OPTIMASI REGRESI LOGISTIK MENGGUNAKAN PENDEKATAN *LEVENBERG-MARQUARDT* DAN *GAUSS SEIDEL* UNTUK PENYELESAIAN KLASIFIKASI BINER

SKRIPSI

OLEH
ARIDA NUR BAIZURIANA
NIM. 210601110084

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

OPTIMASI REGRESI LOGISTIK MENGGUNAKAN PENDEKATAN *LEVENBERG-MARQUARDT* DAN *GAUSS SEIDEL* UNTUK PENYELESAIAN KLASIFIKASI 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
Arida Nur Baizuriana
NIM. 210601110084

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

OPTIMASI REGRESI LOGISTIK MENGGUNAKAN PENDEKATAN *LEVENBERG-MARQUARDT* DAN *GAUSS SEIDEL* UNTUK PENYELESAIAN KLASIFIKASI BINER

SKRIPSI

Oleh
Arida Nur Baizuriana
NIM. 210601110084

Telah Disetujui Untuk Diuji

Malang, 8 Desember 2025

Dosen Pembimbing I	Dosen Pembimbing II
Dr. Mohammad Jamhuri, M.Si	Dr. Fachrur Rozi, M.Si
NIP. 19810502 200501 1 004	NIP. 19800527 200801 1 012

Mengetahui,
Ketua Program Studi Matematika

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

OPTIMASI REGRESI LOGISTIK MENGGUNAKAN PENDEKATAN *LEVENBERG-MARQUARDT* DAN *GAUSS SEIDEL* UNTUK PENYELESAIAN KLASIFIKASI BINER

SKRIPSI

Oleh
Arida Nur Baizuriana
NIM. 210601110084

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	: Ari Kusumastuti, M.Pd., M.Si	....
Anggota Penguji 1	: Hisyam Fahmi, M.Kom	....
Anggota Penguji 2	: Dr. Mohammad Jamhuri, M.Si	....
Anggota Penguji 3	: Dr. Fachrur Rozi, M.Si	....

Mengetahui,
Ketua Program Studi Matematika

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

PERNYATAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama	: Arida Nur Baizuriana
NIM	: 210601110084
Program Studi	: Matematika
Fakultas	: Sains dan Teknologi
Judul Skripsi	: Optimasi Regresi Logistik Menggunakan Pendekatan Levenberg Marquardt dan Gauss Seidel untuk Penyelesaian Klasifikasi 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

Arida Nur Baizuriana
NIM. 210601110084

MOTO

“Dan bersabarlah kamu, sesungguhnya janji Allah adalah benar”
(QS. Ar-Ruum: 60)

Nabi bersabda orang iman yang kuat itu lebih baik dan lebih di senangi oleh Allah dari pada orang iman yang lemah dan dari setiap masing-masing orang iman itu ada kebaikannya. Maka engkau semangatlah pada apa-apa yang mendatangkan manfaat bagi engkau dan jangan lemah engkau (jangan malas). Maka ketika suatu perkara telah mengalahkan pada engkau, maka engkau berkatalah ini sudah menjadi qodar Allah dan Allah mengerjakan pada apa-apa yang dikehendaki dan kalian takutlah pada angan-angan. Sesungguhnya angan-angan itu membuka pengamalan setan.
(HR. Ibnu Majah)

“god have perfect timing, never early, never late. It takes a little patience and it takes a lot of faith, but it's a worth the wait”

PERSEMBAHAN

Puji syukur penulis tujukan atas kehadiran Allah SWT yang telah melimpahkan rahmat dan karunianya, sehingga penulis berhasil menempuh pendidikan di Universitas Islam Negeri Maulana Malik Ibrahim Malang serta dapat menyelesaikan penulisan skripsi ini. Penulisan persembahan skripsi ini kepada:

1. Kedua orang tua, Ayah Sarimo, S.E dan Ibu Darmini, dua orang yang sangat berjasa dalam kehidupan penulis, terima kasih telah memberikan kasih sayang, pengorbanan, ketulusan, do'a, usaha, dukungan, dan cinta kasih yang tiada terhingga, serta selalu mengusahakan yang terbaik untuk anak-anaknya. Semoga dengan adanya skripsi ini menjadi langkah awal untuk membuat ayah dan ibu bangga karena telah berhasil menjadikan anak perempuan pertama ini menyandang gelar sarjana yang diharapkan. Besar harapan penulis semoga ayah dan ibu selalu sehat, panjang umur, dan bisa menyaksikan keberhasilan lainnya yang akan penulis raih di masa yang akan datang.
2. Adik-adik tersayang, Almh. Nadia Firna Nur Zuriana, Ardia Zaky Yudistira, Arniza Zahra Asilah, Azhariah Inara Nuha Zahira, Arfirza Irsyad Zahir Firdaus, Aziva Chelya Auristella, yang selalu memberikan dukungan, semangat, menciptakan canda tawa, dan membuat penulis termotivasi untuk bisa terus belajar menjadi sosok kakak yang memberikan pengaruh positif dan menjadi panutan yang baik. Untuk Almh. Nadia Firda Nur Zuriana, kepergianmu di tengah perjuangan ini menjadi duka yang dalam, namun nama mu tetap menjadi semangat dan motivasi bagi penulis.
3. Seluruh keluarga besar baik dari pihak ayah maupun dari pihak ibu, terima kasih selalu memberikan dukungan, motivasi, serta do'a, semua sangat berharga bagi penulis.
4. Seluruh sahabat dan teman penulis, yang selalu membersamai dalam proses penyusunan skripsi ini, selalu membantu, memotivasi dan saling menguatkan. Terima kasih untuk cerita, canda tawa serta air mata, dan semangat hingga akhirnya penulis mampu menuntaskan skripsi ini.

5. *And last but not least,* Arida Nur Baizuriana, ya! Diri saya sendiri. Terima kasih telah bertanggung jawab untuk menyelesaikan apa yang telah dimulai. Terima kasih karena tetap terus berusaha dan tidak menyerah di tengah rasa lelah yang tak hanya fisik tapi juga mental. Anak perempuan pertama yang menjadi harapan pertama, yang takut mengecewakan, yang selalu berperang dengan pikiran sendiri, tetapi akhirnya mampu melawan keraguan dan tekanan yang selalu datang. Setiap air mata, do'a, dan usaha yang dilakukan dalam diam telah menjadi saksi berharganya proses ini. Terima kasih karena selalu berusaha untuk ikhlas atas takdir yang kadang tidak sesuai harapan, selalu meyakinkan diri sendiri bahwa semua punya waktu yang tepat. Terima kasih karena tidak menyerah dan tetap bertahan. Saya bangga dengan diri saya sendiri. Semoga menjadi pribadi yang lebih baik lagi, semoga langkah kebaikan selalu menyertai. Berbahagialah dengan diri sendiri apapun kekurangan dan kelebihanmu, rayakan apapun dalam dirimu. Sehat-sehat diri sendiri, perjalanan untuk membahagiakan dan membanggakan orang tua dan keluarga masih panjang.

KATA PENGANTAR

Segala puji dan syukur dipanjatkan kehadirat Allah SWT atas segala rahmat, hidayah, dan karunia-Nya yang senantiasa melimpah dalam perjalanan penyusunan skripsi yang berjudul “Optimasi Regresi Logistik Menggunakan Pendekatan *Levenberg Marquardt* dan *Gauss Seidel* untuk Penyelesaian Klasifikasi Biner”. Sholawat serta salam yang tiada terhingga kepada Nabi Muhammad SAW, utusan Allah yang menjadi suri tauladan sempurna dalam setiap aspek kehidupan. Semoga keberkahan dan inspirasi dari Rasulullah senantiasa mengiringi langkah-langkah penulis dalam menggapai kesuksesan dan kebermanfaatan.

