian (Menard, 2001).

2. Metode Gauss-Newton

Metode Gauss-Newton adalah teknik optimasi berbasis kuadrat terkecil yang digunakan untuk menyelesaikan masalah regresi nonlinier. Metode ini bekerja dengan melakukan pendekatan Taylor terhadap fungsi yang akan diminimalkan dan kemudian menyelesaikan sistem persamaan linear secara iteratif (Björck, 2024). Metode ini sering digunakan dalam pemrosesan sinyal, optimasi regresi, dan machine learning.

3. Metode Iterasi Jacobi

Metode iterasi Jacobi adalah metode numerik yang digunakan untuk menyelesaikan sistem persamaan linier dengan pendekatan iteratif. Setiap elemen dalam vektor solusi dihitung secara independen menggunakan nilai dari iterasi sebelumnya (Burden & Faires, 2011). Metode ini digunakan dalam berbagai aplikasi matematika komputasi, termasuk dalam optimasi regresi logistik (Saad, 2003).

4. Optimasi dalam regresi logistik

Optimasi dalam regresi logistik bertujuan untuk menemukan parameter model yang memaksimalkan likelihood atau meminimalkan fungsi *loss*. Berbagai teknik seperti metode Newton-Raphson, metode Gauss-Newton, dan algoritma berbasis gradien digunakan untuk tujuan ini (Bishop & Nasrabadi, 2006).

5. Data pima indian diabetes

Dataset Pima Indian Diabetes berisi data medis yang dikumpulkan oleh *National Institute of Diabetes and Digestive and Kidney Diseases*. Dataset ini digunakan untuk memprediksi kemungkinan seseorang menderita diabetes

berdasarkan fitur kesehatan seperti kadar glukosa dan tekanan darah (Smith et al., 1988).

6. Matriks Jacobian

Matriks Jacobian adalah matriks yang terdiri dari turunan parsial suatu fungsi terhadap setiap variabelnya. Dalam konteks optimasi regresi logistik, matriks ini digunakan dalam metode Gauss-Newton untuk memperkirakan parameter model (Press, 2007).

7. Konvergensi metode iteratif

Konvergensi dalam metode iteratif merujuk pada proses pendekatan solusi hingga mencapai toleransi error yang diinginkan. Faktor-faktor seperti nilai awal, toleransi, dan sifat matriks mempengaruhi kecepatan konvergensi metode iterasi Jacobi (Quarteroni et al., 2010).

8. Metode numerik dalam machine learning

Metode numerik seperti Gauss-Newton dan Jacobi sering digunakan dalam pembelajaran mesin (machine learning) untuk mengoptimalkan parameter model. Penggunaan metode ini dapat meningkatkan efisiensi komputasi dalam algoritma regresi logistik dan deep learning (Goodfellow, 2016).

BAB II

KAJIAN TEORI

2.1 Teori Pendukung

2.1.1 Regresi Logistik Biner

Regresi logistik biner adalah metode statistika yang digunakan untuk menentukan model terbaik yang menjelaskan hubungan antara variabel respon (y) yang bersifat biner atau dikotomis dengan variabel prediktor (x). Variabel prediktor tersebut bersifat kualitatif, kuantitatif atau kombinasi keduanya. Variabel respon y terdiri dari dua kategori yaitu “Tinggi” dan “Rendah” yang direpresentasikan sebagai $y = 1$ untuk (Tinggi) dan $y = 0$ untuk (Rendah) (Rahmadani et al., 2023).

Model regresi logistik biner sering diterapkan dalam berbagai bidang aplikasi, termasuk dalam bidang kesehatan, industri, kependudukan, dan lain-lainnya. Regresi logistik biner digunakan untuk menganalisis pada data respon yang memiliki dua kategori. Ketika variabel prediktor dengan skala campuran mengalami masalah multikolinearitas, pendekatan statistik yang dapat digunakan untuk mengatasinya adalah komponen utama non-linear (Islamiyati, 2015).

Diberikan data pelatihan $(\vec{x}_i, y_i), \forall i = 1, 2, \dots, r$. Setiap vektor $\vec{x}_i \in \mathbb{R}^n$ merepresentasikan fitur, sedangkan label $y_i \in \{1, 0\}$ menunjukkan kelas dari sampel ke- i . Secara umum data tersebut dapat dituliskan dalam bentuk matriks fitur dan matriks label sebagai berikut:

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{r1} & x_{r2} & \cdots & x_{rn} \end{bmatrix}, Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_r \end{bmatrix} \quad (2.1)$$

Untuk memudahkan perhitungan, setiap vektor prediktor diperluas menjadi $\vec{x}_i = (1, x_{i,1}, \dots, x_{i,n})$ dengan elemen pertama berupa konstanta 1 yang berfungsi sebagai bias (*intercept*). Parameter yang akan diestimasi ditulis sebagai $\vec{w} = (w_0, w_1, \dots, w_n)$.

Fungsi dasar dari regresi logistik biner adalah model probabilistik yang dinyatakan melalui fungsi sigmoid sebagai berikut:

$$h(z_i) = \frac{1}{1 + \exp(-z_i)} \quad (2.2)$$

di mana:

$$z_i = w_0 x_{i,0} + w_1 x_{i,1} + \cdots + w_n x_{i,n} \quad (2.3)$$

Dalam hal ini, w_0 adalah intersep, sedangkan w_1, w_2, \dots, w_n adalah koefisien regresi yang menentukan pengaruh dari masing-masing variabel independen. Tujuan utama dari regresi logistik adalah menemukan nilai parameter \vec{w} yang paling sesuai, sehingga model mampu memprediksi probabilitas suatu sampel termasuk ke dalam kelas positif ($y = 1$) atau negatif ($y = 0$) secara lebih akurat berdasarkan variabel-variabel prediktor.

Dalam optimasi regresi logistik, salah satu tujuan utama adalah meminimalkan fungsi loss untuk mendapatkan parameter-parameter minimum yang sesuai dengan data. Karena model logistik bersifat non-linear, proses ini sering kali memerlukan pendekatan numerik. Fungsi loss (fungsi biaya) pada regresi logistik biner dinyatakan sebagai berikut:

$$L(\vec{w}) = -\frac{1}{r} \sum_{i=1}^r (y_i \log(h(z_i)) + (1 - y_i) \log(1 - h(z_i))) \quad (2.4)$$

Untuk meminimalkan Persamaan 2.4, digunakan metode gradient descent agar diperoleh pembaruan bobot yang menghasilkan nilai minimum sebagai berikut:

$$w_j^{k+1} = w_j^k - \alpha \frac{\partial L(\vec{w})}{\partial w_j} \quad (2.5)$$

Dengan α adalah *learning rate*, dan turunan parsial dari fungsi loss terhadap w_j diberikan oleh:

$$\frac{\partial L(\vec{w})}{\partial w_j} = -\frac{1}{r} \sum_{i=1}^r \left(y_i \frac{\partial \log(h(z_i))}{\partial w_j} + (1 - y_i) \frac{\partial \log(1 - h(z_i))}{\partial w_j} \right) \quad (2.6)$$

Untuk menghitung dua turunan parsial di dalam tanda kurung pada Persamaan 2.6, digunakan hasil dari Persamaan 2.7 dan 2.8 berikut:

$$\begin{aligned} \frac{\partial \log(h(z_i))}{\partial w_j} &= \frac{\partial \log h(z_i)}{\partial h(z_i)} \frac{\partial h(z_i)}{\partial z_i} \frac{\partial z_i}{\partial w_j} \\ &= \frac{1}{h(z_i)} h(z_i)(1 - h(z_i)) x_{i,j} \\ &= (1 - h(z_i)) x_{i,j} \end{aligned} \quad (2.7)$$

$$\begin{aligned} \frac{\partial \log(1 - h(z_i))}{\partial w_j} &= \frac{\partial \log h(z_i)}{\partial (1 - h(z_i))} \frac{\partial (1 - h(z_i))}{\partial z_i} \frac{\partial z_i}{\partial w_j} \\ &= \frac{1}{(1 - h(z_i))} (-h(z_i))(1 - h(z_i)) x_{i,j} \\ &= (-h(z_i)) x_{i,j} \end{aligned} \quad (2.8)$$

Dengan mensubstitusikan hasil pada Persamaan 2.7 dan 2.8 ke dalam Persamaan 2.6, diperoleh:

$$\frac{\partial L(\vec{w})}{\partial w_j} = -\frac{1}{r} \sum_{i=1}^r (y_i(1 - h(z_i))x_{ij} + (1 - y_i)(-h(z_i))x_{ij}) \quad (2.9)$$

Bentuk ini masih dapat disederhanakan dengan menggabungkan suku-suku yang sejenis. Setelah dilakukan penyederhanaan aljabar, diperoleh:

$$\frac{\partial L(\vec{w})}{\partial w_j} = -\frac{1}{r} \sum_{i=1}^r (h(z_i) - y_i)x_{ij}, \forall j = 0, 1, \dots, n \quad (2.10)$$

Hasil pada Persamaan 2.10 menunjukkan bahwa gradien fungsi loss logistik ditentukan oleh rata-rata selisih antara probabilitas prediksi $h(z_i)$ dan label aktual y_i , yang kemudian ditimbang oleh nilai fitur x_{ij} dari setiap sampel. Dengan demikian, gradien tidak hanya mengukur perbedaan prediksi terhadap label, tetapi juga mempertimbangkan kontribusi masing-masing fitur dalam memengaruhi parameter w_j .

Berdasarkan gradien ini, parameter diperbarui secara iteratif menggunakan algoritma gradient descent:

$$w_j^{k+1} = w_j^k - \alpha \sum_{i=1}^r (y_i - h(z_i))x_{i,j}, \forall j = 0, 1, \dots, n \quad (2.11)$$

Meskipun algoritma gradient descent merupakan pendekatan paling umum untuk optimasi regresi logistik, metode ini sering kali memerlukan jumlah iterasi yang besar untuk mencapai konvergensi, terutama pada dataset dengan dimensi tinggi. Oleh karena itu, dikembangkan metode alternatif yang lebih efisien dalam mempercepat proses estimasi parameter. Salah satu metode yang akan digunakan untuk tujuan ini adalah metode Gauss–Newton, yang akan dibahas pada subbagian berikutnya.

2.1.2 Metode Gauss-Newton

Algoritma Gauss-Newton atau disebut juga metode linierisasi adalah suatu prosedur iterasi yang digunakan untuk memecahkan masalah kuadrat terkecil (*least-square*). Metode Gauss-Newton meminimalkan suatu fungsi obyektif dengan syarat orde pertama (*first order condition*). Metode Gauss-Newton adalah versi modifikasi metode Newton yang memanfaatkan gradien dan Hessian dari fungsi jumlah kuadrat residu (Saleh, 2010).

Dalam regresi logistik biner, metode Gauss-Newton digunakan untuk memperkirakan parameter-parameter model. Dan metode Gauss-Newton bekerja dengan meminimalkan fungsi kesalahan (*error function*) antara prediksi model dan nilai data yang diamati.

Untuk mengoptimalkan fungsi biaya menggunakan metode Gauss-Newton, sistem persamaan linear diatur dan diselesaikan pada setiap iterasi. *Error* untuk data ke- i didefinisikan sebagai berikut:

$$L_i = -(y_i \log(h(z_i)) + (1 - y_i) \log(1 - h(z_i))) \quad (2.12)$$

Untuk menerapkan metode Gauss-Newton, fungsi non-linear di dalam L_i yaitu $\log(h(z_i))$ dan $\log(1 - h(z_i))$, didekati menggunakan ekspansi deret Taylor orde pertama di sekitar nilai parameter \vec{w} . Ekspansi deret Taylor tersebut dapat ditulis sebagai berikut:

$$\begin{aligned} \log(h(z_i) + \Delta z_i) &\approx \log(h(z_i)) + \sum_{j=0}^n \frac{\partial \log(h(z_i))}{\partial w_j} \Delta w_j \\ \log((1 - h(z_i)) + \Delta z_i) &\approx \log(1 - h(z_i)) + \sum_{j=0}^n \frac{\partial \log(1 - h(z_i))}{\partial w_j} \Delta w_j \end{aligned} \quad (2.13)$$

Turunan dari fungsi $\log(h(z_i))$ dan $\log(1 - h(z_i))$ terhadap w_j , sebagaimana telah diturunkan pada Persamaan 2.7 dan 2.8, diberikan oleh:

$$\begin{aligned}\frac{\partial \log(h(z_i))}{\partial w_j} &= (1 - h(z_i))x_{ij} \\ \frac{\partial \log(1 - h(z_i))}{\partial w_j} &= -h(z_i)x_{ij}\end{aligned}\tag{2.14}$$

Dengan mensubstitusi Persamaan 2.14 ke dalam Persamaan 2.13, diperoleh bentuk linear dari fungsi loss individu pada setiap data ke- i :

$$L_i \approx -y_i (\log(h(z_i)) + \sum_{j=0}^n (1 - h(z_i))x_{ij}\Delta w_j) - (1 - y_i)(\log(1 - h(z_i)) - \sum_{j=0}^n h(z_i)x_{ij}\Delta w_j)$$

Suku-suku diatas dapat dikelompokkan sehingga diperoleh:

$$(y_i(1 - h(z_i)) - (1 - y_i)h(z_i)) \sum_{j=0}^n x_{ij}\Delta w_j = -(y_i \log(h(z_i)) + (1 - y_i) \log(1 - h(z_i)))$$

Selanjutnya, bagian di dalam tanda kurung pada ruas kiri dapat disederhanakan karena:

$$y_i(1 - h(z_i)) - (1 - y_i)h(z_i) = y_i - h(z_i)$$

sehingga diperoleh hubungan linear berikut:

$$(y_i - h(z_i)) \sum_{j=0}^n x_{ij}\Delta w_j = -(y_i \log(h(z_i)) + (1 - y_i) \log(1 - h(z_i)))\tag{2.15}$$

Persamaan 2.15 menggambarkan hubungan linear antara perubahan bobot $\Delta \vec{w}$ dan fungsi loss individu L_i . Jika persamaan ini diterapkan untuk seluruh data $i = 1, 2, \dots, r$, maka terbentuk system persamaan linear sebagai berikut:

$$\begin{aligned}(y_1 - h(z_1))\vec{x}_1 \cdot \Delta \vec{w} &= L_1 \\ (y_2 - h(z_2))\vec{x}_2 \cdot \Delta \vec{w} &= L_2 \\ &\vdots \\ (y_r - h(z_r))\vec{x}_r \cdot \Delta \vec{w} &= L_r\end{aligned}$$

Sistem tersebut dapat ditulis sebagai persamaan matriks berikut

$$\begin{bmatrix} y_1 - h(z_1) & (y_1 - h(z_1))x_{1,1} & \cdots & (y_1 - h(z_1))x_{1,r} \\ y_2 - h(z_2) & (y_2 - h(z_2))x_{2,1} & \cdots & (y_2 - h(z_2))x_{2,r} \\ \vdots & \vdots & \ddots & \vdots \\ y_r - h(z_r) & (y_r - h(z_r))x_{r,1} & \cdots & (y_r - h(z_r))x_{r,n} \end{bmatrix} \begin{bmatrix} \Delta w_0 \\ \Delta w_1 \\ \vdots \\ \Delta w_n \end{bmatrix} = \begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_r \end{bmatrix} \quad (2.16)$$

Sistem akan mempunyai satu solusi, apabila jumlah data $r = n$. Jika $r < n$ maka sistem akan mempunyai banyak solusi, sedangkan untuk $r > n$ solusi diperoleh melalui pendekatan kuadrat terkecil (*least squares*). Dengan menerapkan prinsip kuadrat terkecil, diperoleh system persamaan normal sebagai berikut:

$$A^T A W = A^T B \quad (2.17)$$

dengan

$$A = \begin{bmatrix} y_1 - h(z_1) & (y_1 - h(z_1))x_{1,1} & \cdots & (y_1 - h(z_1))x_{1,r} \\ y_2 - h(z_2) & (y_2 - h(z_2))x_{2,1} & \cdots & (y_2 - h(z_2))x_{2,r} \\ \vdots & \vdots & \ddots & \vdots \\ y_r - h(z_r) & (y_r - h(z_r))x_{r,1} & \cdots & (y_r - h(z_r))x_{r,n} \end{bmatrix}$$

$$W = \begin{bmatrix} \Delta w_0 \\ \Delta w_1 \\ \vdots \\ \Delta w_n \end{bmatrix} \quad (2.18)$$

$$B = \begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_r \end{bmatrix}$$

Misalkan pada persamaan $A^T A = H$ dan $A^T B = C$, maka sistem persamaan linear tersebut menjadi

$$HW = C \quad (2.19)$$

Kemudian, Langkah penyelesaian sistem linear pada Persamaan 2.19 selanjutnya akan dilakukan menggunakan metode Iterasi Jacobi, yang akan dijelaskan pada subbab berikutnya.

2.1.3 Metode Iterasi Jacobi

Dalam aljabar linear telah dibahas berbagai metode untuk menyelesaikan Sistem Persamaan Linear, salah satunya adalah metode langsung (*direct method*). Dalam tulisan ini akan dibahas pendekatan lain untuk menyelesaikan Sistem Persamaan Linear, yaitu metode tidak langsung yang dikenal sebagai metode iteratif. Metode iteratif dimulai dengan perkiraan awal, diikuti dengan siklus perhitungan yang diulang-ulang hingga mencapai tingkat ketelitian yang diharapkan. Banyaknya perhitungan dalam metode iteratif bergantung pada tingkat akurasi yang diinginkan (Darmono, 2012).

Metode Jacobi adalah bentuk iterasi titik tetap untuk sistem persamaan linear. Langkah pertama adalah menulis ulang persamaan, memecahkan yang tidak diketahui. Metode Jacobi melakukannya dengan cara standar berikut; memecahkan persamaan ke- i yang tidak diketahui. Kemudian, iteratif seperti dalam titik tetap, dimulai dengan tebakan awal (Sauer, 2017).

Salah satu syarat penting agar metode Jacobi dijamin konvergen adalah matriks koefisien harus memiliki sifat dominan diagonal.

Definisi 2.9 Matriks $n \times n$, matriks $A = a_{ij}$ secara ketat bersifat dominan diagonal jika, untuk setiap $1 \leq i \leq n$ berlaku:

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}|$$

Artinya, elemen diagonal utama a_{ii} pada setiap baris harus lebih besar daripada jumlah semua elemen lain pada baris tersebut (Sauer, 2017).

Metode Jacobi merupakan bentuk iterasi titik tetap yang dapat diterapkan pada sistem persamaan linear hasil metode Gauss-Newton pada Persamaan 2.17, yaitu:

$$HW = C \quad (2.20)$$

dimana:

H adalah matriks koefisien berukuran $(n + 1) \times (n + 1)$, W adalah vektor solusi berukuran $(n + 1) \times 1$, C adalah vektor hasil berukuran $(n + 1) \times 1$.

Untuk mempermudah proses iterasi, Matriks H dapat diuraikan menjadi tiga komponen, yaitu:

$$H = D + L + U \quad (2.21)$$

dimana:

D adalah matriks diagonal dari H , L adalah matriks segitiga bawah (tanpa elemen diagonal), U adalah matriks segitiga atas (tanpa elemen diagonal).

Dengan substitusi dekomposisi tersebut, sistem $HW = C$ dapat dituliskan menjadi:

$$(D + L + U)W = C \quad (2.22)$$

Yang kemudian dapat disusun ulang sebagai:

$$DW = C - (L + U)W \quad (2.23)$$

Persamaan 2.23 menjadi dasar untuk membentuk rumus iterasi Jacobi. Solusi W diperbarui pada setiap iterasi berdasarkan formula berikut:

$$w^{k+1} = D^{-1}(C - (L + U)w^k) \quad (2.24)$$

Untuk memeriksa konvergensi, iterasi dihentikan berdasarkan capaian akurasi validasi. Proses iterasi dilanjutkan hingga nilai akurasi validasi (validation accuracy) mencapai 0,75 (75%), yang ditetapkan sebagai batas ketelitian atau kriteria konvergensi model.