Skripsi ini disusun sebagai salah satu syarat untuk menyelesaikan studi Strata 1 (S1) Program Studi Matematika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Penyusunan skripsi ini tidak lepas dari banyaknya bantuan, dukungan, arahan, bimbingan dan doa dari berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih 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 sekaligus Dosen Pembimbing II, yang telah memberikan arahan, bimbingan, serta masukan yang sangat bermanfaat, khususnya dalam mengintegrasikan nilai-nilai agama dengan penelitian penulis.
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. Ari Kusumastuti, M.Si., M.Pd, selaku Ketua Penguji, yang dengan penuh perhatian memberikan penilaian , masukan, serta rekomendasi yang sangat berharga demi penyempurnaan skripsi ini.
6. Hisyam Fahmi, M.Kom, selaku Anggota Penguji I, yang telah memberikan saran serta masukan guna memperbaiki dan menyempurnakan skripsi ini.

7. Seluruh dosen Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.
8. Orang tua dan seluruh keluarga yang selalu memberi dukungan moral maupun materi, doa, motivasi, serta kasih sayang yang luar biasa.
9. Teman-teman Program Studi Matematika angkatan 2021 yang selalu mendukung satu sama lain dalam proses penyelesaian skripsi ini.

Penulis menyadari bahwa terdapat banyak kekurangan dalam penulisan skripsi ini. Maka dari itu, penulis berharap diberikan kritik serta saran yang membangun untuk menjadi bahan perbaikan bagi penulis. Penulis juga berharap semoga penelitian ini dapat memberi manfaat kepada penulis maupun pembaca. Mohon maaf atas segala kekurangan pada penulisan skripsi ini.

Malang, 23 Desember 2025

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR SIMBOL

Simbol-simbol yang digunakan pada penelitian ini memiliki penjelasan sebagai berikut:

$\mathbb{R}^{n+1}$	:	Ruang fitur berdimensi $n + 1$
X	:	Matriks fitur
Y	:	Matriks label
y_i	:	Label kelas
$\boldsymbol{x}_i$	:	Vektor yang berisi nilai dari fitur-fitur variabel yang tersedia
m	:	Jumlah sampel (data) dalam dataset
n	:	Jumlah fitur (variabel) untuk setiap sampel
$\boldsymbol{a}$	:	Koefisien bobot
a_0	:	Bobot bias yang bekerja bersama nilai konstan 1
$\sigma(z_i)$	:	Fungsi sigmoid
z_i	:	Kombinasi linier
L	:	Fungsi biaya
∇L	:	Gradien dari fungsi biaya
J	:	Matriks Jacobian
λ	:	Parameter *damping* yang mengatur keseimbangan antara metode *Gauss Newton* dan *gradient descent*
η	:	Vektor perubahan parameter
B	:	Vektor target
μ_j	:	Rata-rata fitur ke-j
σ_j	:	Simpangan baku fitur ke-j

DAFTAR LAMPIRAN

ABSTRAK

Baizuriana, Arida Nur. 2025. **Optimasi Regresi Logistik Menggunakan Pendekatan *Levenberg Marquardt* dan *Gauss Seidel* untuk Penyelesaian Klasifikasi Biner.** Skripsi. Program Studi Matematika Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (1) Dr. Mohammad Jamhuri M.Si. (2) Dr. Fachrur Rozi, M.Si.

Kata kunci: Regresi logistik, *Levenberg–Marquardt*, *Gauss–Seidel*, Optimasi, Klasifikasi Biner.

Penelitian ini dilatarbelakangi oleh kelemahan metode optimasi sederhana seperti *Gradient Descent* (GD) dan Adam, yang umumnya membutuhkan jumlah *epoch* yang banyak, waktu komputasi yang lama, serta konvergensi yang relatif lambat saat digunakan untuk melatih model regresi logistik. Untuk mengatasi keterbatasan tersebut, penelitian ini membahas optimasi regresi logistik untuk klasifikasi biner menggunakan metode *Levenberg–Marquardt* (LM) yang dipadukan dengan *Gauss–Seidel* (GS). Dataset yang digunakan adalah *Pima Indians Diabetes*. Tahapan penelitian meliputi pra-pemrosesan data, pembentukan matriks Jacobian, penyelesaian sistem linear LM dengan GS, serta evaluasi model menggunakan akurasi, presisi, *recall*, dan *F1-score*. Hasil penelitian menunjukkan bahwa LMGS mencapai konvergensi lebih cepat dan akurasi lebih tinggi dibandingkan *Gradient Descent* dan Adam yaitu 0,784. Selain itu, model mampu mencapai akurasi pelatihan $\geq$ 75% dengan jumlah *epoch* dan waktu komputasi yang lebih efisien yaitu memerlukan 10 *epoch* dan waktu 5,76 ms, sehingga metode ini efektif digunakan dalam optimasi regresi logistik.