Dengan demikian, metode iterasi Jacobi berfungsi sebagai solver numerik yang efisien untuk sistem linear yang muncul dalam setiap langkah Gauss–Newton. Integrasi ini membuat keseluruhan proses optimasi menjadi lebih praktis, karena menghindari komputasi invers matriks yang mahal namun tetap menjaga konvergensi menuju solusi optimum. Berdasarkan uraian teori mengenai regresi logistik, metode Gauss–Newton, dan iterasi Jacobi, dapat dilihat bahwa ketiganya saling melengkapi dalam membentuk suatu kerangka optimasi yang koheren. Oleh karena itu, bagian berikut merangkum keterkaitan teoritis tersebut serta menjelaskan implikasinya bagi metodologi penelitian ini.

2.2 Integrasi Al-Qur'an dalam Teori Optimasi

Dalam Islam, ilmu pengetahuan merupakan bagian yang tak terpisahkan dari ajaran agama. Al-Qur'an, sebagai pedoman hidup umat manusia, memberikan landasan filosofis yang mendorong umatnya untuk terus belajar, memahami, dan mengaplikasikan ilmu demi kemaslahatan. Ilmu pengetahuan memiliki kedudukan yang tinggi dalam Islam. Sebagaimana disebutkan dalam QS. Al-Mujadilah: 11 (Kemenag, 2024), Allah SWT berfirman:

يَا أَيُّهَا الَّذِينَ آمَنُوا إِذَا قِيلَ لَكُمْ تَفَسَّحُوا فِي الْمَجَالِسِ فَافْسَحُوا يَفْسَحِ اللَّهُ لَكُمْ وَإِذَا قِيلَ انشُرُوا فَانْشُرُوا يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنْكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ وَاللَّهُ بِمَا تَعْمَلُونَ خَبِيرٌ ﴿١١﴾

Artinya: "Wahai orang-orang yang beriman, apabila dikatakan kepadamu "Berilah kelapangan di dalam majelis-majelis," lapangkanlah, niscaya Allah SWT akan memberi kelapangan untukmu. Apabila dikatakan, "Berdirilah," (kamu) berdirilah. Allah SWT niscaya akan mengangkat orang-orang yang beriman di antaramu dan orang-orang yang diberi ilmu beberapa derajat. Allah SWT Maha teliti terhadap apa yang kamu kerjakan."

Ayat ini mengajarkan bahwa orang-orang berilmu memiliki tanggung jawab untuk menggunakan pengetahuannya demi kemaslahatan. Ilmu yang diamalkan dengan ikhlas dapat menjadi sarana ibadah dan menambah keberkahan, karena setiap usaha dalam memahami dan mengaplikasikan ilmu, sejatinya adalah wujud penghambaan kepada Allah SWT. Upaya memahami pola-pola dalam data melalui metode regresi logistik mencerminkan usaha untuk memaksimalkan potensi akal yang merupakan karunia Allah SWT.

Sebagaimana ilmu memiliki kedudukan yang tinggi dalam Islam, tidak hanya sebagai alat untuk memecahkan masalah duniawi tetapi juga sebagai sarana mendekatkan diri kepada Allah SWT. Penerapan gabungan metode Gauss-Newton dan metode iterasi Jacobi ke dalam regresi logistik biner adalah wujud pemanfaatan ilmu pengetahuan yang dianugerahkan oleh Allah SWT. Metode ini tidak hanya membantu menyelesaikan masalah teknis seperti analisis data yang kompleks, tetapi juga mencerminkan bentuk ibadah melalui penggunaan akal untuk memahami dan mengelola ciptaan Allah SWT.

Maka dari itu, penerapan ilmu dalam analisis regresi logistik biner ini, yang berfokus pada optimasi dan pemahaman data, menjadi salah satu cara untuk mengembangkan akal dan meningkatkan kualitas hidup umat manusia. Oleh karena itu, setiap langkah dalam penelitian ini dapat dianggap sebagai upaya untuk meningkatkan pemahaman dan keimanan kepada Allah SWT melalui aplikasi ilmu yang bermanfaat, yang pada gilirannya membawa berkah bagi peneliti dan masyarakat.

Dalam QS. Al-Ankabut: 43 (Kemenag, 2024), Allah SWT memberikan berbagai perumpamaan dan pelajaran dalam Al-Qur'an agar manusia dapat memahami makna yang mendalam. Allah SWT berfirman:

وَتِلْكَ الْأَمْثَالُ نَضْرِبُهَا لِلنَّاسِ وَمَا يَعْقِلُهَا إِلَّا الْعُلَمَاءُ ﴿٤٣﴾

Artinya: "Dan perumpamaan-perumpamaan ini Kami buat untuk manusia; dan tiada yang memahaminya kecuali orang-orang yang berilmu."

Ayat ini mengajarkan bahwa pemahaman mendalam terhadap perumpamaan-perumpamaan tersebut tidak hanya memberikan manfaat praktis tetapi juga mengarahkan manusia untuk mengagumi kebesaran Allah SWT. Dalam konteks penelitian ini, pola-pola data yang diungkapkan melalui metode regresi logistik adalah bukti betapa teraturnya ciptaan Allah SWT. Dengan memahami pola ini, manusia tidak hanya memperoleh ilmu tetapi juga semakin menyadari kebesaran Allah SWT yang menciptakan hukum-hukum alam secara sempurna. Oleh karena itu, dapat dipandang sebagai bentuk tafakur yang mendalam terhadap sunnatullah melalui pendekatan ilmiah yang bermanfaat.

Dan memberikan pelajaran bahwa perumpamaan yang disampaikan Allah SWT dapat ditemukan dalam fenomena alam, pola kehidupan, dan bahkan model matematis yang ditemukan oleh manusia. Dalam metode Regresi logistik biner, model ini dapat dipandang sebagai salah satu cara manusia untuk memahami hubungan sebab-akibat dalam data. Melalui metode Gauss-Newton dan metode iterasi Jacobi, optimasi model ini menjadi lebih efektif, sehingga memberikan gambaran yang lebih akurat tentang pola yang terkandung dalam data.

Optimasi regresi logistik biner dalam penelitian ini yang dapat dianggap sebagai refleksi sunnatullah, yaitu hukum-hukum Allah SWT yang berlaku di alam semesta.

Metode matematis seperti metode Gauss-Newton dan metode iterasi Jacobi adalah cara manusia untuk memahami hukum-hukum tersebut, baik dalam konteks data maupun dalam fenomena nyata. Allah SWT menciptakan alam semesta dengan sistem yang teratur, dan manusia diberi kemampuan untuk mempelajarinya melalui akal. Oleh karena itu, penelitian ini bukan hanya sekadar analisis matematis, tetapi juga upaya untuk memahami kebesaran Allah SWT melalui hukum-hukum-Nya.

Dalam Islam, kejujuran dalam menyampaikan informasi, termasuk dalam pengambilan dan penggunaan data, sangat ditekankan. Sebagaimana Allah SWT berfirman pada QS. Al-Isra': 36 (Kemenag, 2024):

وَلَا تَقْفُ مَا لَيْسَ لَكَ بِهِ عِلْمٌ إِنَّ السَّمْعَ وَالْبَصَرَ وَالْفُؤَادَ كُلُّ أُولَٰئِكَ كَانَ عَنْهُ مَسْئُولًا ﴿٣٦﴾

Artinya: "Janganlah engkau mengikuti sesuatu yang tidak kau ketahui. Sesungguhnya pendengaran, penglihatan, dan hati nurani semua itu akan diminta pertanggungjawabannya."

Ayat ini mengajarkan bahwa setiap informasi yang digunakan harus didasarkan pada pengetahuan yang benar dan tidak boleh sembarangan mengikuti data atau kesimpulan yang tidak memiliki dasar ilmiah. prinsip dalam ayat ini, proses iteratif dalam metode ini membutuhkan ketelitian, validasi, dan justifikasi ilmiah yang kuat. Jika suatu data atau perhitungan diabaikan tanpa dasar yang jelas, maka hasil yang diperoleh bisa menyesatkan dan tidak mencerminkan realitas sebenarnya. Oleh karena itu, pendekatan matematis dalam optimasi ini harus dilakukan dengan prinsip kehati-hatian dan ketelitian agar tidak mengikuti kesimpulan yang salah.

Selain itu, ayat ini menekankan pentingnya pertanggungjawaban atas informasi yang diperoleh dan digunakan. Dalam konteks penelitian ini, seorang peneliti harus memastikan bahwa data yang digunakan telah melalui proses validasi yang ketat dan metode yang diterapkan sesuai dengan prinsip ilmiah yang benar. Kesalahan dalam

proses analisis atau manipulasi data dapat berakibat pada hasil yang tidak dapat dipertanggungjawabkan secara akademik maupun moral. Oleh karena itu, metode Gauss-Newton dan iterasi Jacobi harus diterapkan dengan dasar yang kuat, menggunakan perhitungan yang transparan dan dapat diuji kembali.

Dengan demikian, ayat ini memberikan pelajaran berharga dalam proses penelitian ilmiah, terutama dalam penerapan metode optimasi seperti Gauss-Newton dan iterasi Jacobi. Ayat ini mengajarkan bahwa dalam setiap tahap pengolahan data dan analisis, seorang peneliti harus bertanggung jawab atas hasil yang diperoleh dan tidak boleh mengambil kesimpulan yang tidak didukung oleh fakta yang valid. Dengan menerapkan prinsip ini, penelitian yang dilakukan akan memiliki nilai keilmuan yang lebih tinggi serta dapat memberikan manfaat yang lebih besar bagi perkembangan ilmu pengetahuan.

2.3 Kajian Topik dengan Teori Pendukung

Regresi logistik biner adalah metode statistik yang digunakan untuk memodelkan hubungan antara variabel independen dengan variabel dependen biner (dua kelas), seperti "ya/tidak" atau "sukses/gagal" (Hosmer Jr et al., 2013). Model ini berbasis fungsi sigmoid yang mengubah kombinasi linear variabel prediktor menjadi probabilitas kejadian suatu kategori. Regresi logistik biner sering digunakan dalam berbagai bidang seperti kesehatan, keuangan, dan kecerdasan buatan, terutama dalam klasifikasi data (Menard, 2001). Optimasi parameter dalam regresi logistik dilakukan dengan metode numerik karena model ini tidak memiliki solusi eksak dalam bentuk analitik.

Metode Gauss-Newton adalah salah satu teknik numerik yang digunakan untuk menyelesaikan masalah optimasi, termasuk dalam regresi logistik biner. Metode ini

merupakan pendekatan kuadrat terkecil yang menyederhanakan perhitungan dengan mengabaikan elemen non-linear dari matriks Hessian, sehingga lebih efisien dibandingkan metode newton-raphson (Björck, 2024). Dengan iterasi bertahap, metode ini membantu menemukan parameter yang meminimalkan fungsi kesalahan dalam model regresi logistik. Keunggulan Gauss-Newton terletak pada konvergensi yang lebih cepat dibandingkan metode berbasis gradien, namun tetap memiliki keterbatasan dalam menangani sistem persamaan yang sangat besar.

Di sisi lain, iterasi Jacobi adalah metode numerik yang digunakan untuk menyelesaikan sistem persamaan linear yang sering muncul dalam setiap langkah optimasi Gauss-Newton. Metode ini merupakan pendekatan iteratif yang memperbarui nilai variabel solusi secara paralel berdasarkan nilai sebelumnya (Saad, 2003). Iterasi Jacobi sangat efektif dalam menangani sistem berukuran besar dan jarang (sparse), yang sering dijumpai dalam model regresi logistik dengan banyak parameter (Quarteroni et al., 2010). Dengan menggunakan metode ini, estimasi parameter regresi logistik dapat dilakukan secara lebih efisien tanpa perlu melakukan invers matriks langsung, yang sering kali mahal secara komputasi.

Dalam optimasi parameter regresi logistik biner, perhitungan estimasi parameter sering kali melibatkan penyelesaian sistem persamaan non-linear yang kompleks. Metode Gauss-Newton digunakan untuk mempercepat konvergensi terhadap solusi optimal, karena metode ini berbasis pendekatan kuadrat terkecil yang secara efisien menangani sistem non-linear. Sementara itu, metode Iterasi Jacobi diterapkan dalam penyelesaian sistem linear yang muncul dalam setiap iterasi algoritma Gauss-Newton untuk mengurangi kompleksitas komputasi dan menghindari inversi matriks langsung.

BAB III

METODE PENELITIAN

3.1 Jenis Penelitian

Jenis penelitian ini termasuk dalam kategori penelitian kuantitatif, di mana metode numerik Gauss-Newton dan iterasi Jacobi diterapkan dan dievaluasi untuk optimasi regresi logistik biner. Penelitian ini menggunakan pendekatan eksperimen, yang bertujuan untuk menilai efektivitas dan efisiensi dari gabungan metode Gauss-Newton dan iterasi Jacobi dalam menyelesaikan masalah optimasi parameter regresi logistik biner.

3.2 Data dan Sumber Data

Dataset yang digunakan dalam penelitian ini merupakan data sekunder, yaitu data yang telah dikumpulkan. Data sekunder ini berupa *Pima Indian Diabetes Dataset*, yang diperoleh dari repositori publik Kaggle melalui tautan berikut:

<https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database>.

Dataset ini banyak digunakan dalam penelitian terkait klasifikasi medis, khususnya untuk diagnosis penyakit diabetes. Data ini berisi data medis perempuan keturunan suku Pima Indian, dengan total 768 sampel. Setiap sampel memiliki 8 variabel prediktor numerik yang mencerminkan kondisi medis pasien, serta 1 variabel target biner (*Outcome*) yang menunjukkan status diagnosis diabetes. Adapun deskripsi setiap fitur pada dataset tersebut yaitu sebagai berikut:

1. *Pregnancies* (Jumlah kehamilan)
2. *Glucose* (Kadar glukosa plasma)

3. *Blood Pressure* (Tekanan darah diastolic)
4. *SkinThickness* (Ketebalan lipatan kulit trisep)
5. *Insulin* (Konsentrasi insulin serum)
6. *BMI* (Indeks massa tubuh)
7. *Diabetes Pedigree Function* (Riwayat diabetes keluarga)
8. *Age* (Usia)

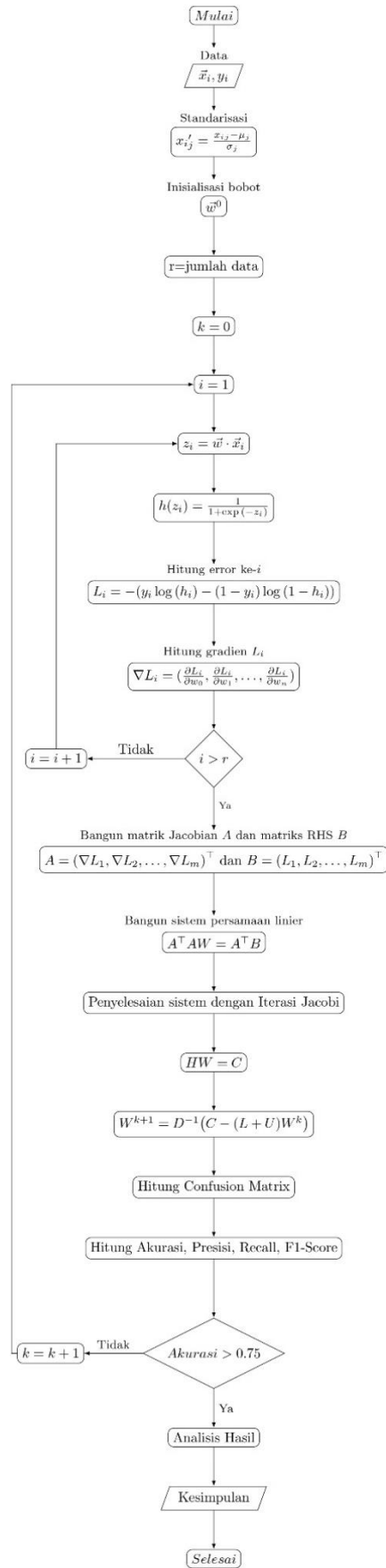
Selain kedelapan fitur prediktor tersebut, dataset ini juga memiliki variabel target biner yang disebut *Outcome*, dengan ketentuan sebagai berikut

- a. *Outcome* = 1 merupakan pasien terdiagnosis diabetes
- b. *Outcome* = 0 merupakan pasien tidak terdiagnosis diabetes

Seluruh fitur pada dataset bersifat numerik, artinya setiap kolom berisi nilai hasil pengukuran medis dan tidak terdapat nilai kategorikal seperti teks atau label. Dengan demikian, dataset ini sangat sesuai untuk digunakan dalam pengujian efektivitas metode Gauss–Newton dan iterasi Jacobi dalam optimasi regresi logistik biner.

3.3 Langkah-langkah Penyelesaian

Subbab ini menjelaskan tahapan penyelesaian penelitian secara sistematis, mulai dari pengolahan data hingga evaluasi hasil klasifikasi. Seluruh tahapan penelitian dirangkum dalam *flowchart* 3.1 yang menggambarkan alur kerja metode secara keseluruhan. Berikut ditampilkan flowchart tahapan penelitian yang digunakan dalam penerapan metode Gauss–Newton dan iterasi Jacobi pada regresi logistik biner.



Gambar 3.1 Flowchart Metode Gauss-Newton dan Iterasi Jacobi

Flowchart pada Gambar 3.1 menunjukkan bahwa proses penelitian diawali dengan pengambilan data, dilanjutkan dengan tahap pra-pemrosesan data, pembagian data, pelatihan model menggunakan metode Gauss–Newton dan iterasi Jacobi, hingga tahap evaluasi kinerja model. Setiap tahapan tersebut dijelaskan secara rinci pada subbab berikutnya.

3.3.1 Praproses Data

Praproses data merupakan tahapan awal dalam analisis yang bertujuan untuk memastikan kualitas data sebelum digunakan dalam model. Beberapa langkah yang dilakukan dalam praproses data meliputi:

1. Pembagian Data

Membagi data menjadi set pelatihan (70%) digunakan untuk melatih model, validasi (15%) digunakan untuk menyetel parameter model dan test (15%) digunakan menguji generalisasi model.

2. Standarisasi Fitur

Standarisasi fitur merupakan salah satu teknik praproses data yang digunakan untuk menyamakan skala antar variabel numerik. Standarisasi mengubah distribusi data setiap fitur sehingga memiliki rata-rata dan simpangan baku.

Secara matematis, transformasi standarisasi dituliskan sebagai:

$$x'_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j}$$

dimana x'_{ij} adalah nilai setelah standarisasi, x_{ij} adalah nilai asli fitur ke- j pada sampel ke- i , μ_j adalah rata-rata nilai fitur ke- j , σ_j adalah simpangan baku fitur ke- j .

3.3.2 Penerapan Regresi Logistik Biner

Regresi logistik biner digunakan untuk memodelkan probabilitas suatu kejadian berdasarkan variabel prediktor. Tahapan penyelesaiannya meliputi:

1. Menghitung fungsi sigmoid

Fungsi sigmoid yang digunakan untuk memetakan nilai input z_i ke dalam rentang probabilitas $[0,1]$. Fungsi yang dirumuskan sebagai:

$$h(z_i) = \frac{1}{1 + \exp(-z_i)}$$

dengan

$$z_i = w_0x_{i,0} + w_1x_{i,1} + \dots + w_nx_{i,n}$$

2. Menghitung fungsi *loss*

Fungsi *loss* digunakan untuk mengukur seberapa baik model dalam memprediksi kelas target. Fungsi *loss* pada regresi logistik berbasis log-likelihood negative dirumuskan sebagai:

$$L(\vec{w}) = -\frac{1}{r} \sum_{i=1}^r (y_i \log(h(z_i)) + (1 - y_i) \log(1 - h(z_i)))$$

di mana:

$y_i \in \{0,1\}$ adalah label kelas untuk sampel ke-i, $h(z_i)$ adalah nilai probabilitas hasil prediksi model untuk sampel ke-i, r adalah jumlah total sampel.

3. Meminimalkan fungsi *loss* melalui pembaruan bobot

Untuk mendapatkan parameter optimal, bobot w_j diperbarui secara iteratif menggunakan metode gradien dengan persamaan:

$$w_j^{k+1} = w_j^k - \alpha \frac{\partial L(\vec{w})}{\partial w_j}$$

yang dalam bentuk terperinci dapat ditulis sebagai:

$$w_j^{k+1} = w_j^k - \alpha \sum_{i=1}^r (y_i - (h(z_i)x_{i,j})), \forall j = 0, 1, \dots, n$$

3.3.3 Penerapan Metode Gauss-Newton

Metode Gauss-Newton digunakan meminimalkan fungsi *loss* $L(w)$ untuk mempercepat konvergensi pembaruan bobot dengan pendekatan kuadrat terkecil (*least squares*). Proses iterasinya dilakukan berdasarkan solusi sistem persamaan linear hingga kriteria konvergensi terpenuhi. Langkah-langkah penerapannya adalah sebagai berikut:

1. Menetapkan iterasi awal $k = 1$.
2. Menghitung ∇L_i

Nilai gradien ∇L_i diperoleh melalui turunan fungsi *loss* terhadap bobot, yang dirumuskan dalam persamaan berikut:

$$\frac{\partial L_i}{\partial w_j} = (y_i - h(z_i))x_{i,j}$$

3. Membangun matriks A dan vektor B pada sistem persamaan linear dalam persamaan (2.18).
4. Menyusun sistem persamaan linier berdasarkan hasil perhitungan, yaitu:

$$A^T A W = A^T B$$

3.3.4 Penerapan Metode Iterasi Jacobi

Untuk menyelesaikan sistem persamaan linear pada sub-bab sebelumnya, metode iterasi Jacobi digunakan dengan langkah-langkah sebagai berikut:

1. Mendefinisikan $H = A^T A$ dan $C = A^T B$, sehingga sistem persamaan yang harus diselesaikan adalah

$$HW = C$$

2. Menguraikan matriks H menjadi:

$$H = D + L + U$$

di mana:

D adalah matriks diagonal dari H , L adalah matriks segitiga bawah (tanpa elemen diagonal), U adalah matriks segitiga atas (tanpa elemen diagonal).

3. Substitusikan ke dalam sistem persamaan menghasilkan bentuk iteratif:

$$DW = C - (L + U)W$$

sehingga iterasi Jacobi diperbarui menggunakan rumus:

$$W^{k+1} = D^{-1}(C - (L + U)W^k)$$

4. Melanjutkan iterasi hingga nilai akurasi validasi (validation accuracy) mencapai batas ketelitian yang telah ditentukan, yaitu 0,75 (75%).

3.3.5 Evaluasi Model

Untuk menilai kinerja model regresi logistik biner yang telah diterapkan, digunakan beberapa metrik evaluasi berikut:

1. Akurasi (Accuracy)

Akurasi mengukur proporsi prediksi yang benar terhadap total jumlah sampel, dirumuskan sebagai:

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN}$$

2. Presisi (Precision)

Presisi mengukur seberapa andal model dalam memprediksi kelas positif, dihitung dengan:

$$Presisi = \frac{TP}{TP + FP}$$

3. Recall (Sensitivitas)

Recall mengukur seberapa banyak kasus positif yang benar-benar terdeteksi oleh model, dirumuskan sebagai:

$$Recall = \frac{TP}{TP + FN}$$

4. F1 Score

F1 Score adalah rata-rata harmonik antara presisi dan recall, digunakan untuk menyeimbangkan trade-off antara keduanya:

$$F1\ Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall}$$

Pada metrik-metrik di atas, berikut adalah definisi dari istilah yang digunakan:

TP (True Positive): Kasus dimana model memprediksi positif dan kenyataannya juga positif,

TN (True Negative): Kasus dimana model memprediksi negatif dan kenyataannya juga negatif.

FP (False Positive): Kasus dimana model memprediksi positif namun kenyataannya negatif.

FN (False Negative): Kasus dimana model memprediksi negatif namun kenyataannya positif.

3.3.6 Analisis Hasil

Setelah model dievaluasi menggunakan metrik-metrik di atas, dilakukan analisis hasil untuk memahami efektivitas pendekatan yang diterapkan. Analisis ini mencakup beberapa aspek berikut:

1. Konvergensi Metode

Evaluasi dilakukan terhadap kecepatan dan stabilitas konvergensi metode Gauss-Newton yang dikombinasikan dengan iterasi Jacobi. Hasilnya dibandingkan dengan metode lain, seperti gradient descent, untuk menilai keunggulan dan keterbatasannya.

2. Kinerja Klasifikasi

Model diuji pada dataset *Pima Indian Diabetes* untuk mengevaluasi akurasi serta metrik lainnya. Analisis ini membantu menentukan sejauh mana metode yang diterapkan efektif dalam menangani data klasifikasi medis yang bersifat biner.

3. Efisiensi Komputasi

Kinerja model tidak hanya dinilai berdasarkan akurasi, efisiensi algoritma juga menjadi aspek penting yang dianalisis, meliputi waktu eksekusi dan sumber daya komputasi yang digunakan selama pelatihan model.

Dalam penelitian ini, hasil yang diperoleh menggunakan metode Gauss-Newton yang dikombinasikan dengan iterasi Jacobi dibandingkan dengan metode optimisasi dasar, yaitu *Stochastic Gradient Descent with*

Momentum (SGDM) dan *Root Mean Square Propagation* (RMSprop). Selain dibandingkan metode optimisasi dasar, hasil yang diperoleh dari penelitian ini akan dibandingkan dengan hasil dari penelitian-penelitian terdahulu.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Deskripsi Data

Dataset yang digunakan dalam penelitian ini adalah *Pima Indian Diabetes Dataset* yang diperoleh dari Kaggle. Dataset ini merupakan salah satu dataset medis yang banyak digunakan dalam penelitian klasifikasi berbasis *machine learning*. Dataset ini berisi 768 sampel data pasien perempuan keturunan Pima Indian yang berusia di atas 21 tahun. Setiap sampel memiliki delapan variabel prediktor yang menggambarkan kondisi klinis pasien, serta satu variabel target (*Outcome*) yang menunjukkan status diabetes pasien, dengan nilai 1 untuk pasien yang terdiagnosis diabetes dan 0 untuk pasien yang tidak terdiagnosis diabetes.

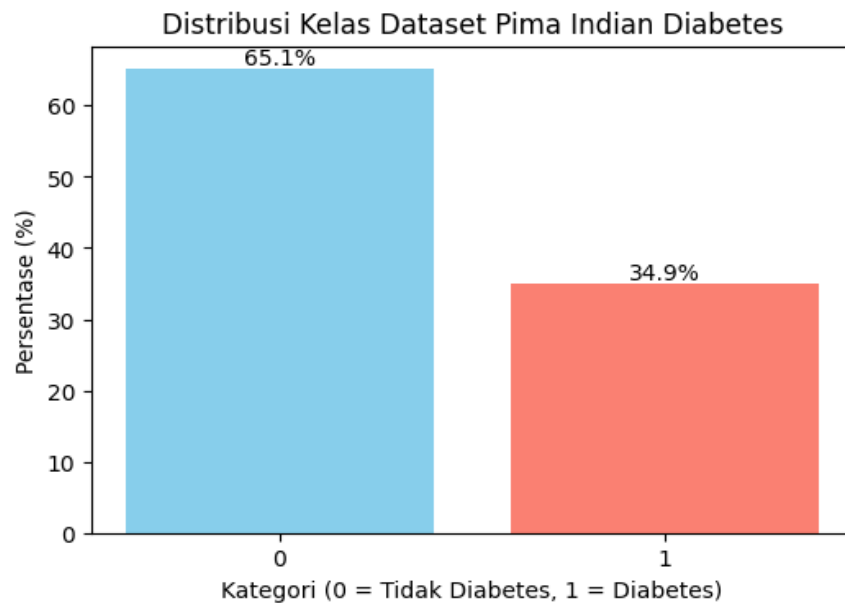
Secara umum, beberapa fitur yang terdapat dalam dataset meliputi jumlah kehamilan (*Pregnancies*), kadar glukosa plasma (*Glucose*), tekanan darah diastolik (*BloodPressure*), ketebalan lipatan kulit triceps (*SkinThickness*), kadar insulin (*Insulin*), indeks massa tubuh atau *Body Mass Index (BMI)*, fungsi pedigree diabetes (*DiabetesPedigreeFunction*), serta usia pasien (*Age*). Variabel target *Outcome* menjadi fokus utama klasifikasi, yaitu untuk memprediksi apakah pasien menderita diabetes atau tidak berdasarkan nilai-nilai fitur tersebut.

Sebagai ilustrasi, beberapa sampel data lengkap dari dataset ini ditampilkan pada Tabel 4.1 yang menunjukkan enam baris pertama dengan seluruh fitur yang ada.

Tabel 4.1 Contoh Sampel Data 6 Baris Pertama dengan Seluruh Fitur

<i>Pregnancies</i>	<i>Glucose</i>	<i>Blood Pressure</i>	<i>Skin Thickness</i>	<i>Insulin</i>	<i>BMI</i>	<i>Diabetes Pedigree Function</i>	<i>Age</i>	<i>Outcome</i>
6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0
0	137	40	35	168	43.1	2.288	33	1
5	116	74	0	0	26.6	0.201	30	0

Langkah berikutnya adalah menganalisis distribusi kelas pada variabel target (*Outcome*) untuk mengetahui proporsi antara pasien yang terdiagnosis dan tidak terdiagnosis diabetes. Visualisasi distribusi kelas tersebut disajikan pada Gambar 4.1 berikut.

Gambar 4.1 *Presentase* Tidak Diabetes dan Diabetes pada Data Pima Indian Diabetes

Berdasarkan visualisasi pada Gambar 4.1, dapat diketahui bahwa distribusi kelas pada variabel target (*Outcome*) tidak seimbang. Kategori tidak diabetes (*Outcome* = 0) mendominasi dengan proporsi sekitar 65,1%, sedangkan kategori diabetes (*Outcome* = 1) hanya berjumlah sekitar 34,9% dari keseluruhan data.

Kondisi distribusi kelas yang tidak seimbang ini perlu diperhatikan karena dapat memengaruhi performa model klasifikasi, terutama dalam mendeteksi kelas minoritas, sehingga model cenderung bias terhadap kelas mayoritas.

Untuk mempermudah ilustrasi perhitungan manual, diambil subset data berjumlah enam sampel yang terdiri dari tiga pasien tanpa diabetes (*Outcome* = 0) dan tiga pasien dengan diabetes (*Outcome* = 1). Subset ini diambil enam sampel data awal yang dapat dilihat pada Tabel 4.2. Fitur yang digunakan dalam subset ini adalah *Glucose* dan *BMI*, karena keduanya memiliki nilai korelasi tertinggi terhadap variabel target (*Outcome*) berdasarkan hasil analisis korelasi Pearson. Dengan demikian, kedua fitur tersebut dipilih sebagai representasi utama yang paling berpengaruh terhadap kemungkinan seseorang terdiagnosis diabetes.

Subset ini akan digunakan pada bagian berikutnya untuk implementasi manual metode Gauss–Newton yang dikombinasikan dengan iterasi Jacobi, sehingga setiap langkah perhitungan dapat ditelusuri secara rinci dan mudah dipahami.

Tabel 4.2 Subset 6 Sample Pertama dengan 2 Fitur Berkorelasi Tinggi

<i>i</i>	<i>X</i>		<i>Outcome (y)</i>
	<i>Glucose</i> (x_1)	<i>BMI</i> (x_2)	
1	148	33.6	1
2	183	23.3	1
3	137	43.1	1
4	85	26.6	0
5	89	28.1	0
6	78	33.7	0

4.2 Implementasi Metode secara Manual

4.2.1 Pra-pemrosesan Data Secara Manual

Pada tahap perhitungan manual, pra-pemrosesan data hanya mencakup standarisasi fitur, karena proses ini menggunakan enam sampel data sebagai

ilustrasi perhitungan matematis. Tahapan lain yaitu pembagian data tidak diterapkan, sebab jumlah data yang digunakan terlalu sedikit untuk dilakukan pemisahan maupun penyesuaian proporsi kelas. Dengan demikian, standarisasi dilakukan semata untuk menyamakan skala antar fitur agar proses optimisasi pada metode Gauss–Newton dan iterasi Jacobi berjalan stabil dan seimbang.

1. Standarisasi Fitur

Tahapan selanjutnya adalah standarisasi (*z-score*) untuk mengubah skala setiap fitur. Pemilihan standarisasi didasarkan pada kebutuhan menjaga kestabilan proses optimisasi. Karena metode Gauss-Newton dan iterasi Jacobi sangat sensitif terhadap perbedaan skala antarvariabel. Fitur dengan rentang nilai besar dapat mendominasi perhitungan pada matriks *Hessian* dan memperlambat konvergensi proses pelatihan. Dengan menerapkan standarisasi, seluruh fitur berada pada skala yang sama, sehingga setiap variabel memiliki kontribusi yang seimbang dalam proses optimisasi parameter. Rumus Standarisasi adalah sebagai berikut:

$$x'_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j}$$

Pada penelitian ini, proses standarisasi dilakukan terhadap data pada Tabel 4.2, yaitu mengambil enam data pertama $i = 1, 2, \dots, 6$ dan dua fitur utama $j = 1, 2$ yang memiliki korelasi tertinggi terhadap variable target *Outcome* yaitu *Glucose* dan *BMI*. Vektor data mentah dari kedua fitur tersebut adalah sebagai berikut:

$$\vec{x}_1 = (148; 85; 183; 89; 137; 116)$$

$$\vec{x}_2 = (33.6; 26.6; 23.3; 28.1; 43.1; 25.6)$$

Hitung rata-rata μ dan standar deviasi σ :

Menghitung rata-rata μ

$$\mu_1 = \frac{148+183+137+85+89+116+78}{6} = \frac{758}{6} = 126.33$$

$$\mu_2 = \frac{33.6+23.3+43.1+26.6+28.1+33.7}{6} = \frac{180.3}{6} = 30.05$$

Menghitung standar deviasi σ

$$\sigma = \sqrt{\frac{\sum (x_i - \mu)^2}{n}}$$

$$\sigma_1 = \sqrt{\frac{(148-126.33)^2 + (183-126.33)^2 + (137-126.33)^2 + (85-126.33)^2 + (89-126.33)^2 + (116-126.33)^2}{6}}$$

$$= \sqrt{\frac{469.5889 + 1708.1689 + 1393.5289 + 1393.5289 + 113.8489 + 106.7089}{6}}$$

$$= \sqrt{\frac{7003.3334}{6}} = \sqrt{1167.2222} = 34.16$$

$$\sigma_2 = \sqrt{\frac{(33.6-30.05)^2 + (23.3-30.05)^2 + (43.1-30.05)^2 + (26.6-30.05)^2 + (28.1-30.05)^2 + (33.7-30.05)^2}{6}}$$

$$= \sqrt{\frac{12.6025 + 11.9025 + 45.5625 + 3.8025 + 170.3025 + 19.8025}{6}}$$

$$= \sqrt{\frac{263.975}{6}} = \sqrt{43.9958} = 6.6329$$

Dengan demikian diperoleh:

$$\mu_1 = 126.33, \mu_2 = 30.05, \sigma_1 = 34.16, \sigma_2 = 6.6329$$

a. Standarisasi untuk x_{i1}

- Untuk $i = 1, x'_{11} = \frac{x_{11} - \mu_1}{\sigma_1} = \frac{148 - 126.33}{34.16} = 0.6343$

- Untuk $i = 2, x'_{21} = \frac{x_{21} - \mu_1}{\sigma_1} = \frac{85 - 126.33}{34.16} = -1.2098$

- Untuk $i = 3, x'_{31} = \frac{x_{31} - \mu_1}{\sigma_1} = \frac{183 - 126.33}{34.16} = 1.6589$

- Untuk $i = 4, x'_{41} = \frac{x_{41} - \mu_1}{\sigma_1} = \frac{89 - 126.33}{34.16} = -1.0927$

- Untuk $i = 5$, $x'_{51} = \frac{x_{51} - \mu_1}{\sigma_1} = \frac{137 - 126.33}{34.16} = 0.3123$
- Untuk $i = 6$, $x'_{61} = \frac{x_{61} - \mu_1}{\sigma_1} = \frac{78 - 126.33}{34.16} = -0.3024$

Sehingga vektor normalisasi untuk x_{i1} adalah

$$\vec{x}_1 = (0.6343; -1.2098; 1.1548; -1.6589; 0.3123; -0.3024)$$

b. Standarisasi untuk x_{i2}

- Untuk $i = 1$, $x'_{12} = \frac{x_{12} - \mu_2}{\sigma_2} = \frac{33.6 - 30.05}{6.6329} = 0.5352$
- Untuk $i = 2$, $x'_{22} = \frac{x_{22} - \mu_2}{\sigma_2} = \frac{26.6 - 30.05}{6.6329} = -0.5201$
- Untuk $i = 3$, $x'_{32} = \frac{x_{32} - \mu_2}{\sigma_2} = \frac{23.3 - 30.05}{6.6329} = -1.0176$
- Untuk $i = 4$, $x'_{42} = \frac{x_{42} - \mu_2}{\sigma_2} = \frac{28.1 - 30.05}{6.6329} = -0.2939$
- Untuk $i = 5$, $x'_{52} = \frac{x_{52} - \mu_2}{\sigma_2} = \frac{43.1 - 30.05}{6.6329} = 1.9674$
- Untuk $i = 6$, $x'_{62} = \frac{x_{62} - \mu_2}{\sigma_2} = \frac{25.6 - 30.05}{6.6329} = -0.6708$

Sehingga vektor normalisasi untuk x_{i1} adalah

$$\vec{x}_2 = (0.3431; -1.2635; 1.8251; -0.7487; 1.0451; -0.6239)$$

Hasil standarisasi ditunjukkan dalam Tabel 4.4

Tabel 4.3 Tabel Hasil Standarisasi 6 Data Awal dengan 2 Fitur

i	X		Outcome (y)
	Glucose(x_1)	BMI(x_2)	
1	0.6343	0.5352	1
2	-1.2098	-0.5201	0
3	1.6589	-1.0176	1
4	-1.0927	-0.2939	0
5	0.3123	1.9674	1
6	-0.3024	-0.6708	0

Data yang telah melalui proses standarisasi sebagaimana ditampilkan pada Tabel 4.3 selanjutnya digunakan sebagai *input* utama dalam perhitungan regresi logistik biner. Setiap sampel data ditambahkan kolom konstanta $x_0 = 1$ agar bias (w_0) dapat dihitung bersama dengan bobot lainnya. Dengan demikian, setiap baris data memiliki tiga komponen, yaitu x_0 (konstanta 1), x_1 (*Glucose*), dan x_2 (*BMI*). Dengan penambahan kolom x_0 , matriks fitur X menjadi:

$$X = \begin{bmatrix} 1 & 0.6343 & 0.3431 \\ 1 & -1.2098 & -1.2635 \\ 1 & 1.6589 & 1.8251 \\ 1 & -1.0927 & -0.7487 \\ 1 & 0.3123 & 1.0451 \\ 1 & -0.3024 & -0.6239 \end{bmatrix}$$

4.2.2 Inisialisasi Bobot

Nilai awal bobot ditetapkan secara sederhana agar perhitungan manual mudah ditelusuri. Vektor bobot awal dinyatakan sebagai berikut:

$$w_0, w_1, w_2 = (0.1, 0.1, 0.1)$$

Penggunaan tiga bobot dilakukan karena terdapat dua variabel prediktor (*Glucose* dan *BMI*) ditambah satu bobot bias w_0 yang merepresentasikan konstanta. Dengan demikian, jumlah bobot yang dihitung manual hanya mencakup w_0, w_1 , dan w_2 . Adapun pada implementasi komputer semua delapan variabel prediktor dalam data Pima Indian Diabetes tetap digunakan.