ABSTRACT

Baizuriana, Arida Nur. 2025. **Optimization of Logistic Regression Using the Levenberg–Marquardt and Gauss–Seidel Approaches for Solving Binary Classification Problems.** Undergraduate Thesis. Department of Mathematics, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Advisors: (1) Dr. Mohammad Jamhuri, M.Si. (2) Dr. Fachrur Rozi, M.Si.

Keywords: Logistic Regression, Levenberg–Marquardt, Gauss–Seidel, Optimization, Binary Classification.

This study is motivated by the limitations of simple optimization methods such as Gradient Descent (GD) and Adam, which generally require a large number of *epoch*s, long computational time, and exhibit relatively slow convergence when applied to logistic regression models. To address these limitations, this research investigates the optimization of logistic regression for binary classification using the Levenberg–Marquardt (LM) method combined with the Gauss–Seidel (GS) approach. The dataset used in this study is the Pima Indians Diabetes dataset. The research stages include data preprocessing, construction of the Jacobian matrix, solving the LM linear system using the Gauss–Seidel method, and model evaluation based on accuracy, precision, recall, and F1-score. The results demonstrate that the LMGS method achieves faster convergence and higher accuracy compared to Gradient Descent and Adam, with an accuracy of 0,784. Furthermore, the model reaches a training accuracy of at least 75% with fewer *epoch*s and more efficient computational time, requiring only 10 *epoch*s and 5,76 ms. These findings indicate that the LMGS method is effective for optimizing logistic regression models.

مستخلص البحث

بايزوريانا، عاريدة نور .٢٠٢٥ . ***تحسين الانحدار اللوجستي باستخدام طريقتي ليفنبرغ–ماركوارت وغاوسسايدل لحل مسائل التصنيف الثنائي***. البحث الجامعي. قسم الرياضيات، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج .المشرف :(١) الدوكتور. محمد جمهوري، الماجستير. (٢) الدوكتور. فخر الرازي، الماجستير.

الكلمات الأساسية: الانحدار اللوجستي، ليفنبرغ–ماركوارت، غاوس–سايدل، التحسين، التصنيف الثنائي.

تنطلق هذه الدراسة من أوجه القصور في طرائق التحسين البسيطة مثل الانحدار للتدرّج وطريقة آدم، والتي تتطلب عادةً عددًا كبيرًا من دورات التدريب، ووقتًا حسابيًا طويلًا، إضافةً إلى بطء نسبي في الوصول إلى حالة التقارب عند تدريب نموذج الانحدار اللوجستي .وللتغلب على هذه القيود، هدفت هذه الدراسة إلى تحسين نموذج الانحدار اللوجستي للتصنيف الثنائي باستخدام طريقة ليفنبرغ–ماركوارت المدمجة مع طريقة غاوس–زايدل .تم تطبيق الدراسة على مجموعة بيانات بيما –للسكري .تشمل مراحل البحث المعالجة المسبقة للبيانات، وبناء مصفوفة جاكوبيان، وحل النظام الخطي في طريقة ليفنبرغ –ماركوارت باستخدام غاوس–زايدل، بالإضافة إلى تقييم أداء النموذج باستخدام مقاييس الدقة، والدقة الإيجابية، والاستدعاء ودرجة إف–واحد .ظَهَرَتْ نتائج الدراسة أن الطريقة المقترحة تحقق تقاربًا أسرع ودقة أعلى مقارنةً بطرائق الانحدار للتدرّج وطريقة آدم، حيث بلغت الدقة ٠٫٧٨٤ .علاوةً على ذلك، تمكّن النموذج من تحقيق دقة تدريب لا تقل عن ٧٥٪ بعدد أقل من دورات التدريب ووقت حسابي أكثر كفاءة، إذ احتاج إلى ١٠ دورات تدريبية وزمن قدره ٥٫٧٦ ميلي ثانية، مما يدل على فعالية الطريقة المقترحة في تحسين نماذج الانحدار اللوجستي