Perlu diperhatikan bahwa inisialisasi ini berbeda dengan nilai awal yang digunakan pada metode iterasi Jacobi. Pada regresi logistik, bobot awal berfungsi sebagai titik awal perhitungan fungsi logistik dan gradien. Sedangkan pada iterasi

Jacobi, nilai awal yang diberikan bukan lagi bobot model, melainkan tebakan awal solusi dari sistem persamaan linier.

4.2.3 Menghitung Fungsi Sigmoid

1. Menghitung z_i

Setelah bobot ditentukan, langkah berikutnya adalah menghitung nilai z_i , untuk setiap data ke- i . Nilai z_i diperoleh dari perkalian (*dot product*) antara vektor fitur \vec{x}_i dan vektor bobot \vec{w} , yang rumusnya ditunjukkan pada Persamaan (2.3)

a. Untuk $i = 1$

$$z_1 = w_0x_{1,0} + w_1x_{1,1} + w_2x_{1,2}$$

$$z_1 = 0.1 \times 1 + 0.1 \times 0.6343 + 0.1 \times 0.5352$$

$$z_1 = 0.1 + 0.0634 + 0.0535$$

$$z_1 = 0.2169$$

b. Untuk $i = 2$

$$z_2 = w_0x_{2,0} + w_1x_{2,1} + w_2x_{2,2}$$

$$z_2 = 0.1 \times 1 + 0.1 \times (-1.2098) + 0.1 \times (-0.5201)$$

$$z_2 = 0.1 + (-0.1209) + (-0.0520)$$

$$z_2 = -0.0729$$

c. Untuk $i = 3$

$$z_3 = w_0x_{3,0} + w_1x_{3,1} + w_2x_{3,2}$$

$$z_3 = 0.1 \times 1 + 0.1 \times 1.6589 + 0.1 \times (-1.0176)$$

$$z_3 = 0.1 + 0.1658 + (-0.1017)$$

$$z_3 = 0.1641$$

d. Untuk $i = 4$

$$z_4 = w_0x_{4,0} + w_1x_{4,1} + w_2x_{4,2}$$

$$z_4 = 0.1 \times 1 + 0.1 \times (-1.0927) + 0.1 \times (-0.2939)$$

$$z_4 = 0.1 + (-0.1092) + (-0.0293)$$

$$z_4 = (-0.0385)$$

e. Untuk $i = 5$

$$z_5 = w_0x_{5,0} + w_1x_{5,1} + w_2x_{5,2}$$

$$z_5 = 0.1 \times 1 + 0.1 \times 0.3123 + 0.1 \times 1.9674$$

$$z_5 = 0.1 + 0.0312 + 0.1967$$

$$z_5 = 0.3279$$

f. Untuk $i = 6$

$$z_6 = w_0x_{6,0} + w_1x_{6,1} + w_2x_{6,2}$$

$$z_6 = 0.1 \times 1 + 0.1 \times (-0.3024) + 0.1 \times (-0.6708)$$

$$z_6 = 0.1 + (-0.0302) + (-0.0670)$$

$$z_6 = 0.0028$$

2. Menghitung $h(z_i)$

Setelah nilai z_i diperoleh, langkah selanjutnya adalah menghitung fungsi sigmoid untuk masing-masing sampel. Fungsi sigmoid didefinisikan pada Persamaan 2.2. Fungsi ini akan mengubah nilai z_i menjadi probabilitas di rentang $(0,1)$, yang menunjukkan kemungkinan data ke- i termasuk ke dalam kelas positif ($y = 1$). Hasil perhitungan $h(z_i)$ untuk setiap data adalah:

a. Untuk $i = 1$

$$h(z_1) = \frac{1}{1 + \exp(-z_1)} = \frac{1}{1 + \exp(-0.2169)} = 0.5540$$

b. Untuk $i = 2$

$$h(z_2) = \frac{1}{1 + \exp(-z_2)} = \frac{1}{1 + \exp(-(-0.0729))} = 0.4817$$

c. Untuk $i = 3$

$$h(z_3) = \frac{1}{1 + \exp(-z_3)} = \frac{1}{1 + \exp(-0.1641)} = 0.5409$$

d. Untuk $i = 4$

$$h(z_4) = \frac{1}{1 + \exp(-z_4)} = \frac{1}{1 + \exp(-(-0.0385))} = 0.4903$$

e. Untuk $i = 5$

$$h(z_5) = \frac{1}{1 + \exp(-z_5)} = \frac{1}{1 + \exp(-0.3279)} = 0.5812$$

f. Untuk $i = 6$

$$h(z_6) = \frac{1}{1 + \exp(-z_6)} = \frac{1}{1 + \exp(-0.0028)} = 0.5007$$

4.2.4 Menghitung Fungsi Loss

Untuk mengukur performa prediksi model terhadap nilai sebenarnya digunakan fungsi *binary cross-entropy* pada Persamaan 2.4. Nilai *loss* untuk setiap sampel dihitung menggunakan Persamaan 2.12.

1. Untuk $i = 1$

$$l_1 = -(y_1 \log(h(z_1)) + (1 - y_1) \log(1 - h(z_1)))$$

$$l_1 = -(1 \log(0.5540) + (1 - 1) \log(1 - 0.5540))$$

$$l_1 = -(1 \times (-0.2564) + 0 \times (-0.3506))$$

$$l_1 = -(-0.2564) = 0.2564$$

2. Untuk $i = 2$

$$l_2 = -(y_2 \log(h(z_2)) + (1 - y_2) \log(1 - h(z_2)))$$

$$l_2 = -(0 \log(-0.4817) + (1 - 0) \log(1 - (-0.4817)))$$

$$l = -(0 \times (-0.3171) + 1 \times (-0.2854))$$

$$l_2 = -(-0.2854) = 0.2854$$

3. Untuk $i = 3$

$$l_3 = -(y_3 \log(h(z_3)) + (1 - y_3) \log(1 - h(z_3)))$$

$$l_3 = -(1 \log(0.5409) + (1 - 1) \log(1 - 0.5409))$$

$$l_3 = -(1 \times (-0.2668) + 0 \times (-0.3381))$$

$$l_3 = -(-0.2668) = 0.2668$$

4. Untuk $i = 4$

$$l_4 = -(y_4 \log(h(z_4)) + (1 - y_4) \log(1 - h(z_4)))$$

$$l_4 = -(0 \log(0.4903) + (1 - 0) \log(1 - 0.4903))$$

$$l_4 = -(0 \times (-0.3094) + 1 \times (-0.2927))$$

$$l_4 = -(-0.2927) = 0.2927$$

5. Untuk $i = 5$

$$l_5 = -(y_5 \log(h(z_5)) + (1 - y_5) \log(1 - h(z_5)))$$

$$l_5 = -(1 \log(0.5812) + (1 - 1) \log(1 - 0.5812))$$

$$l_5 = -(1 \times (-0.2356) + 0 \times (-0.3780))$$

$$l_5 = -(-0.2356) = 0.2356$$

6. Untuk $i = 6$

$$l_6 = -(y_6 \log(h(z_6)) + (1 - y_6) \log(1 - h(z_6)))$$

$$l_6 = -(0 \log(0.5007) + (1 - 0) \log(1 - 0.5007))$$

$$l_6 = -(0 \times (-0.3004) + 1 \times (-0.3016))$$

$$l_6 = -(-0.3016) = 0.3016$$

Dan diperoleh total loss

$$L(\vec{w}) = -\frac{1}{r} \sum_{i=1}^r [y_i \log(h(z_i)) + (1 - y_i) \log(1 - h(z_i))]$$

$$L(\vec{w}) = \frac{1}{6} (0.2564 + 0.2854 + 0.2668 + 0.2927 + 0.2356 + 0.3016)$$

$$L(\vec{w}) = \frac{1}{6} (1.6385) = 0.2730$$

4.2.5 Formulasi Gauss-Newton

Pada tahap formulasi Gauss-Newton, dilakukan proses penurunan fungsi *loss* terhadap setiap bobot untuk mendapatkan gradien ∇l_i . Gradien ini menunjukkan arah perubahan *loss* terhadap setiap parameter model pada sampel ke-*i*. Nilai gradien tiap sampel selanjutnya menjadi dasar dalam pembentukan matriks Jacobian *A* yang digunakan dalam penyusunan sistem persamaan linier. Secara matematis, gradien untuk bobot w_j yang terdapat pada Persamaan

$$\frac{\partial L_i}{\partial w_j} = (y_i - h(z_i))x_{i,j}, \quad i = 1, 2, \dots, 6 \text{ dan } j = 0, 1, 2, \dots, n.$$

dengan $x_{i,0} = 1$ sebagai konstanta untuk bias

1. Menghitung $\nabla L_1 = \left(\frac{\partial L_1}{\partial w_0}, \frac{\partial L_1}{\partial w_1}, \frac{\partial L_1}{\partial w_2} \right)$,

- a. Untuk $i = 1, j = 0$

$$\frac{\partial L_1}{\partial w_0} = (y_1 - h(z_1))x_{1,0} = (1 - 0.5540) \times 1 = 0.4459$$

- b. Untuk $i = 1, j = 1$

$$\frac{\partial L_1}{\partial w_1} = (y_1 - h(z_1))x_{1,1} = (1 - 0.5540) \times 0.6343 = 0.2828$$

- c. Untuk $i = 1, j = 2$

$$\frac{\partial L_1}{\partial w_2} = (y_1 - h(z_1))x_{1,2} = (1 - 0.5540) \times 0.5352 = 0.2386$$

Hasil perhitungan nilai ∇L_1 yaitu sebagai berikut.

$$\nabla L_1 = (0.4459; 0.2828; 0.2386)$$

2. Menghitung $\nabla L_2 = \left(\frac{\partial L_2}{\partial w_0}, \frac{\partial L_2}{\partial w_1}, \frac{\partial L_2}{\partial w_2} \right)$

- a. Untuk $i = 2, j = 0$

$$\frac{\partial L_2}{\partial w_0} = (y_2 - h(z_2))x_{2,0} = (0 - 0.4817) \times 1 = -0.4817$$

- b. Untuk $i = 2, j = 1$

$$\frac{\partial L_2}{\partial w_1} = (y_2 - h(z_2))x_{2,1} = (0 - 0.4817) \times (-1.2098) = 0.5828$$

- c. Untuk $i = 2, j = 2$

$$\frac{\partial L_2}{\partial w_2} = (y_2 - h(z_2))x_{2,2} = (0 - 0.4817) \times (-0.5201) = 0.2505$$

Hasil perhitungan nilai ∇L_2 yaitu sebagai berikut

$$\nabla L_2 = (-0.4817; 0.5828; 0.2505)$$

3. Menghitung $\nabla L_3 = \left(\frac{\partial L_3}{\partial w_0}, \frac{\partial L_3}{\partial w_1}, \frac{\partial L_3}{\partial w_2} \right)$

- a. Untuk $i = 3, j = 0$

$$\frac{\partial L_3}{\partial w_0} = (y_3 - h(z_3))x_{3,0} = (1 - 0.5409) \times 1 = 0.4590$$

b. Untuk $i = 3, j = 1$

$$\frac{\partial L_3}{\partial w_1} = (y_3 - h(z_3))x_{31} = (1 - 0.5409) \times 1.6589 = 0.7615$$

c. Untuk $i = 3, j = 2$

$$\frac{\partial L_3}{\partial w_2} = (y_3 - h(z_3))x_{32} = (1 - 0.5409) \times (-1.0176) = -0.4671$$

Hasil perhitungan nilai ∇L_3 yaitu sebagai berikut

$$\nabla L_3 = (0.4590; 0.7615; -0.4671)$$

Perhitungan nilai ∇L_i untuk $i = 4, 5$, dan 6 dilakukan menggunakan prosedur yang sama seperti pada perhitungan sebelumnya, yaitu sebagai berikut.

Hasil perhitungan nilai ∇L_i untuk 6 data yaitu sebagai berikut.

$$\nabla L_1 = (0.4459; 0.2828; 0.2386)$$

$$\nabla L_2 = (-0.4817; 0.5828; 0.2505)$$

$$\nabla L_3 = (0.4590; 0.7615; -0.4671)$$

$$\nabla L_4 = (-0.4903; 0.5358; 0.1441)$$

$$\nabla L_5 = (0.4187; 0.5358; 0.8238)$$

$$\nabla L_6 = (-0.5007; 0.1514; 0.3358)$$

4.2.6 Menyusun matriks Jacobian A dan matriks RHS

Setelah nilai gradien terhadap setiap bobot diperoleh, langkah selanjutnya adalah menyusunnya ke dalam bentuk matriks Jacobian A. Masing-masing baris pada matriks ini merepresentasikan turunan fungsi *loss* terhadap semua parameter (w_0, w_1, w_2) untuk satu sampel data, sedangkan kolom merepresentasikan gradien terhadap satu bobot untuk keseluruhan. Secara keseluruhan, bentuk matriks Jacobian A adalah:

$$A = \begin{bmatrix} 0.4459 & 0.2828 & 0.2386 \\ -0.4817 & 0.5828 & 0.2505 \\ 0.4590 & 0.7615 & -0.4671 \\ -0.4903 & 0.5358 & 0.1441 \\ 0.4817 & 0.1397 & 0.8238 \\ -0.5007 & 0.1514 & 0.3358 \end{bmatrix}$$

Selain matriks A, dibentuk pula matriks RHS (*Right Hand Side*) yang diperoleh dari hasil *loss* dari setiap sampel yang dinotasikan dengan B:

$$B = \begin{bmatrix} 0.2565 \\ 0.2855 \\ 0.2669 \\ 0.2927 \\ 0.2356 \\ 0.3016 \end{bmatrix}$$

Langkah berikutnya adalah menghitung hasil perkalian transpose dari Jacobian dengan dirinya sendiri, yaitu:

$$A^T A = \begin{bmatrix} 1.3082 & -0.0888 & -0.1225 \\ -0.0888 & 1.3268 & 0.0936 \\ -0.1225 & 0.0936 & 1.1502 \end{bmatrix}$$

Kemudian dilakukan perkalian antara transpose Jacobian dan matriks RHS untuk memperoleh ruas kanan dari sistem persamaan linier:

$$A^T B = \begin{bmatrix} 0.0965 \\ -0.6754 \\ -0.3457 \end{bmatrix}$$

Dengan demikian, sistem persamaan linier metode Gauss–Newton dapat ditulis pada Persamaan 2.17. Sistem persamaan linier ini selanjutnya akan diselesaikan menggunakan metode iterasi Jacobi untuk memperoleh nilai pembaruan bobot. Dengan pendekatan iteratif ini, proses optimisasi tidak perlu menghitung invers matriks secara langsung, sehingga lebih efisien secara komputasi.

4.2.7 Penyelesaian SPL menggunakan iterasi Jacobi

Setelah diperoleh hasil perhitungan $A^T A$ dan $A^T B$, tahap selanjutnya adalah menyusunnya ke dalam bentuk sistem persamaan linier pada metode Gauss–Newton. Dalam hal ini ditetapkan bahwa $H = A^T A$ dan $C = A^T B$, sehingga sistem dapat dituliskan sebagai:

$$HW = C$$

Setelah persamaan $HW = C$ terbentuk, selanjutnya melakukan dekomposisi matriks H ke dalam tiga komponen, yaitu D (diagonal utama), L (elemen segitiga bawah), dan U (elemen segitiga atas). Dengan demikian dapat dituliskan bahwa:

$$H = D + L + U.$$

Bentuk dekomposisi ini penting karena menjadi dasar dalam penerapan metode iterasi Jacobi. Selanjutnya, proses pembaruan bobot dilakukan menggunakan formula:

$$W^{k+1} = D^{-1}(C - (L + U)W^k)$$

1. Inisialisasi bobot

Pada tahap sebelumnya, inisialisasi bobot pada regresi logistik dilakukan untuk memberikan nilai awal bagi parameter model yang akan dipelajari dari data. Bobot tersebut ditetapkan secara sederhana, misalnya dengan nilai yang sama pada setiap parameter, agar proses perhitungan manual lebih mudah ditelusuri.

Sementara itu, pada metode iterasi Jacobi, peran inisialisasi bobot berbeda. Nilai yang ditetapkan bukan lagi parameter regresi logistik, melainkan tebakan awal solusi sistem persamaan linier. Dengan demikian, inisialisasi bobot awal ditetapkan sebagai:

$$W^0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, k = 0$$

2. Dekomposisi matriks H

Matriks H didekomposisi menjadi tiga bagian, yaitu matriks diagonal D , matriks segitiga bawah L , dan matriks segitiga atas U , sehingga dapat dituliskan pada Persamaan 2.21 dengan:

$$H = \begin{bmatrix} 1.3082 & -0.0888 & -0.1225 \\ -0.0888 & 1.3268 & 0.0936 \\ -0.1225 & 0.0936 & 1.1502 \end{bmatrix}$$

$$D = \begin{bmatrix} 1.3082 & 0 & 0 \\ 0 & 1.3268 & 0 \\ 0 & 0 & 1.1502 \end{bmatrix}$$

$$L = \begin{bmatrix} 0 & 0 & 0 \\ -0.0888 & 0 & 0 \\ -0.1225 & 0.0936 & 0 \end{bmatrix}$$

$$U = \begin{bmatrix} 0 & -0.0888 & -0.1225 \\ 0 & 0 & 0.0936 \\ 0 & 0 & 0 \end{bmatrix}$$

3. Hitung invers matriks diagonal

Untuk keperluan perhitungan iterasi Jacobi, diperlukan invers dari diagonal utama:

$$D = \begin{bmatrix} 1.3082 & 0 & 0 \\ 0 & 1.3268 & 0 \\ 0 & 0 & 1.1502 \end{bmatrix}$$

$$D^{-1} = \begin{bmatrix} 0.7643 & 0 & 0 \\ 0 & 0.7536 & 0 \\ 0 & 0 & 0.8693 \end{bmatrix}$$

4. Hitung matriks $L + U$

Gabungan elemen non-diagonal:

$$(L + U) = \left[\begin{pmatrix} 0 & 0 & 0 \\ -0.0888 & 0 & 0 \\ -0.1225 & 0.0936 & 0 \end{pmatrix} + \begin{pmatrix} 0 & -0.0888 & -0.1225 \\ 0 & 0 & 0.0936 \\ 0 & 0 & 0 \end{pmatrix} \right]$$

$$(L + U) = \begin{bmatrix} 0 & -0.0888 & -0.1225 \\ -0.0888 & 0 & 0.0936 \\ -0.1225 & 0.0936 & 0 \end{bmatrix}$$

5. Formula iterasi Jacobi

Proses iterasi dilakukan menggunakan Persamaan 2.24:

$$W^{k+1} = D^{-1}(C - (L + U)W^k)$$

Iterasi 1

Untuk k=0

$$W^{0+1} = D^{-1}(C - (L + U)W^0)$$

$$W^1 = \begin{bmatrix} 0.7643 & 0 & 0 \\ 0 & 0.7536 & 0 \\ 0 & 0 & 0.8693 \end{bmatrix}^{-1} \left(\begin{bmatrix} 0.0965 \\ -0.6754 \\ -0.3457 \end{bmatrix} - \left(\begin{bmatrix} 0 & 0 & 0 \\ -0.0888 & 0 & 0 \\ -0.1225 & 0.0936 & 0 \end{bmatrix} + \begin{bmatrix} 0 & -0.0888 & -0.1225 \\ 0 & 0 & 0.0936 \\ 0 & 0 & 0 \end{bmatrix} \right) \times \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right)$$

$$W^1 = \begin{bmatrix} 0.7643 & 0 & 0 \\ 0 & 0.7536 & 0 \\ 0 & 0 & 0.8693 \end{bmatrix} \times \left(\begin{bmatrix} 0.0965 \\ -0.6754 \\ -0.3457 \end{bmatrix} - \begin{bmatrix} 0 & -0.0888 & -0.1225 \\ -0.0888 & 0 & 0.0936 \\ -0.1225 & 0.0936 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right)$$

$$W^1 = \begin{bmatrix} 0.0737 \\ -0.5090 \\ -0.3005 \end{bmatrix}$$

Sehingga $W^1 = (0.0737, -0.5090, -0.3005)$

Iterasi 2

k=1

$$W^{1+1} = D^{-1}(C - (L + U)W^1)$$

$$W^2 = \begin{bmatrix} 0.7643 & 0 & 0 \\ 0 & 0.7536 & 0 \\ 0 & 0 & 0.8693 \end{bmatrix}^{-1} \left(\begin{bmatrix} 0.0965 \\ -0.6754 \\ -0.3457 \end{bmatrix} - \left(\begin{bmatrix} 0 & 0 & 0 \\ -0.0888 & 0 & 0 \\ -0.1225 & 0.0936 & 0 \end{bmatrix} + \begin{bmatrix} 0 & -0.0888 & -0.1225 \\ 0 & 0 & 0.0936 \\ 0 & 0 & 0 \end{bmatrix} \right) \times \begin{bmatrix} 0.0737 \\ -0.5090 \\ -0.3005 \end{bmatrix} \right)$$

$$W^2 = \begin{bmatrix} 0.7643 & 0 & 0 \\ 0 & 0.7536 & 0 \\ 0 & 0 & 0.8693 \end{bmatrix} \times \left(\begin{bmatrix} 0.0965 \\ -0.6754 \\ -0.3457 \end{bmatrix} - \begin{bmatrix} 0 & -0.0888 & -0.1225 \\ -0.0888 & 0 & 0.0936 \\ -0.1225 & 0.0936 & 0 \end{bmatrix} \times \begin{bmatrix} 0.0737 \\ -0.5090 \\ -0.3005 \end{bmatrix} \right)$$

$$W^2 = \begin{bmatrix} 0.0110 \\ -0.4829 \\ -0.2512 \end{bmatrix}$$

Sehingga $w^2 = (0.0110, -0.4829, -0.2512)$

Iterasi 3

k=2

$$W^{2+1} = D^{-1}(C - (L + U)W^2)$$

$$W^3 = \begin{bmatrix} 0.7643 & 0 & 0 \\ 0 & 0.7536 & 0 \\ 0 & 0 & 0.8693 \end{bmatrix}^{-1} \left(\begin{bmatrix} 0.0965 \\ -0.6754 \\ -0.3457 \end{bmatrix} - \left(\begin{bmatrix} 0 & 0 & 0 \\ -0.0888 & 0 & 0 \\ -0.1225 & 0.0936 & 0 \end{bmatrix} + \begin{bmatrix} 0 & -0.0888 & -0.1225 \\ 0 & 0 & 0.0936 \\ 0 & 0 & 0 \end{bmatrix} \right) \times \begin{bmatrix} 0.0110 \\ -0.4829 \\ -0.2512 \end{bmatrix} \right)$$

$$W^3 = \begin{bmatrix} 0.7643 & 0 & 0 \\ 0 & 0.7536 & 0 \\ 0 & 0 & 0.8693 \end{bmatrix} \times \left(\begin{bmatrix} 0.0965 \\ -0.6754 \\ -0.3457 \end{bmatrix} - \begin{bmatrix} 0 & -0.0888 & -0.1225 \\ -0.0888 & 0 & 0.0936 \\ -0.1225 & 0.0936 & 0 \end{bmatrix} \times \begin{bmatrix} 0.0110 \\ -0.4829 \\ -0.2512 \end{bmatrix} \right)$$

$$W^3 = \begin{bmatrix} 0.0174 \\ -0.4906 \\ -0.26 \end{bmatrix}$$

Sehingga $w^3 = (0.0174, -0.4906, -0.26)$

Tahapan ini menunjukkan proses pembaruan bobot secara iterative dengan metode Jacobi. Dalam penelitian ini, iterasi manual hanya dilakukan hingga iterasi ke-3 untuk keperluan penjelasan matematis dan pelacakan langkah perhitungan. Batas tiga iterasi dipilih agar proses dapat ditelusuri tanpa menampilkan keseluruhan komputasi program. Dalam implementasi sebenarnya, proses iterasi dilanjutkan secara otomatis hingga memenuhi kriteria berhenti, yaitu *validation accuracy* > 0.75 . Dan Nilai W^3 menunjukkan hasil sementara bobot yang akan digunakan untuk menghitung nilai prediksi \hat{y} pada subbab berikutnya.

4.2.8 Perhitungan Prediksi

Setelah diperoleh vektor bobot hasil iterasi Jacobi pada tahap sebelumnya, yaitu $W^3 = [0.0174, -0.4906, -0.26]$, langkah berikutnya adalah menghitung nilai prediksi \hat{y} berdasarkan data yang telah distandarisasi. Pada regresi logistik, proses prediksi dilakukan dengan menghitung fungsi aktivasi sigmoid $h(z_i)$, yang mengubah hasil kombinasi linear z_i menjadi nilai probabilitas prediksi (\hat{y}) antara 0

dan 1. Secara umum, hubungan antara vektor bobot dan data hasil standarisasi dapat ditulis sebagai:

$$z_i = w_0x_{i,0} + w_1x_{i,1} + w_2x_{i,2}$$

Dengan $x_{i,0} = 1$ sebagai bias. Nilai z_i kemudian digunakan dalam fungsi sigmoid untuk menghasilkan probabilitas hasil prediksi sebagai berikut:

$$h(z_i) = \frac{1}{1 + \exp(-z_i)}$$

Nilai probabilitas $h(z_i)$ menunjukkan seberapa besar kemungkinan suatu data termasuk ke dalam kelas positif (label 1). Untuk mengubah nilai probabilitas tersebut menjadi kelas prediksi biner, digunakan ambang batas (threshold) sebesar 0,5. Apabila $h(z_i) \geq 0.5$ maka sampel diklasifikasikan sebagai kelas 1 (positif), sedangkan jika $h(z_i) < 0.5$, maka diklasifikasikan sebagai kelas 0 (negatif). Aturan keputusan ini dapat dinyatakan sebagai:

$$\hat{y} = \begin{cases} 1, & h(z_i) \geq 0.5 \\ 0, & h(z_i) < 0.5 \end{cases}$$

1. Perhitungan nilai Probabilitas

Nilai z_i dihitung menggunakan Persamaan 2.3:

a. Untuk $i = 1$

$$z_1 = w_0x_{1,0} + w_1x_{1,1} + w_2x_{1,2}$$

$$z_1 = 0.0174 \times 1 + (-0.4906) \times 0.6343 + (-0.2600) \times 0.5352$$

$$z_1 = 0.0174 + (-0.3122) + (-0.1391)$$

$$z_1 = (-0.4339)$$

b. Untuk $i = 2$

$$z_2 = w_0x_{2,0} + w_1x_{2,1} + w_2x_{2,2}$$

$$z_2 = 0.0174 \times 1 + (-0.4906) \times (-1.2098) + (-0.2600) \times (-0.5201)$$

$$z_2 = 0.0174 + 0.5935 + 0.1352$$

$$z_2 = 0.7113$$

c. Untuk $i = 3$

$$z_3 = w_0x_{3,0} + w_1x_{3,1} + w_2x_{3,2}$$

$$z_3 = 0.0174 \times 1 + (-0.4906) \times 1.6589 + (-0.2600) \times (-1.0176)$$

$$z_3 = 0.0174 + (-0.3138) + 0.2645$$

$$z_3 = -0.0319$$

d. Untuk $i = 4$

$$z_4 = w_0x_{4,0} + w_1x_{4,1} + w_2x_{4,2}$$

$$z_4 = 0.0174 \times 1 + (-0.4906) \times (-1.0927)$$

$$+ (-0.2600) \times (-0.2939)$$

$$z_4 = 0.0174 + 0.5360 + 0.0764$$

$$z_4 = 0.6298$$

e. Untuk $i = 5$

$$z_5 = w_0x_{5,0} + w_1x_{5,1} + w_2x_{5,2}$$

$$z_5 = 0.0174 \times 1 + (-0.4906) \times 0.3123 + (-0.2600) \times 1.9674$$

$$z_5 = 0.0174 + (-0.1532) + (-0.5115)$$

$$z_5 = (-0.6473)$$

f. Untuk $i = 6$

$$z_6 = w_0x_{6,0} + w_1x_{6,1} + w_2x_{6,2}$$

$$z_6 = 0.0174 \times 1 + (-0.4906) \times (-0.3024) + (-0.2600) \times (-0.6708)$$

$$z_6 = 0.0174 + 0.1483 + (-0.1744)$$

$$z_6 = (-0.0087)$$

2. Menghitung fungsi probabilitas (sigmoid)

Nilai probabilitas diperoleh dengan fungsi sigmoid

$$h(z_i) = \frac{1}{1 + \exp(-z_i)}$$

a. Untuk $i = 1$

$$h(z_1) = \frac{1}{1 + \exp(-z_1)} = \frac{1}{1 + \exp(-(-0.4339))} = 0.3932$$

b. Untuk $i = 2$

$$h(z_2) = \frac{1}{1 + \exp(-z_2)} = \frac{1}{1 + \exp(-0.7113)} = 0.6706$$

c. Untuk $i = 3$

$$h(z_3) = \frac{1}{1 + \exp(-z_3)} = \frac{1}{1 + \exp(-0.1641)} = 0.5079$$

d. Untuk $i = 4$

$$h(z_4) = \frac{1}{1 + \exp(-z_4)} = \frac{1}{1 + \exp(-0.6298)} = 0.6524$$

e. Untuk $i = 5$

$$h(z_5) = \frac{1}{1 + \exp(-z_5)} = \frac{1}{1 + \exp(-(-0.6473))} = 0.3435$$

f. Untuk $i = 6$

$$h(z_6) = \frac{1}{1 + \exp(-z_6)} = \frac{1}{1 + \exp(-(-0.0087))} = 0.4978$$

3. Penentuan Kelas Prediksi

Kelas hasil prediksi ditentukan menggunakan ambang batas 0.5 dengan aturan keputusan sebagai berikut:

$$\hat{y} = \begin{cases} 1, h(z_i) \geq 0.5 \\ 0, h(z_i) < 0.5 \end{cases}$$

Tabel berikut menyajikan hasil perhitungan untuk seluruh sampel data standarisasi:

Tabel 4.4 Ringkasan Hasil Prediksi

i	x_1 (<i>Glucose</i>)	x_2 (<i>BMI</i>)	$h(z_i)$	y	\hat{y}	Kategori
1	0.6343	0.5352	0.3932	1	0	FP
2	-1.2098	-0.5201	0.6706	0	1	FN
3	1.6589	-1.0176	0.5079	1	1	TP
4	-1.0927	-0.2939	0.6524	0	1	FN
5	0.3123	1.9674	0.3435	1	0	FP
6	-0.3024	-0.6708	0.4978	0	0	TN

Hasil prediksi pada Tabel 4.4 kemudian dievaluasi untuk mengetahui performa model serta memastikan apakah proses iterasi telah mencapai konvergensi. Tahap evaluasi hasil tersebut dibahas pada subbab berikutnya

4.2.9 Evaluasi Model

Subbab ini membahas evaluasi hasil prediksi model menggunakan bobot W^3 untuk mengetahui tingkat akurasi dan konvergensi proses pembelajaran. Berdasarkan hasil perhitungan pada Tabel 4.4 diperoleh:

$$TP = 1, TN = 1, FP = 2, FN = 2$$

Sehingga metrik evaluasi dihitung menggunakan rumus:

$$\begin{aligned}
 \text{Akurasi} &= \frac{TP + TN}{TP + TN + FP + FN} \\
 &= \frac{1 + 1}{1 + 1 + 2 + 2} = \frac{2}{6} = 0.34
 \end{aligned}$$

$$\begin{aligned}
 \textit{Presisi} &= \frac{TP}{TP + FP} \\
 &= \frac{1}{1 + 2} = \frac{1}{3} = 0.34
 \end{aligned}$$

$$\begin{aligned}
 \textit{Recall} &= \frac{TP}{TP + FN} \\
 &= \frac{1}{1 + 2} = \frac{1}{3} = 0.34
 \end{aligned}$$

$$\begin{aligned}
 \textit{F1 Score} &= 2 \times \frac{\textit{Presisi} \times \textit{Recall}}{\textit{Presisi} + \textit{Recall}} \\
 &= 2 \times \frac{0.34 \times 0.34}{0.34 + 0.34} = 2 \times \frac{0.1156}{0.68} = 0.3447
 \end{aligned}$$

Untuk mencapai hasil yang lebih baik, proses iterasi dilanjutkan hingga memenuhi kriteria berhenti, yaitu validation accuracy > 0.75. Kriteria tersebut digunakan untuk menentukan batas evaluasi pada data validasi, sehingga proses iterasi tidak dihentikan sebelum model mencapai tingkat akurasi validasi yang memadai.

Dengan melanjutkan iterasi hingga memenuhi kriteria validation accuracy tersebut, diharapkan bobot yang diperoleh menjadi semakin stabil dan model mampu memberikan performa yang lebih baik pada data uji. Tahap evaluasi ini menunjukkan bahwa hasil sementara dari iterasi Jacobi telah menghasilkan arah pembaruan bobot yang konsisten menuju kondisi konvergen, sehingga dapat dijadikan dasar untuk analisis performa model secara keseluruhan pada subbab berikutnya.

4.3 Hasil Implementasi Secara Komputasional

4.3.1 Pra-Pemrosesan Data Secara Komputasional

Setelah sebelumnya dilakukan perhitungan manual menggunakan enam data awal untuk memberikan ilustrasi langkah perhitungan pada gabungan metode Gauss–Newton dan iterasi Jacobi, tahap ini membahas pra-pemrosesan data secara komputasional. Tujuan utama pada tahap ini adalah menyiapkan dataset agar dapat digunakan dalam pelatihan model regresi logistik biner dengan metode Gauss–Newton dan iterasi Jacobi.

Pra-pemrosesan dilakukan secara komputasional dilakukan menggunakan bahasa pemrograman Python dan meliputi dua tahap utama yaitu: pembagian data dan standarisasi data. Berbeda dengan pra-pemrosesan data secara manual yang hanya menggunakan sebagian kecil data, tahap ini menggunakan seluruh dataset Pima Indian Diabetes yang terdiri atas 768 sampel dengan delapan fitur prediktor dan satu variabel target (*Outcome*).

1. Pembagian Data

Langkah pertama dalam tahap pra-pemrosesan adalah melakukan pembagian dataset menjadi tiga bagian utama, yaitu data latih, data validasi, dan data uji. Pembagian ini dilakukan dengan proporsi 70% : 15% : 15% dari total 768 sampel pada *Pima Indian Diabetes dataset*. Perhitungan pembagian dataset tersebut dirangkum pada Tabel 4.5.

Tabel 4.5 Perhitungan Proporsi Pembagian Dataset

No	Jenis Data	Proporsi	Perhitungan	Jumlah sampel (dibulatkan)
1	Data Latih	70%	0.70×768 $= 537.6$	538
2	Data Validasi	15%	0.15×768 $= 115.2$	115
3	Data Uji	15%	0.15×768 $= 115.2$	115
Total		100%		768

Secara rinci, tujuan pembagian data dilakukan sebagai berikut:

- a. Data latih (70%) digunakan untuk melatih model, yaitu menyesuaikan parameter-parameter model regresi logistik agar mampu mengenali pola hubungan antara fitur dan variabel target.
- b. Data validasi (15%) digunakan untuk mengevaluasi performa model selama proses pelatihan, membantu menentukan parameter optimal (misalnya jumlah iterasi atau tingkat toleransi konvergensi), serta mencegah terjadinya *overfitting*.
- c. Data uji (15%) digunakan setelah model selesai dilatih untuk mengukur kinerja akhir model pada data yang belum pernah dilihat sebelumnya. Bagian ini berfungsi menilai kemampuan generalisasi model terhadap data baru.

2. Standarisasi Fitur

Seluruh fitur numerik kemudian distandarisasi menggunakan metode *z-score*. Proses ini memastikan semua variabel berada pada skala yang sebanding, sehingga perhitungan gradien dan pembaruan bobot pada metode Gauss–Newton maupun iterasi Jacobi berjalan lebih stabil. Hasil proses standarisasi secara komputasional ditunjukkan pada Tabel 4.6 dan Tabel 4.7,

yang masing-masing menampilkan cuplikan dari keseluruhan 768 sampel dataset.

Tabel 4.6 Hasil Data Sebelum Standarisasi

No	Fitur							
	1	2	3	4	5	6	7	8
1	1	119	54	13	50	22.3	0.205	24
2	6	165	68	26	168	33.6	0.631	49
3	2	109	92	0	0	42.7	0.845	54
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
766	6	124	72	0	0	27.6	0.368	29
767	12	151	70	40	271	41.8	0.742	38
768	3	84	72	32	0	37.2	0.267	28

Tabel 4.7 Hasil Data Setelah Standarisasi

No	Fitur							
	1	2	3	4	5	6	7	8
1	-0.8448	-0.0592	-0.7809	-0.4727	-0.2587	-1.2301	-0.8059	-0.78862
2	0.6399	1.3803	-0.0571	0.3427	0.7658	0.2040	0.4805	1.3409
3	-0.5479	-0.3722	1.1835	-1.2882	-0.6928	1.3589	1.1268	1.7663
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
766	0.6399	0.0971	0.1496	-1.2882	-0.6928	-0.5575	-0.3137	-0.3608
767	2.4217	0.9422	0.0462	1.2209	1.6601	1.2447	0.8158	0.4049
768	-0.2509	-1.1546	0.1496	0.7190	-0.6928	0.6609	-0.6187	-0.4459

Dataset yang telah distandarisasi inilah yang kemudian digunakan sebagai input pada tahap pelatihan model regresi logistik menggunakan metode Gauss–Newton dan iterasi Jacobi pada subbab berikutnya.

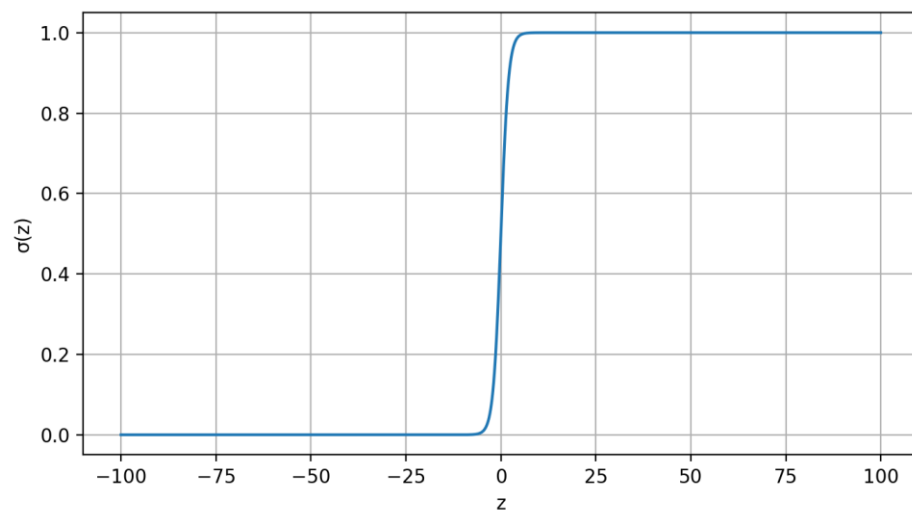
4.3.2 Penanganan Masalah Numerik dengan Clipping

Pada tahap awal implementasi, model mengalami kendala numerik akibat sifat eksponensial pada fungsi sigmoid. Ketika nilai z sangat besar (positif) atau sangat kecil (negatif), perhitungan $\exp(-z)$ menghasilkan nilai yang terlalu ekstrem (*overflow* atau *underflow*). Hal ini menyebabkan log *loss* menjadi tidak stabil dan

menghasilkan nilai *NaN*. Dampaknya, loss tidak dapat menurun secara konsisten, dan akurasi model sulit mencapai konvergensi.