BAB I

PENDAHULUAN

1.1 Latar Belakang

Regresi logistik adalah salah satu metode statistik yang paling sering digunakan dalam kasus klasifikasi biner, di mana tujuan utama adalah memprediksi probabilitas suatu kejadian tergantung pada satu atau lebih variabel independen (Scott dkk., 1991). Regresi logistik merupakan salah satu model klasifikasi yang banyak digunakan karena kesederhanaan, interpretabilitas, serta kemampuannya dalam memodelkan probabilitas kelas biner secara efektif (Hastie dkk., 2009). Regresi logistik digunakan secara luas dalam berbagai bidang, termasuk kedokteran, pemasaran, dan ilmu sosial, untuk memprediksi variabel dependen biner seperti hasil "ya/tidak", "benar/salah", atau "1/0".

Meskipun regresi logistik merupakan salah satu metode yang populer dan banyak digunakan dalam pemodelan klasifikasi, salah satu tantangan utamanya adalah menemukan parameter model yang benar-benar optimal. Parameter yang optimal sangat penting untuk memastikan bahwa model mampu menghasilkan prediksi yang akurat dan andal. Tanpa parameter yang tepat, performa model dapat menurun, yang berdampak pada ketepatan klasifikasi dan keandalan pengambilan keputusan berbasis data. Oleh karena itu, proses pencarian parameter yang optimal menjadi langkah krusial dalam pelatihan model regresi logistik.

Dalam penerapan regresi logistik, terutama pada dataset yang kompleks dan besar, metode optimasi yang lebih baik diperlukan. Hal ini penting untuk meningkatkan akurasi model dan kecepatan komputasi agar model dapat diterapkan dengan efektif di dunia nyata, di mana waktu dan sumber daya komputasi sangat berharga. Namun, dalam beberapa kasus, metode optimasi yang lebih baik diperlukan untuk mempercepat konvergensi atau memperbaiki akurasi model, terutama jika menghadapi masalah non-linearitas atau data yang kompleks. Metode yang diusulkan untuk mempercepat konvergensi atau memperbaiki akurasi model adalah *Levenberg-Marquardt* (LM).

Metode *Levenberg–Marquardt* pertama kali diperkenalkan sebagai pendekatan untuk menyelesaikan permasalahan non-linear least squares dengan mengombinasikan metode *Gauss–Newton* dan *gradient descent* (Levenberg, 1944). *Levenberg-Marquardt* (LM) menggabungkan keuntungan dari metode gradient descent dan *Gauss-Newton*, yang membuatnya sangat efisien dalam mengatasi masalah optimasi non-linear. Algoritma ini efisien dan menunjukkan konvergensi yang stabil, khususnya dalam analisis data runtun waktu (Sumarauw dkk., 2018). LM terkenal karena kecepatan konvergensinya yang lebih cepat dibandingkan metode gradient descent biasa. Dengan demikian, penerapan LM dalam optimasi regresi logistik diharapkan mampu meningkatkan performa model, terutama dalam menangani masalah non-linear yang sering muncul pada data klasifikasi biner.

Metode *Gauss-Seidel (GS)* adalah teknik iteratif yang digunakan untuk menyelesaikan sistem persamaan linear. Metode ini merupakan modifikasi dari metode Jacobi, di mana nilai-nilai solusi yang baru dihitung langsung digunakan dalam perhitungan nilai-nilai berikutnya dalam satu iterasi, bukan menunggu hingga seluruh vektor selesai dihitung (Burden dkk., 2015). Dalam konteks LM, penggunaan metode ini diharapkan dapat mempercepat estimasi parameter model, terutama ketika sistem persamaan non-linear diselesaikan dengan pendekatan iteratif.