Untuk mengatasi masalah tersebut, diterapkan fungsi *clipping* dengan memanfaatkan perintah `np.clip()` pada *Python*. Fungsi ini menjaga agar nilai z tetap berada dalam rentang yang aman, yaitu: $[-100, 100]$. Pembatasan ini mengikuti aturan sebagai berikut:

1. Jika $z > 100$, maka nilai z menjadi 100
2. Jika $z < -100$, maka nilai z menjadi -100
3. Jika $-100 \leq z \leq 100$, maka nilai z tetap dipertahankan tanpa perubahan



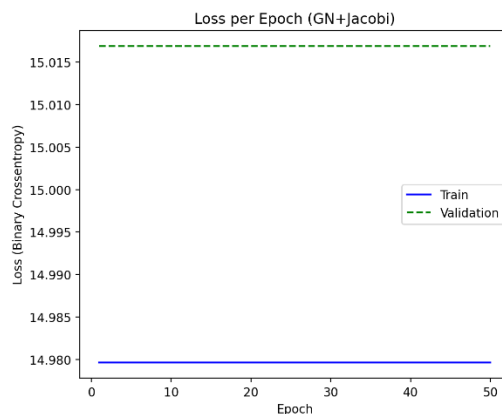
Gambar 4. 2 Grafik fungsi sigmoid dengan pembatasan nilai z_i untuk menjaga kestabilan numerik

Gambar 4.2 menunjukkan bahwa pada rentang nilai $z \in [-100, 100]$, fungsi sigmoid menghasilkan kurva yang stabil dan berada pada interval $[0, 1]$. Hal ini menunjukkan bahwa rentang tersebut merupakan rentang yang aman secara numerik. Oleh karena itu, pembatasan nilai z_i pada rentang ini dilakukan untuk

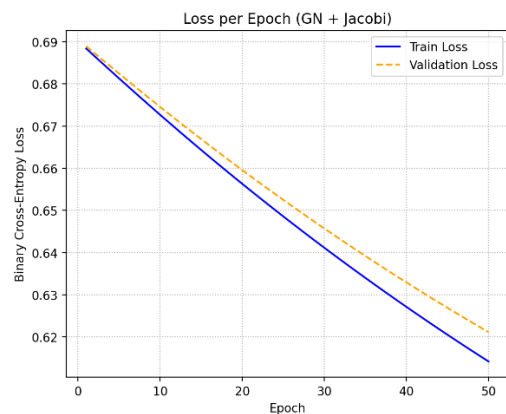
mencegah ketidakstabilan perhitungan tanpa mengubah karakteristik fungsi sigmoid.

Dengan pembatasan tersebut, nilai $\exp(-z_i)$ tidak akan menjadi terlalu besar atau terlalu kecil, sehingga hasil fungsi sigmoid tetap berada dalam rentang yang stabil, mendekati (0, 1) tanpa menyebabkan kesalahan komputasi. Konsep ini tidak mengubah bentuk fungsi sigmoid secara signifikan, tetapi mencegah probabilitas ekstrem yang dapat merusak kestabilan proses iteratif.

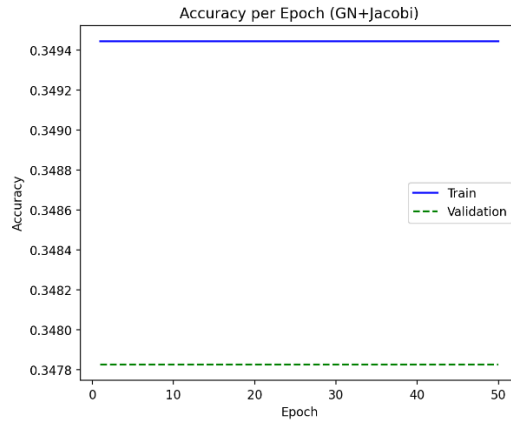
Efektivitas penerapan fungsi `np.clip()` diperlihatkan melalui perbandingan grafik *loss* dan akurasi sebelum dan sesudahnya penerapan *clipping*, sebagaimana ditampilkan pada Gambar 4.3 hingga Gambar 4.5.



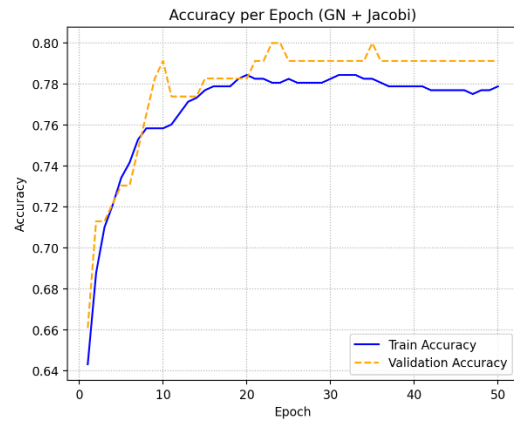
Gambar 4.3 Grafik Loss Sebelum Clipping



Gambar 4.4 Grafik Loss Sesudah Clipping



Gambar 4.5 Grafik Akurasi Sebelum Clipping



Gambar 4.6 Grafik Akurasi Sesudah Clipping

Hasil perbandingan menunjukkan bahwa sebelum diterapkan clipping, nilai *loss* cenderung tinggi dan stagnan, sedangkan akurasi yang rendah serta tidak mengalami peningkatan. Hal ini menunjukkan ketidakstabilan numerik akibat nilai eksponensial $\exp(z_i)$ yang terlalu besar atau terlalu kecil, sehingga proses pembaruan bobot tidak berjalan efektif.

Setelah diterapkan clipping pada rentang nilai $z \in [-100, 100]$, fungsi sigmoid menjadi lebih stabil karena terhindar dari *overflow* dan *underflow*. Akibatnya, perhitungan *loss* dapat dilakukan dengan benar, *loss* menurun secara konsisten, dan akurasi meningkat stabil baik pada data latih maupun validasi. Dengan demikian, penerapan fungsi `np.clip()` terbukti meningkatkan kestabilan komputasi, mempercepat konvergensi, dan menghasilkan model yang lebih akurat dibandingkan kondisi tanpa clipping.

4.3.3 Hasil pelatihan model

Hasil pelatihan model ditunjukkan pada Tabel 4.8, yang merangkum nilai *loss* dan akurasi pada data training, validasi, dan test tiap 5 *epoch*.

Tabel 4.8 Perkembangan *Loss* dan Akurasi Selama Proses Konvergensi

<i>Epoch</i>	<i>Training Loss</i>	<i>Validation Loss</i>	<i>Training Accuracy</i>	<i>Validation Accuracy</i>
1	0.6884	0.6889	0.6431	0.6609
5	0.6796	0.6809	0.7416	0.7304
11	0.6710	0.6730	0.7602	0.7739
16	0.6627	0.6654	0.7788	0.7826
21	0.6547	0.6581	0.7825	0.7913
26	0.6471	0.6511	0.7807	0.7913
31	0.6397	0.6444	0.7844	0.7913
36	0.6326	0.6379	0.7807	0.7913
41	0.6258	0.6317	0.7788	0.7913
46	0.6192	0.6257	0.7770	0.7913
50	0.6142	0.6211	0.7788	0.7913

Tabel 4.8 menunjukkan bahwa *loss* dan *accuracy* pada data latih dan data validasi turun konsisten seiring bertambahnya *epoch*, sementara *accuracy* meningkat hingga mencapai kestabilan (konvergen). Pada *epoch* awal (*epoch* 1), *training loss* dan *validation loss* masing-masing 0.6884 dan 0.6889, dengan *training accuracy* 0.6431 dan *validation accuracy* 0.6609 menandakan bobot masih jauh dari optimum. Seiring pembaruan bobot, performa membaik pada *epoch* 11, *validation accuracy* mencapai 0.7739. Hingga *epoch* 50, *training loss* dan *validation loss* turun menjadi 0.6142 dan 0.6211, dengan *training accuracy* 0.7788 dan *validation accuracy* 0.7913. Perubahan *loss* antar *epoch* yang semakin kecil dan *accuracy* pada akhir pelatihan menandakan bahwa model telah mencapai konvergensi, dengan performa validasi yang stabil tanpa *overfitting*.

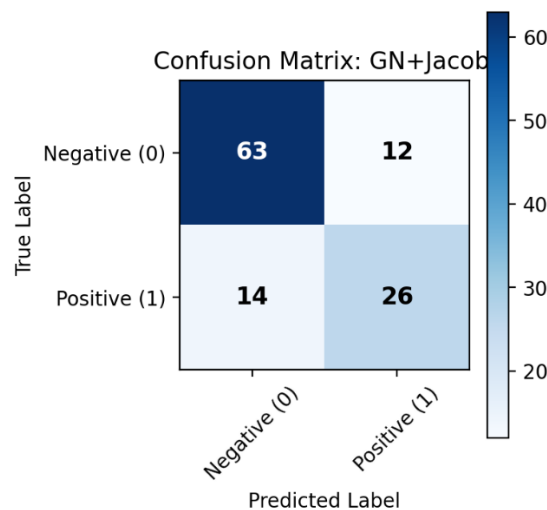
Perkembangan nilai *loss* dan *accuracy* selama proses pelatihan metode Gauss–Newton + iterasi Jacobi divisualisasikan pada Gambar 4.3 dan Gambar 4.5. Pada Gambar 4.3 menunjukkan penurunan yang halus dan hampir linear baik pada data latih maupun validasi. Kedua kurva saling berdekatan, menandakan bahwa model memiliki kestabilan tinggi dan tidak mengalami *overfitting*. Perbedaan kecil

antara *training loss* dan *validation loss* menunjukkan bahwa model mampu beradaptasi terhadap data baru tanpa kehilangan kemampuan generalisasi dan Gambar 4.5 memperlihatkan peningkatan tajam pada 10 *epoch* pertama, dari sekitar 0.64 menjadi lebih dari 0.77. Setelah itu, akurasi model cenderung stabil di kisaran 0.78–0.79 hingga *epoch* ke-50, yang mengindikasikan bahwa proses pembaruan bobot sudah mencapai titik konvergensi.

Secara keseluruhan, visualisasi ini menegaskan bahwa kombinasi metode Gauss–Newton dan iterasi Jacobi mampu mencapai konvergensi cepat, stabilitas numerik, dan akurasi tinggi tanpa indikasi *overfitting*. Nilai *validation accuracy* yang mendekati 0.8 (80%) menunjukkan bahwa model berhasil mempelajari pola yang representatif dari data dan memiliki kemampuan generalisasi yang baik. Selanjutnya, performa model terhadap data uji dibahas pada Subbab selanjutnya untuk menilai kemampuan generalisasi model secara menyeluruh.

4.3.4 Evaluasi pada data uji

Evaluasi pada data uji dinilai untuk menilai kemampuan model dalam mengklasifikasikan data baru yang tidak digunakan selama proses pelatihan. Hasil prediksi divisualisasikan dalam *confusion matrix* seperti ditunjukkan pada Gambar 4.8.



Gambar 4.7 *Confusion Matrix* Hasil Prediksi Metode Gauss-Newton + iterasi Jacobi

Berdasarkan Gambar 4.7, model Gauss-Newton + Iterasi Jacobi mampu mengklasifikasikan 63 data Non-Diabetes dan 26 data Diabetes secara benar. Namun, masih terdapat 12 data Non-Diabetes yang salah diprediksi sebagai Diabetes (*False Positive*) serta 14 data Diabetes yang salah diprediksi sebagai Non-Diabetes (*False Negative*). Berdasarkan hasil tersebut, diperoleh nilai akurasi sebesar 77.39%, yang menunjukkan bahwa model mampu melakukan klasifikasi dengan kinerja yang cukup baik secara keseluruhan pada data uji.

Perhitungan metrik evaluasi menunjukkan bahwa performa model pada kelas negatif (Non-Diabetes) lebih tinggi dibandingkan kelas positif, dengan presisi kelas negatif sebesar 81.82%, *recall* kelas negatif sebesar 84%, dan F1-score kelas negatif sebesar 82.89%. Kinerja yang lebih tinggi ini dipengaruhi oleh ketidakseimbangan data, di mana jumlah sampel kelas negatif lebih banyak sehingga model lebih mudah mempelajari pola dari kelas tersebut. Sementara itu, pada kelas positif (Diabetes) diperoleh presisi sebesar 68.42%, *recall* sebesar 65%, dan F1 score sebesar 66.67%, yang menunjukkan bahwa kemampuan model dalam mendeteksi kasus Diabetes masih lebih rendah.

Meskipun demikian, fokus evaluasi tetap diarahkan pada kelas positif karena kesalahan prediksi *False Negative* memiliki konsekuensi klinis yang jauh lebih serius. Pasien yang sebenarnya Diabetes tetapi diprediksi sebagai Non-Diabetes berpotensi tidak terdeteksi dan tidak memperoleh penanganan yang tepat waktu, sehingga meningkatkan risiko kesehatan. Sebaliknya, *False Positive* umumnya hanya menyebabkan pemeriksaan lanjutan yang tidak membahayakan pasien. Oleh karena itu, meskipun performa kelas negatif lebih tinggi, metrik pada kelas positif menjadi acuan utama dalam menilai kualitas model dalam konteks medis.

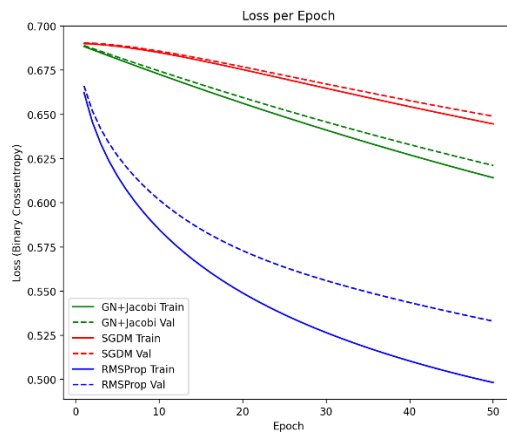
Secara keseluruhan, kombinasi Gauss–Newton + Iterasi Jacobi menghasilkan konvergensi cepat dan stabil serta generalisasi yang baik pada data uji, dengan tingkat kesalahan yang relatif kecil pada kedua kelas. Dengan demikian, pendekatan ini efektif untuk optimasi regresi logistik biner, karena mencapai stabilitas numerik dan akurasi tinggi dengan waktu pelatihan yang efisien. Pada subbab berikutnya, performa metode ini akan dibandingkan dengan algoritma lain (SGDM dan RMSProp) untuk menilai keunggulan relatif pendekatan yang diusulkan.

4.4 Analisis Hasil

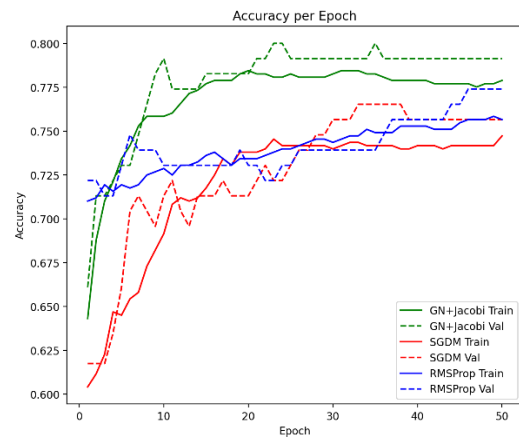
4.4.1 Perbandingan dengan Metode Dasar

Perbandingan antara metode Gauss–Newton + Iterasi Jacobi (GN+Jacobi), Stochastic Gradient Descent Momentum (SGDM), dan RMSProp dilakukan untuk menilai efektivitas pendekatan yang diusulkan dalam hal konvergensi, performa klasifikasi, serta efisiensi waktu komputasi. Hasil analisis disajikan dalam bentuk

grafik konvergensi (*loss* dan *accuracy* per *epoch*), tabel evaluasi metrik dan waktu eksekusi.



Gambar 4.8 Perbandingan Grafik *Loss* pada Data Training dan Validasi antara Gauss-Newton + iterasi Jacobi, SGDM, RMSProp



Gambar 4.9 Perbandingan Grafik Akurasi pada Data Training dan Validasi antara Gauss-Newton + iterasi Jacobi, SGDM, RMSProp

Berdasarkan Gambar 4.8 dan Gambar 4.9, seluruh metode menunjukkan penurunan nilai *loss* dan peningkatan *accuracy* seiring bertambahnya *epoch*, namun dengan pola konvergensi yang berbeda. Metode Gauss–Newton + Iterasi Jacobi memperlihatkan penurunan *loss* yang stabil dan konsisten sejak awal pelatihan, dengan arah konvergensi yang lebih terarah menuju nilai minimum. Hal ini disebabkan oleh proses pembaruan bobot melalui iterasi Jacobi yang menghasilkan perubahan bobot secara bertahap dan halus, sehingga mengurangi osilasi selama proses optimasi.

Berdasarkan grafik, metode GN+Jacobi mulai menunjukkan kestabilan akurasi validasi sejak sekitar *epoch* ke-10, yang mengindikasikan bahwa model telah mendekati kondisi konvergen. Namun, jika ditinjau secara numerik pada data validasi, nilai akurasi tertinggi dan stabil, yaitu sekitar 0.79 (79%), dipertahankan

secara konsisten mulai *epoch* ke-20 hingga akhir pelatihan. Oleh karena itu, *epoch* ke-20 dijadikan sebagai acuan konvergensi model yang stabil dalam penelitian ini.

Sebaliknya, SGDM dan RMSProp memerlukan lebih banyak *epoch* untuk mencapai stabilitas. SGDM cenderung fluktuatif di awal karena sensitivitas terhadap *learning rate*, sehingga arah konvergensinya sering beresilasi sebelum mendekati titik minimum. Sementara itu, RMSProp memiliki arah konvergensi yang lebih lambat tetapi bertahap, karena mekanisme adaptifnya menyesuaikan laju pembaruan pada setiap parameter sehingga proses mendekati optimum berlangsung lebih hati-hati.

Hal ini menunjukkan bahwa metode iterasi Jacobi dalam proses pembaruan bobot Gauss–Newton mempercepat proses konvergensi sekaligus menjaga arah konvergensi tetap stabil, tanpa fluktuasi besar pada *validation loss*. Dengan demikian, GN+Jacobi mampu mencapai kondisi konvergen dalam jumlah *epoch* yang lebih sedikit dibanding metode lainnya.

Secara kuantitatif, hasil perbandingan performa ketiga metode tersebut dirangkum pada Tabel 4.9 berikut. Pengujian dilakukan sebanyak lima kali (5 *running*) untuk memastikan konsistensi hasil, dan seluruh nilai yang diperoleh stabil pada setiap percobaan.