Regresi logistik sering diterapkan pada data yang besar. Pada kondisi ini, metode optimasi konvensional mungkin tidak efisien karena membutuhkan waktu komputasi yang lama dan memiliki keterbatasan dalam penanganan kompleksitas data. Oleh karena itu, pendekatan optimasi yang lebih efisien, seperti LM dan GS, menjadi sangat penting.

Berdasarkan penelitian sebelumnya oleh Khan dkk. (2013), masalah klasifikasi kondisi diabetes dikaji dengan fokus pada penggunaan algoritma LM untuk mengurangi kesalahan dalam klasifikasi. Penelitian ini menggunakan model klasifikasi berbasis jaringan saraf tiruan (JST) atau *Artificial Neural Networks* (ANN) dan membandingkan kinerja metode LM dengan metode pelatihan lainnya. Hasil penelitian menunjukkan bahwa metode LM efektif dalam meningkatkan akurasi diagnosis. Namun, penelitian ini masih terbatas pada satu algoritma optimasi tanpa mempertimbangkan alternatif lain yang mungkin lebih efisien. Model klasifikasi dalam penelitian ini menggunakan ANN, bukan regresi logistik. Metode utama yang diterapkan adalah metode LM, yang digunakan untuk melatih ANN dengan fokus pada reduksi kesalahan selama proses pelatihan. Penyelesaian sistem dilakukan dengan pendekatan iteratif, di mana metode LM dipilih untuk mempercepat konvergensi dan meminimalkan kesalahan dalam model ANN. Secara keseluruhan, penelitian ini mengevaluasi efektivitas teknik LM dalam meningkatkan akurasi klasifikasi kondisi diabetes.

Dalam konteks regresi logistik biner, penelitian oleh Indriani (2011) membandingkan metode *Gauss-Newton* dan LM dalam penaksiran parameter, dengan tujuan mengevaluasi efisiensi dan akurasi kedua metode tersebut. Penelitian ini menggunakan data klasifikasi biner untuk memprediksi variabel dependen biner, dengan regresi logistik biner sebagai model klasifikasi utama. Penyelesaian sistem dilakukan melalui pendekatan iterasi non-linear, di mana kedua metode optimasi diterapkan untuk mengestimasi parameter model. Hasil analisis menunjukkan bahwa metode LM memiliki kecepatan konvergensi yang lebih baik dan dianggap lebih stabil dibandingkan metode *Gauss-Newton* dalam beberapa kasus. Meskipun demikian, penelitian ini belum mengeksplorasi aplikasi metode optimasi lain yang mungkin mampu meningkatkan efisiensi estimasi parameter regresi logistik secara lebih lanjut.

Penelitian oleh Mohamad dkk. (2010) membahas tantangan dalam diagnosis kanker payudara menggunakan teknik klasifikasi berbasis *Multi-Layer Perceptron* (MLP), dengan membandingkan dua metode pelatihan, yaitu *Levenberg-Marquardt* (LM) dan *Scaled Conjugate Gradient* (SCG). Model klasifikasi yang digunakan adalah MLP, bukan regresi logistik, dan diterapkan pada data diagnosis kanker payudara untuk mengklasifikasikan kondisi pasien. Proses pelatihan dilakukan secara iteratif, di mana kedua algoritma digunakan untuk mengoptimalkan pembelajaran model MLP. Hasil penelitian menunjukkan bahwa metode LM memberikan akurasi yang lebih tinggi dan waktu pelatihan yang lebih singkat dibandingkan SCG. Namun, pendekatan ini masih terbatas pada penggunaan jaringan saraf tiruan tanpa mengeksplorasi potensi regresi logistik dan teknik optimasi lain yang mungkin menawarkan efisiensi dan fleksibilitas yang lebih baik dalam penyelesaian masalah klasifikasi medis.

Penelitian oleh Koshy dkk. (2024) mengusulkan model jaringan saraf dalam (*deep neural network*/DNN) bernama LMHistNet untuk klasifikasi gambar histopatologi kanker payudara, dengan menerapkan metode LM sebagai metode pelatihan guna meningkatkan akurasi dan kecepatan klasifikasi. Data yang digunakan berupa citra histopatologi kanker payudara, dan model klasifikasi yang digunakan adalah jaringan saraf dalam, bukan regresi logistik. Penyelesaian sistem dilakukan secara iteratif menggunakan metode LM untuk mempercepat konvergensi dan mengoptimalkan akurasi dalam mendeteksi kanker. Meskipun LMHistNet menunjukkan hasil yang menjanjikan dalam konteks klasifikasi berbasis citra, fokus yang kuat pada jaringan saraf dalam membatasi aplikasinya dalam kerangka regresi logistik, sehingga penelitian ini masih memperkuat pemisahan antara pendekatan jaringan saraf dan metode regresi logistik dalam penyelesaian masalah klasifikasi medis.

Penelitian oleh Hagan dkk. (1994) membahas pelatihan jaringan saraf maju (*feedforward neural networks*) menggunakan metode LM, dengan menekankan efisiensi LM dibandingkan metode *backpropagation* dalam mempercepat proses konvergensi dan mengurangi kesalahan selama pelatihan. Model yang digunakan dalam penelitian ini adalah jaringan saraf tiruan, bukan regresi logistik, dan penyelesaian sistem dilakukan melalui pendekatan iterasi non-linear untuk mengoptimalkan parameter jaringan guna meningkatkan akurasi prediksi dan performa model secara keseluruhan. Meskipun menunjukkan hasil yang positif, fokus penelitian ini pada jaringan saraf tiruan membatasi penerapan metode LM dalam konteks regresi logistik, sehingga peluang eksplorasi lebih lanjut terhadap penggunaannya dalam model lain seperti regresi logistik tetap terbuka.

Dalam perspektif Islam, mencari ilmu sangat dianjurkan bagi umatnya untuk mengembangkan metode yang lebih baik dalam menyelesaikan masalah. Metode optimasi dalam regresi logistik, seperti LM dan GS, merupakan bentuk dari upaya mencari solusi yang lebih baik dalam analisis data dan pengambilan keputusan berbasis klasifikasi biner. Dalam QS. Thaha: 114, Allah berfirman:

فَتَعٰلَى اللّٰهُ الْمَلِكُ الْحَقُّۚ وَلَا تَعْجَلْ بِالْقُرْاٰنِ مِنْ قَبْلِ اَنْ يُّقْضٰٓى اِلَيْكَ وَحْيُهٗ ۖ وَقُلْ رَّبِّ زِدْنِيْ عِلْمًا ١١٤

Artinya: *"Mahatinggi Allah, Raja yang sebenar-benarnya. Janganlah engkau (Nabi Muhammad) tergesa-gesa (membaca) Al-Qur'an sebelum selesai pewahyuannya kepadamu dan katakanlah, "Ya Tuhanku, tambahkanlah ilmu kepadaku""* (Kementerian Agama, 2022).

Dalam ayat tersebut dijelaskan bahwa Nabi Muhammad SAW dilarang bersikap tergesa-gesa dalam menerima pengetahuan baru. Allah SWT memerintahkan beliau untuk terus meningkatkan ilmunya. Sikap terburu-buru yang dimaksud adalah kebiasaan Nabi Muhammad SAW mengulangi bacaan malaikat Jibril sebelum Jibril menyelesaikan seluruh wahyu, dengan tujuan agar beliau bisa segera menghafal dan memahami ayat-ayat yang diturunkan. Dari ayat ini, kita dapat mengambil pelajaran bahwa untuk memahami ilmu secara menyeluruh, diperlukan kesungguhan dan upaya yang maksimal dalam belajar (Damanik dkk., 2024).

Menurut riwayat dalam hadits Bukhari, Rasulullah SAW biasa menggerakkan bibirnya saat menerima wahyu. Pada awalnya, beliau merasa kesulitan dalam menghafal ayat-ayat Al-Qur'an. Oleh sebab itu, ketika malaikat Jibril menyampaikan wahyu, beliau segera mengikuti bacaan tersebut dengan gerakan lidah dan bibir karena khawatir tidak dapat mengingatnya. Hal ini dilakukan meskipun Jibril belum menyelesaikan bacaannya. Peristiwa ini terjadi sebelum turunnya Surah Taha. Setelah Allah menegur melalui ayat tersebut, Rasulullah SAW pun menjadi lebih tenang dalam menerima wahyu dan tidak lagi merasa perlu tergesa-gesa dalam menghafalnya (Damanik dkk., 2024).