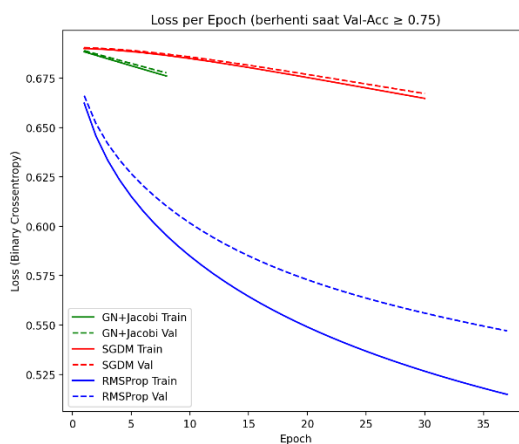
Tabel 4.9 Perbandingan Hasil Performa Gauss-Newton +Iterasi Jacobi, SGDM, RMSProp

Metode	Akurasi	Presisi	Recall	F1_Score
Gauss Newton + iterasi Jacobi	0.7739	0.6842	0.65	0.6667
SGDM	0.7217	0.5870	0.6750	0.6279
RMSProp	0.7565	0.6429	0.6750	0.6585

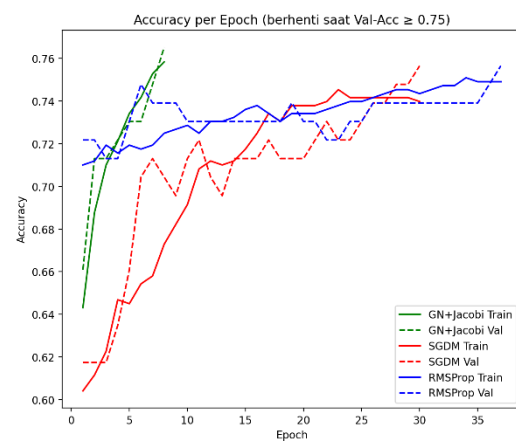
Kestabilan konvergensi GN+Jacobi berpengaruh langsung terhadap hasil klasifikasi akhir model. Berdasarkan *Confusion Matrix* pada Gambar 4.6, metode

ini berhasil mengklasifikasikan 63 data Non-Diabetes dan 26 data Diabetes secara benar, menghasilkan akurasi keseluruhan 77.39 %, precision 68.42 %, recall 65.00%, dan F1 Score 66.67 %. Hasil ini konsisten dengan Tabel 4.9, di mana GN+Jacobi memperoleh nilai tertinggi pada seluruh metrik utama, diikuti RMSProp dan SGDM. Dengan demikian, kestabilan konvergensi pada GN+Jacobi memberikan *trade-off* optimal antara precision dan recall, menjadikannya metode paling seimbang dalam mendeteksi kedua kelas.

Setelah performa tiga metode dibandingkan, analisis selanjutnya dilakukan terhadap mekanisme pemberhentian iterasi (*stopping criteria*). Proses pelatihan setiap metode dihentikan secara otomatis ketika *validation accuracy* mencapai 0.75 atau ketika peningkatan akurasi antar *epoch* menjadi sangat kecil.



Gambar 4.10 Grafik Perbandingan *Loss* dengan Pemberhentian Iterasi pada *Validation Accuracy* > 0.75 antara Gauss-Newton + iterasi Jacobi, SGDM, RMSProp



Gambar 4.11 Grafik Perbandingan Akurasi dengan Pemberhentian Iterasi pada *Validation Accuracy* > 0.75 antara Gauss-Newton + iterasi Jacobi, SGDM, RMSProp

Dari kedua grafik pada Gambar 4.10 dan 4.11 terlihat bahwa GN+Jacobi mencapai kondisi berhenti lebih cepat dengan pola penurunan loss yang stabil.

Sebaliknya, SGDM dan RMSProp masih mengalami fluktuasi akurasi pada validation set hingga *epoch* lebih tinggi. Hal ini menunjukkan bahwa kombinasi Gauss–Newton dan iterasi Jacobi mampu menyesuaikan pembaruan bobot secara efisien sehingga memenuhi kriteria berhenti lebih awal.

Selain performa prediktif, aspek efisiensi komputasi juga menjadi pertimbangan penting dalam menilai kualitas metode optimasi. Metode yang baik tidak hanya memiliki akurasi tinggi, tetapi juga mampu mencapainya dengan jumlah *epoch* dan waktu pelatihan minimal.

Tabel 4.10 Perbandingan *Epoch* Konvergensi ($acc-val < 0.75$) dan Waktu Eksekusi

Metode	<i>Epoch</i>	Total Time (ms)
GN+Jacobi	8	7.956
SGDM	30	7.152
RMSProp	37	10.828

Tabel 4.10 menunjukkan perbandingan jumlah *epoch* dan total waktu pelatihan pada tiga metode optimasi, yaitu GN+Jacobi, SGDM, dan RMSProp. Metode GN+Jacobi mencapai konvergensi hanya dalam 8 *epoch* dengan total waktu 7.956 ms, jauh lebih sedikit dibandingkan SGDM yang memerlukan 30 *epoch* dan RMSProp yang membutuhkan 37 *epoch*. Meskipun waktu komputasi per *epoch* pada GN+Jacobi relatif lebih besar dibandingkan SGDM, hal ini disebabkan oleh sifat metode Gauss–Newton yang termasuk metode optimasi orde dua, di mana setiap iterasi melibatkan perhitungan matriks aproksimasi Hessian dan penyelesaian sistem persamaan linier menggunakan iterasi Jacobi. Proses tersebut secara komputasi memang lebih kompleks dibandingkan metode orde satu seperti SGDM dan RMSProp yang hanya memanfaatkan informasi gradien.

Namun demikian, penggunaan informasi orde dua memungkinkan setiap langkah pembaruan bobot pada GN+Jacobi menjadi lebih informatif dan efektif dalam mendekati solusi optimum. Akibatnya, proses konvergensi dapat dicapai dalam jumlah iterasi yang jauh lebih sedikit, meskipun durasi setiap *epoch* relatif lebih lama. Dengan demikian, secara keseluruhan GN+Jacobi tetap menjadi metode yang paling efisien, karena mampu menghasilkan kondisi konvergen yang stabil dan akurat dengan jumlah *epoch* minimal serta waktu pelatihan total yang tetap kompetitif dibandingkan metode lainnya.

4.4.2 Perbandingan dengan Penelitian Sebelumnya

Untuk menilai posisi metode yang diusulkan, dilakukan perbandingan hasil dengan beberapa studi yang juga menggunakan dataset Pima Indian Diabetes. Perbandingan dilakukan terhadap metrik utama, yaitu *accuracy*, *precision*, *recall*, *F1_Score*, serta efisiensi waktu eksekusi. Ringkasan hasil perbandingan ditampilkan pada Tabel 4.11

Tabel 4.11 Perbandingan dengan Penelitian Sebelumnya

Autor	Metode	Accuracy	Presisi	Recall	F1_Score
Penelitian ini	GN+Iterasi Jacobi	0.7739	0.6842	0.6500	0.6667
(Reddy & Nagendra, 2019)	Decision Tree	0.7622	-	-	-
	Naïve Bayes	0.7736	-	-	-
	SVM	0.6868	-	-	-
(Xie, 2024)	Random Forest	0.7739	0.8000	0.5455	0.6486
	Logistic Regression	0.7913	0.8571	0.5455	0.6667
	SVM	0.7565	0.8077	0.4773	0.60

Pada tabel 4.11 memperlihatkan perbandingan performa metode *Gauss–Newton* yang dikombinasikan dengan iterasi Jacobi terhadap sejumlah metode klasik seperti *Decision Tree*, *Naïve Bayes*, *Support Vector Machine (SVM)*, *Random Forest*, dan *Logistic Regression* yang telah diuji pada penelitian terdahulu.

Secara umum metode yang diusulkan (GN+Jacobi) menunjukkan kinerja yang kompetitif dibandingkan penelitian sebelumnya. Dari sisi akurasi, model GN+Jacobi mencapai nilai 0.7739 yang berada di atas *Decision Tree* (0.7622), *Naïve Bayes* (0.7736), dan *SVM* (0.6868) pada penelitian Reddy & Nagendra, 2019). Meskipun *logistic regression* pada penelitian (Xie, 2024) memperoleh akurasi sedikit lebih tinggi yaitu 0.7913, selisih tersebut relative kecil (sekitar 1.7%).

Dari efektivitas metrik evaluasi, metode GN + Jacobi menampilkan keseimbangan yang baik antara presisi (0.6842), recall (0.65), dan F1_Score (0.6667). Jika dibandingkan dengan *Logistic Regression* yang memiliki presisi lebih tinggi (0.8571) namun recall lebih rendah (0.5455), maka model GN + Jacobi justru lebih sensitif dalam mendeteksi kasus positif (penderita diabetes). Dengan kata lain, GN+Jacobi mampu mempertahankan keseimbangan anantara ketepatan prediksi, sehingga performa klasifikasinya lebih stabil antar metrik evaluasi.

Selain itu, nilai F1_Score GN+Jacobi yang setara dengan *Logistic Regression* (0,6667) menunjukkan bahwa metode yang diusulkan memiliki kemampuan prediksi yang konsisten, tanpa fluktuasi besar antar metrik. Hal ini menandakan bahwa pembaruan bobot yang dihasilkan dari kombinasi Gauss Newton dan iterasi Jacobi mampu menstabilkan proses pembelajaran model secara efektif.

Secara keseluruhan, hasil perbandingan pada Tabel 4.11 menegaskan bahwa metode Gauss–Newton + Iterasi Jacobi memiliki efektivitas metrik evaluasi yang baik dan efisiensi waktu yang tinggi, sehingga layak dijadikan alternatif untuk optimasi regresi logistik biner pada kasus klasifikasi penyakit diabetes.

4.5 Analisis Integrasi Keislaman

Penelitian ini menunjukkan bahwa penerapan metode Gauss-Newton dan iterasi Jacobi dalam regresi logistik biner merupakan bentuk ikhtiar ilmiah yang dilakukan manusia untuk memperoleh hasil terbaik. Dalam Islam, kerja keras dan usaha sungguh-sungguh merupakan nilai yang sangat ditekankan. Allah SWT berfirman dalam QS. An-Najm: 39 (Kemenag, 2024).

وَأَنْ لَّيْسَ لِلْإِنْسَانِ إِلَّا مَا سَعَى ﴿٣٩﴾

Artinya: "Dan bahwa manusia hanya memperoleh apa yang telah diusahakannya."

Ayat ini menegaskan bahwa keberhasilan yang diperoleh seorang peneliti adalah buah dari jerih payah, ketekunan, serta kesungguhan dalam mengolah data dan menerapkan metode yang tepat. Proses optimasi yang membutuhkan perhitungan berulang-ulang dan validasi sistematis dapat dipandang sebagai manifestasi dari sunnatullah, di mana hukum-hukum Allah di alam semesta dapat dipahami melalui ikhtiar akal manusia.

Lebih jauh, hasil penelitian yang diperoleh bukan sekadar pencapaian akademis, tetapi juga memiliki dimensi spiritual. Ikhtiar dalam memahami keteraturan data melalui metode matematis dapat menjadi sarana tafakur atas keteraturan ciptaan Allah SWT. Pola-pola yang diungkapkan melalui regresi logistik menunjukkan betapa teraturnya sunnatullah yang berlaku, dan hal ini mengarahkan peneliti untuk

semakin mengagumi kebesaran Allah SWT. Dengan demikian, setiap langkah penelitian, mulai dari pengolahan data hingga interpretasi hasil, dapat dimaknai sebagai bentuk pengabdian kepada Allah SWT melalui penggunaan akal yang dianugerahkan-Nya.

Selain menekankan pentingnya ikhtiar, Islam juga mengajarkan nilai kejujuran dan amanah dalam seluruh aspek kehidupan, termasuk penelitian ilmiah. Allah SWT berfirman dalam QS. At-Taubah: 119 (Kemenag, 2024).

يَا أَيُّهَا الَّذِينَ آمَنُوا اتَّقُوا اللَّهَ وَكُونُوا مَعَ الصَّادِقِينَ ﴿١١٩﴾

Artinya: "Wahai orang-orang yang beriman! Bertakwalah kepada Allah, dan hendaklah kamu bersama orang-orang yang jujur."

Ayat ini menegaskan bahwa setiap peneliti harus bersikap jujur dalam mengolah data, melaporkan hasil, dan menarik kesimpulan. Kejujuran dalam penelitian berarti tidak melakukan manipulasi data, tidak menyembunyikan kelemahan metode, serta tidak menambahkan informasi yang tidak didukung oleh bukti. Prinsip ini sangat penting agar hasil penelitian dapat dipertanggungjawabkan, baik secara akademik maupun moral, sehingga membawa manfaat yang luas.

Dengan demikian, kajian Islam dalam penelitian ini memperlihatkan bahwa ilmu pengetahuan dan agama tidak dapat dipisahkan. Usaha ilmiah dalam mengoptimalkan regresi logistik biner melalui metode Gauss-Newton dan iterasi Jacobi merupakan perwujudan dari nilai ikhtiar, kerja keras, dan kejujuran yang diajarkan Islam. Hasil penelitian tidak hanya bermanfaat bagi perkembangan ilmu pengetahuan, tetapi juga memperkaya aspek spiritual peneliti dalam bentuk pengakuan akan kebesaran Allah SWT. Oleh karena itu, penelitian ini dapat

dipandang sebagai ibadah ilmiah, yakni usaha mendekatkan diri kepada Allah SWT melalui ilmu yang bermanfaat dan dapat memberi kontribusi positif bagi masyarakat.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil dan pembahasan yang telah dilakukan, maka diperoleh kesimpulan sebagai berikut:

1. Penelitian ini berhasil menerapkan skema Gauss-Newton untuk meminimalkan *loss* logistik melalui pendekatan kuadrat terkecil taklinier. Pada tiap iterasi, model dilinearisasi di sekitar bobot sehingga diperoleh SPL berbentuk $HW = C$ dengan pendekatan $H \approx A^T A$ dan $C \approx A^T B$. SPL tersebut diselesaikan menggunakan iterasi Jacobi melalui dekomposisi $H = D + L + U$ dan pembaruan $W^{k+1} = D^{-1}(C - (L + U)w^k)$, hingga memenuhi kriteria konvergensi. Strategi ini menghindari invers matriks eksplisit, mudah diimplementasikan, dan stabil untuk ukuran masalah yang diteliti.
2. Pada dataset Pima Indians Diabetes, kombinasi GN+Jacobi menunjukkan kinerja sebagai berikut: akurasi 77.39%, precision 68.42%, recall 65.00%, dan F1 Score 66.67%. Kriteria henti validasi ≥ 0.75 tercapai dalam 8 *epoch* dengan waktu sekitar 7.956 ms. Dibanding SGDM dan RMSProp, penurunan loss GN+Jacobi lebih stabil dan mencapai konvergensi lebih awal pada data ini, sehingga memenuhi aspek efektivitas (performansi klasifikasi memadai) sekaligus efisiensi (jumlah *epoch*/waktu lebih rendah).

5.2 Saran

Beberapa saran yang dapat diberikan adalah sebagai berikut:

1. Penelitian selanjutnya disarankan untuk menerapkan teknik penyeimbangan data.
2. Dapat dilakukan pengujian terhadap dataset lain dengan karakteristik berbeda (misalnya data yang seimbang atau memiliki lebih banyak fitur) untuk menilai konsistensi performa metode gabungan ini.

DAFTAR PUSTAKA

- Afifah, D. N. (2020). *Penerapan metode regresi logistik biner pada kesejahteraan rumah tangga di Kabupaten Mojokerto*. Universitas Islam Negeri Maulana Malik Ibrahim.
- Bishop, C. M., & Nasrabadi, N. M. (2006). *Pattern recognition and machine learning* (Vol. 4, Issue 4). Springer.
- Björck, Å. (2024). *Numerical methods for least squares problems*. SIAM.
- Burden, R. L., & Faires, J. D. (2011). *Numerical Analysis*. Brooks/Cole, CengageLearning. *Boston, USA*.
- Darmono, P. B. (2012). Solusi Sistem Persamaan Linear dengan Metode Jacobi. *Limit*, 02, 222166.
- Goodfellow, I. (2016). *Deep learning*. MIT press.
- Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied logistic regression*. John Wiley & Sons.
- Islamiyati, A. (2015). Estimasi Parameter Model Regresi Logistik Biner Komponen Utama Non Linear dengan Maksimum Likelihood. *Jurnal Matematika Statistika Dan Komputasi*, 11(2), 122–128.
- Jamhuri, M., Mukhlash, I., & Irawan, M. I. (2022). Performance Improvement of Logistic Regression for Binary Classification by Gauss-Newton Method. *ACM International Conference Proceeding Series*, 12–16. <https://doi.org/10.1145/3545839.3545842>
- Jamhuri, M., & Subiono, S. (2021). Penentuan Nilai Awal Iterasi pada Masalah Pendugaan Parameter Regresi Taklinier. *Limits: Journal of Mathematics and Its Applications*, 18(2), 155. <https://doi.org/10.12962/limits.v18i2.8297>
- Kemenag. (2024). Kementrian Agama. *Qur'an Kemenag*.
- Marzuki, C. C., & Herawati. (2015). Penyelesaian Sistem Persamaan Linear Fully Fuzzy Menggunakan Metode Iterasi Jacobi. *Jurnal Sains Matematika Dan Statistika*, 1(1), 1–7.
- Menard, S. (2001). *Applied logistic regression analysis*. SAGE publications.
- Nisa, A., & Asih, T. S. N. (2020). *UNNES Journal of Mathematics*. 9(1), 48–57.
- Press, W. H. (2007). *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press.

- Quarteroni, A., Sacco, R., & Saleri, F. (2010). *Numerical mathematics* (Vol. 37). Springer Science & Business Media.
- Rahmadani, A. A., Putri, A. A., Happy, D. M., Alensia, M., Dala, D., Angka, M. T., Rafiq, M., & Wasono. (2023). Analisis Regresi Logistik Biner Untuk Memprediksi Faktor-Faktor Internal Yang Memengaruhi Keharmonisan Rumah Tangga Menurut Provinsi Di Indonesia Pada Tahun 2021. *Jurnal Fmipa Unmul*, 3(1), 116–127. <http://jurnal.fmipa.unmul.ac.id/index.php/SNMSA/article/view/1177/527>
- Roflin, E., Riana, F., Munarsih, E., & Liberty, I. A. (2023). *Regresi Logistik Biner Dan Multinomial*. Penerbit NEM.
- Saad, Y. (2003). *Iterative methods for sparse linear systems*. SIAM.
- Saleh, M. A. (2010). Penaksiran Parameter Regresi Nonlinier dengan Algoritma Gauss-Newton dan Tafsiran Geometris Least Squares. *Jurnal Matematika, Statistika, & Komputasi*, 7(1), 39–48.
- Salim, A., & Alfian, M. R. (2019). Optimalisasi Regresi Logistik Menggunakan Algoritma Genetika Pada Data Klasifikasi. *Jurnal Teknologi Informasi Dan Terapan*, 6(2), 50–55. <https://doi.org/10.25047/jtit.v6i2.109>
- Sauer, T. (2017). *numerical analysis*. Pearson.
- Siregar, R. W., Tulus, & Ramli, M. (2018). Analysis Local Convergence of Gauss-Newton Method. *IOP Conference Series: Materials Science and Engineering*, 300(1). <https://doi.org/10.1088/1757-899X/300/1/012044>
- Smith, J. W., Everhart, J. E., Dickson, W. C., Knowler, W. C., & Johannes, R. S. (1988). Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. *Proceedings of the Annual Symposium on Computer Application in Medical Care*, 261.
- Sutrisno, E. (2020). *Aplikasi quasicomplete separation dengan metode maximum likelihood pada regresi logistik*. Universitas Islam Negeri Maulana Malik Ibrahim.
- Thammi Reddy, A., & Nagendra, M. (2019). Minimal rule-based classifiers using PCA on pima-indians-diabetes-dataset. *International Journal of Innovative Technology and Exploring Engineering*, 8(12), 4414–4420. <https://doi.org/10.35940/ijitee.L2476.1081219>
- Umaroh, A. K. (2020). *Perbandingan Metode Regresi Logistik Biner dan Classification and Regression Tree pada Klasifikasi Status Kesejahteraan Rumah Tangga di Kota Batu*. Universitas Islam Negeri Maulana Malik Ibrahim.
- Widya, E., Syamsudhuha, S., & Bustami, B. (2014). *Metode Iterasi Jacobi Dan Gauss-Seidel Prekondisi Untuk Menyelesaikan Sistem Persaman Linear*

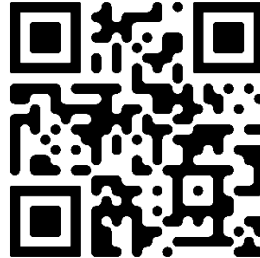
Dengan M-Matriks. Riau University.

Wright, S. J. (2006). *Numerical optimization*. Springer.

Xie, L. (2024). Pima Indian Diabetes Database and Machine Learning Models for Diabetes Prediction. *Highlights in Science, Engineering and Technology*, 88, 97–103. <https://doi.org/10.54097/z7hz7j81>

LAMPIRAN

Lampiran 1 Data Pima Indian Diabetes



Lampiran 2 Script Phyton Metode Gauss-Newton

```
%matplotlib inline
%config InlineBackend.figure_format = 'retina'

# ===== Final Script: GN+Jacobi, SGDM, RMSProp +
Confusion Matrix =====
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import confusion_matrix, classification_report

# ----- Data Preparation & Scaling -----
data = pd.read_csv('/kaggle/input/pima-indians-diabetes-database/diabetes.csv')
X_raw = data.drop('Outcome', axis=1).values
y = data['Outcome'].values

# Split data first, to avoid data leakage!
X_trainval, X_test, y_trainval, y_test = train_split = train_test_split(
    X_raw, y, test_size=115, random_state=42, stratify=y)
X_train, X_val, y_train, y_val = train_test_split(
    X_trainval, y_trainval, test_size=115, random_state=42, stratify=y_trainval)
# 0.1765 x 0.85  $\approx$  0.15

# Pastikan X_train, X_val, X_test sudah ada
def ensure_df(X):
    if isinstance(X, pd.DataFrame):
        return X.copy()
    n_features = X.shape[1]
    cols = [f"fitur{i}" for i in range(1, n_features+1)]
    return pd.DataFrame(X, columns=cols)

X_train = ensure_df(X_train)
X_val = ensure_df(X_val)
```

```

X_test = ensure_df(X_test)

# Gabungkan semua data
X_all = pd.concat([X_train, X_val, X_test], axis=0, ignore_index=True)

# Standarisasi (fit hanya pada data train)
scaler = StandardScaler()
X_all_scaled = pd.DataFrame(
    scaler.fit_transform(X_all),
    columns=X_all.columns
)

# Standarisasi fitur
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_val_scaled = scaler.transform(X_val)
X_test_scaled = scaler.transform(X_test)

# --- Menampilkan seperti di Lampiran PDF ---
def tampilkan_ringkas(df, n=5):
    """Menampilkan 5 data awal dan 5 data akhir seperti format lampiran."""
    print(df.head(n).to_string(index=False))
    print("...")
    print(df.tail(n).to_string(index=False))
    print(f"\n[{len(df)} rows x {df.shape[1]} columns]\n")

print("=== SEBELUM STANDARISASI (Seluruh Data) ===")
tampilkan_ringkas(X_all)

print("=== SESUDAH STANDARISASI (Seluruh Data) ===")
tampilkan_ringkas(X_all_scaled)

# Simpan ke CSV untuk lampiran PDF
X_all.to_csv("hasil_sebelum_standarisasi.csv", index=False)
X_all_scaled.to_csv("hasil_sesudah_standarisasi.csv", index=False)

# Add bias term
X_train = np.hstack([np.ones((X_train_scaled.shape[0],1)), X_train_scaled])
X_val = np.hstack([np.ones((X_val_scaled.shape[0],1)), X_val_scaled])
X_test = np.hstack([np.ones((X_test_scaled.shape[0],1)), X_test_scaled])

print(f"Train: {X_train.shape}, Val: {X_val.shape}, Test: {X_test.shape}")

# ----- Helper Functions -----
def sigmoid(z):
    z = np.clip(z, -100, 100)
    return 1 / (1 + np.exp(-z))

```

```

def loss(y_true, y_pred, eps=1e-10):
    return -np.mean(y_true * np.log(y_pred + eps) + (1 - y_true) * np.log(1 -
y_pred + eps))

def accuracy(y_true, y_pred_proba, threshold=0.5):
    y_pred = (y_pred_proba >= threshold).astype(int)
    return np.mean(y_pred == y_true)

def grad_logreg(X, y, w):
    z = X @ w
    h = sigmoid(z)
    return -(X.T @ (y - h)) / len(y)

def jacobi_solver(H, C, max_iter=100, tol=1e-6):
    n = H.shape[0]
    x = np.zeros(n)
    D = np.diag(H)
    R = H - np.diagflat(D)
    D_safe = np.where(np.abs(D) < 1e-12, 1e-12, D)
    for itr in range(max_iter):
        x_new = (C - R @ x) / D_safe
        if np.linalg.norm(x_new - x) < tol:
            x = x_new
            break
    x = x_new
    return x

def logistic_update_jacobi(X, y, w, alpha=0.01, jacobi_iter=100,
clip_value=1.0):
    # GN step with A=(y-h)*X and B = -[y log h + (1-y) log(1-h)] per-sample
    z = X @ w
    h = sigmoid(z)
    # Vectorized A and B
    A = (y - h).reshape(-1, 1) * X
    eps = 1e-10
    B = - (y * np.log(h + eps) + (1 - y) * np.log(1 - h + eps))
    H = A.T @ A
    C = A.T @ B
    delta_w = jacobi_solver(H, C, max_iter=jacobi_iter)
    delta_w = np.clip(delta_w, -clip_value, clip_value)
    w_new = w + alpha * delta_w
    return w_new, delta_w

# ----- Jacobi (Gauss-Newton + Jacobi) Training Loop -----
-----
np.random.seed(42)
n_features = X_train.shape[1]
epochs = 50

```

```

w_jacobi = np.random.randn(n_features) * 0.01
alpha_jacobi = 0.01
jacobi_iter = 100
clip_value = 0.3

jacobi_loss_train, jacobi_loss_val = [], []
jacobi_acc_train, jacobi_acc_val = [], []

%%time
for epoch in range(epochs):
    w_jacobi, delta_w = logistic_update_jacobi(
        X_train, y_train, w_jacobi,
        alpha=alpha_jacobi, jacobi_iter=jacobi_iter, clip_value=clip_value)
    h_train = sigmoid(X_train @ w_jacobi)
    h_val = sigmoid(X_val @ w_jacobi)
    jacobi_loss_train.append(loss(y_train, h_train))
    jacobi_loss_val.append(loss(y_val, h_val))
    jacobi_acc_train.append(accuracy(y_train, h_train))
    jacobi_acc_val.append(accuracy(y_val, h_val))
    if epoch % 5 == 0 or epoch == epochs - 1:
        print(f"[GN+Jacobi] Epoch {epoch+1:2d} | "
              f"Loss-tr={jacobi_loss_train[-1]:.4f} Acc-tr={jacobi_acc_train[-1]:.4f} "
              | "
              f"Loss-val={jacobi_loss_val[-1]:.4f} Acc-val={jacobi_acc_val[-1]:.4f}")

# Akurasi & prob test Jacobi
h_test_jacobi = sigmoid(X_test @ w_jacobi)
test_acc_jacobi = accuracy(y_test, h_test_jacobi)
print("Akurasi test akhir GN+Jacobi:", test_acc_jacobi)

epochs_list = np.arange(1, len(jacobi_loss_train) + 1)

plt.figure(figsize=(12, 5))

# ----- Plot Loss -----
plt.subplot(1, 2, 1)
plt.plot(epochs_list, jacobi_loss_train, label='Train Loss', color='blue')
plt.plot(epochs_list, jacobi_loss_val, label='Validation Loss', color='orange',
         linestyle='--')
plt.xlabel('Epoch')
plt.ylabel('Binary Cross-Entropy Loss')
plt.title('Loss per Epoch (GN + Jacobi)')
plt.legend()
plt.grid(True, linestyle=':')

# ----- Plot Accuracy -----

```



```

plt.subplot(1, 2, 2)
plt.plot(epochs_list, jacobi_acc_train, label='Train Accuracy', color='blue')
plt.plot(epochs_list, jacobi_acc_val, label='Validation Accuracy',
color='orange', linestyle='--')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.title('Accuracy per Epoch (GN + Jacobi)')
plt.legend()
plt.grid(True, linestyle=':')

plt.tight_layout()
plt.show()

# ----- SGDM Training Loop -----
np.random.seed(42)
w_sgdm = np.random.randn(n_features) * 0.01
v = np.zeros_like(w_sgdm)
alpha_sgdm = 0.01
beta_sgdm = 0.9

sgdm_loss_train, sgdm_loss_val = [], []
sgdm_acc_train, sgdm_acc_val = [], []

%%time
for epoch in range(epochs):
    grad = grad_logreg(X_train, y_train, w_sgdm)
    v = beta_sgdm * v + (1 - beta_sgdm) * grad
    w_sgdm -= alpha_sgdm * v
    h_train = sigmoid(X_train @ w_sgdm)
    h_val = sigmoid(X_val @ w_sgdm)
    sgdm_loss_train.append(loss(y_train, h_train))
    sgdm_loss_val.append(loss(y_val, h_val))
    sgdm_acc_train.append(accuracy(y_train, h_train))
    sgdm_acc_val.append(accuracy(y_val, h_val))
    if epoch % 5 == 0 or epoch == epochs-1:
        print(f"[SGDM] Epoch {epoch+1:2d}: Loss-train={sgdm_loss_train[-1]:.4f}, "
              f"Acc-train={sgdm_acc_train[-1]:.4f} | Loss-val={sgdm_loss_val[-1]:.4f}, "
              f"Acc-val={sgdm_acc_val[-1]:.4f}")

# Akurasi & prob test SGDM
h_test_sgdm = sigmoid(X_test @ w_sgdm)
test_acc_sgdm = accuracy(y_test, h_test_sgdm)
print("Akurasi test akhir SGDM:", test_acc_sgdm)

# ----- RMSProp Training Loop -----
np.random.seed(42)

```

```

w_rms = np.random.randn(n_features) * 0.01
s = np.zeros_like(w_rms)
alpha_rms = 0.01
beta_rms = 0.9
eps_rms = 1e-8

rms_loss_train, rms_loss_val = [], []
rms_acc_train, rms_acc_val = [], []

%%time
for epoch in range(epochs):
    grad = grad_logreg(X_train, y_train, w_rms)
    s = beta_rms * s + (1 - beta_rms) * (grad ** 2)
    w_rms -= alpha_rms * grad / (np.sqrt(s) + eps_rms)
    h_train = sigmoid(X_train @ w_rms)
    h_val = sigmoid(X_val @ w_rms)
    rms_loss_train.append(loss(y_train, h_train))
    rms_loss_val.append(loss(y_val, h_val))
    rms_acc_train.append(accuracy(y_train, h_train))
    rms_acc_val.append(accuracy(y_val, h_val))
    if epoch % 5 == 0 or epoch == epochs-1:
        print(f"[RMSProp] Epoch {epoch+1:2d}: Loss-train={rms_loss_train[-1]:.4f}, "
              f"Acc-train={rms_acc_train[-1]:.4f} | Loss-val={rms_loss_val[-1]:.4f}, "
              f"Acc-val={rms_acc_val[-1]:.4f}")

# Akurasi & prob test RMSProp
h_test_rms = sigmoid(X_test @ w_rms)
test_acc_rms = accuracy(y_test, h_test_rms)
print("Akurasi test akhir RMSProp:", test_acc_rms)

# ----- Plot Loss & Accuracy (All Methods) -----
--
epochs_list = np.arange(1, epochs+1)

plt.figure(figsize=(14, 6))
plt.subplot(1, 2, 1)
plt.plot(epochs_list, jacobi_loss_train, 'g-', label='GN+Jacobi Train')
plt.plot(epochs_list, jacobi_loss_val, 'g--', label='GN+Jacobi Val')
plt.plot(epochs_list, sgdm_loss_train, 'r-', label='SGDM Train')
plt.plot(epochs_list, sgdm_loss_val, 'r--', label='SGDM Val')
plt.plot(epochs_list, rms_loss_train, 'b-', label='RMSProp Train')
plt.plot(epochs_list, rms_loss_val, 'b--', label='RMSProp Val')
plt.xlabel("Epoch")
plt.ylabel("Loss (Binary Crossentropy)")
plt.title("Loss per Epoch")
plt.legend()

```

```

plt.subplot(1, 2, 2)
plt.plot(epochs_list, jacobi_acc_train, 'g-', label='GN+Jacobi Train')
plt.plot(epochs_list, jacobi_acc_val, 'g--', label='GN+Jacobi Val')
plt.plot(epochs_list, sgdm_acc_train, 'r-', label='SGDM Train')
plt.plot(epochs_list, sgdm_acc_val, 'r--', label='SGDM Val')
plt.plot(epochs_list, rms_acc_train, 'b-', label='RMSProp Train')
plt.plot(epochs_list, rms_acc_val, 'b--', label='RMSProp Val')
plt.xlabel("Epoch")
plt.ylabel("Accuracy")
plt.title("Accuracy per Epoch")
plt.legend()

plt.tight_layout()
plt.show()

def show_results(name, y_true, y_proba, thr=0.5):
    y_pred = (y_proba >= thr).astype(int)
    cm = confusion_matrix(y_true, y_pred)
    tn, fp, fn, tp = cm.ravel()
    acc = np.mean(y_pred == y_true)

    # ---- cetak teks hasil ----
    print(f"\n[{name}]")
    print(f"Accuracy : {acc:.4f}")
    print("Confusion Matrix:")
    print(cm)
    print(f"TP={tp} | TN={tn} | FP={fp} | FN={fn}")
    print("Classification Report:")
    print(classification_report(y_true, y_pred, digits=4))

    # ---- tampilkan gambar confusion matrix ----
    plt.figure(figsize=(4, 4))
    plt.imshow(cm, interpolation='nearest', cmap=plt.cm.Blues)
    plt.title(f"Confusion Matrix: {name}")
    plt.colorbar()
    tick_marks = np.arange(2)
    classes = ['Negative (0)', 'Positive (1)']
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)

    # tambahkan angka di setiap kotak
    thresh = cm.max() / 2.0
    for i, j in np.ndindex(cm.shape):
        plt.text(j, i, format(cm[i, j], 'd'),
                 ha="center", va="center",
                 color="white" if cm[i, j] > thresh else "black",
                 fontsize=12, fontweight='bold')

```

```

plt.ylabel('True Label')
plt.xlabel('Predicted Label')
plt.tight_layout()
plt.show()

# Jacobi (GN+Jacobi)
show_results("GN+Jacobi", y_test, h_test_jacobi)

# SGDM
show_results("SGDM", y_test, h_test_sgdm)

# RMSProp
show_results("RMSProp", y_test, h_test_rms)

```

Lampiran 3 Script Phyton Metode Gauss-Newton

Run	Metode	Epoch	Time	Akurasi
1	GN+IT	8	6.91	0.7739
	SGDM	30	6.44	0.7217
	RMSPROP	37	9.54	0.7565
2	GN+IT	8	7.87	0.7739
	SGDM	30	7.63	0.7217
	RMSPROP	37	11.7	0.7565
3	GN+IT	8	7.39	0.7739
	SGDM	30	7.39	0.7217
	RMSPROP	37	11	0.7565
4	GN+IT	8	6.91	0.7739
	SGDM	30	6.91	0.7217
	RMSPROP	37	12.6	0.7565
5	GN+IT	8	10.7	0.7739
	SGDM	30	7.39	0.7217
	RMSPROP	37	9.3	0.7565

RIWAYAT HIDUP



Rienke Anastasya Wyanti, biasa dipanggil Rienke, Lahir di Batu, pada tanggal 15 Juni 2003. Penulis merupakan anak pertama dari dua bersaudara dari pasangan Bapak Wintoko dan Ibu Siti Fatimah. Penulis bertempat tinggal di Dusun Dadapan Kulon, RT.03/RW.04, Desa Bendosari, Kecamatan Pujon, Kabupaten Malang. Pendidikan dasar ditempuh di SDN Bendosari 03, lalu melanjutkan pendidikan menengah pertama di SMPN 01 Ngantang, kemudian melanjutkan pendidikan menengah atas di MAN Kota Batu. Penulis melanjutkan studi di Universitas Islam Negeri Maulana Malik Ibrahim Malang pada Program Studi Matematika, Fakultas Sains dan Teknologi melalui jalur Mandiri. Selama masa studi berfokus pada pengembangan kemampuan akademik di bidang matematika terapan. Selain kegiatan akademik, penulis juga aktif terlibat dalam berbagai kegiatan kepanitiaan yang diselenggarakan oleh Himpunan Mahasiswa Program Studi (HMPS) dan Dewan Mahasiswa (DEMA) Fakultas Sains dan Teknologi. Sebagai bentuk penerapan ilmu pengetahuan serta pengalaman di dunia kerja, penulis telah melaksanakan Praktik Kerja Lapangan (PKL) di PT ASABRI (Persero) Kantor Cabang Malang. Apabila terdapat pertanyaan, saran, maupun kritik terkait penelitian ini, penulis dapat dihubungi melalui *e-mail*: rienkeanastasyawyanti@gmail.com



KEMENTERIAN AGAMA RI
UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Gajayana No.50 Dinoyo Malang Telp. / Fax. (0341)558933

BUKTI KONSULTASI SKRIPSI

Nama : Rienke Anastasya Wyanti
NIM : 210601110103
Fakultas / Jurusan : Sains dan Teknologi / Matematika
Judul Skripsi : Penerapan Metode Gauss-Newton dan Iterasi Jacobi pada Optimasi Regresi Logistik Biner
Pembimbing I : Dr. Mohammad Jamhuri, M.Si.
Pembimbing II : Erna Herawati, M.Pd.

No	Tanggal	Hal	Tanda Tangan
1.	16 September 2024	Konsultasi Topik dan Data	1.
2.	16 Oktober 2024	Konsultasi Bab I, II, dan III	2.
3.	28 Oktober 2024	Konsultasi Bab I, II, dan III	3.
4.	5 November 2024	Konsultasi Bab I, II, dan III	4.
5.	12 November 2024	Konsultasi Bab I, II, dan III	5.
6.	13 Januari 2025	ACC Bab I, II, dan III	6.
7.	14 Januari 2025	Konsultasi Kajian Agama Bab I dan II	7.
8.	16 Januari 2025	Konsultasi Kajian Agama Bab I dan II	8.
9.	23 Januari 2025	Konsultasi Kajian Agama Bab I dan II	9.
10.	6 Februari 2025	ACC Kajian Agama Bab I dan II	10.
11.	28 Maret 2025	ACC Seminar Proposal	11.
12.	15 Juli 2025	Konsultasi Revisi Seminar Proposal	12.
13.	30 Juli 2025	Konsultasi Bab IV dan V	13.
14.	6 Agustus 2025	Konsultasi Bab IV dan V	14.
15.	20 Agustus 2025	Konsultasi Bab IV dan V	15.



KEMENTERIAN AGAMA RI
UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Gajayana No.50 Dinoyo Malang Telp. / Fax. (0341)558933

16.	11 September 2025	Konsultasi Bab IV dan V	16.
17.	17 September 2025	Konsultasi Bab IV dan V	17.
18.	19 September 2025	Konsultasi Bab IV dan V	18.
19.	22 September 2025	ACC Bab IV dan V	19.
20.	8 Oktober 2025	Konsultasi Kajian Agama Bab IV	20.
21.	14 Oktober 2025	ACC Kajian Agama Bab IV	21.
22.	6 November 2025	ACC Seminar Hasil	22.
23.	10 November 2025	ACC Seminar Hasil lanjutan	23.
24.	24 November 2025	Konsultasi Revisi Seminar Hasil	24.
25.	16 Desember 2025	Sidang Skripsi	25.
26.	23 Desember 2025	ACC Keseluruhan	26.

Malang, 23 Desember 2025

Mengetahui,

Ketua Program Studi Matematika



Dr. Fachrur Rozi, M.Si.

NIP. 19800527 200801 1 012