Hal ini relevan dalam konteks optimasi regresi logistik menggunakan pendekatan LM dan GS untuk penyelesaian klasifikasi biner. Dalam dunia ilmu pengetahuan dan teknologi, khususnya dalam bidang *machine learning*, kesabaran serta pendekatan yang sistematis sangat diperlukan untuk mencapai hasil yang optimal.

Ayat ini mengajarkan pentingnya memahami ilmu dengan teliti dan tidak tergesa-gesa, pendekatan optimasi dalam regresi logistik juga memerlukan proses yang cermat agar algoritma dapat bekerja secara efisien dalam menyelesaikan permasalahan klasifikasi biner. Dengan terus menggali ilmu dan mengembangkan metode yang lebih baik, seperti yang ditekankan dalam doa *Rabbi zidni 'ilma* (“Ya Tuhanku, tambahkanlah ilmu kepadaku”), para peneliti dan praktisi dapat berkontribusi dalam kemajuan teknologi, khususnya dalam bidang *machine learning*.

Oleh karena itu, ayat ini menjadi dasar penulis untuk menggunakan metode LM dan GS. Penelitian tentang Optimasi Regresi Logistik Menggunakan Pendekatan Levenberg-Marquardt dan Gauss-Seidel Untuk Penyelesaian Klasifikasi Biner sangat penting dilakukan karena LM dan GS menawarkan peningkatan dalam hal kecepatan dan efisiensi, yang sangat penting dalam menangani data yang besar dan kompleks.

Metode optimasi ini memiliki potensi untuk meningkatkan akurasi model regresi logistik, yang sangat diperlukan dalam penerapan di sektor-sektor kritis seperti kesehatan, keuangan, dan keamanan. Penelitian ini diharapkan mampu memberikan kontribusi yang signifikan dalam pengembangan metode optimasi pada regresi logistik dan dapat menjadi referensi untuk studi lanjut dalam konteks klasifikasi biner dan pengolahan data kompleks.

1.2 Rumusan Masalah

Berdasarkan latar belakang, maka masalah dalam penelitian ini dapat dirumuskan sebagai berikut:

1. Bagaimana optimalisasi regresi logistik menggunakan pendekatan *Levenberg-Marquardt* dan *Gauss-Seidel* dalam penyelesaian klasifikasi biner?
2. Bagaimana performa metode *Levenberg-Marquardt* dan *Gauss-Seidel di dalam* klasifikasi biner menggunakan regresi logistik?

1.3 Tujuan Penelitian

Berdasarkan latar rumusan masalah, maka tujuan dalam penelitian ini yaitu, sebagai berikut:

1. Menganalisis dan mengimplementasikan optimalisasi regresi logistik menggunakan pendekatan *Levenberg-Marquardt* dan *Gauss-Seidel* dalam penyelesaian klasifikasi biner.
2. Mengevaluasi performa metode *Levenberg-Marquardt* dan *Gauss-Seidel di dalam* klasifikasi biner menggunakan regresi logistik.

1.4 Manfaat Penelitian

Berdasarkan tujuan penelitian, maka manfaat penelitian pada penelitian ini sebagai berikut:

1. Manfaat Teoritis
   a. Menambah wawasan dan kajian ilmiah terkait metode optimasi *Levenberg-Marquardt dan Gauss-Seidel dalam regresi logistik.*
   b. Memberikan kontribusi dalam pengembangan algoritma optimasi yang lebih efisien untuk model klasifikasi biner.
   c. Menyediakan perbandingan kinerja dua metode optimasi yang dapat menjadi referensi bagi penelitian selanjutnya dalam bidang machine learning dan statistik.
2. Manfaat Praktis
   a. Membantu praktisi *science data* dan *machine learning* dalam memilih metode optimasi yang lebih efektif untuk meningkatkan akurasi klasifikasi biner menggunakan regresi logistik.
   b. Memberikan solusi bagi pengembang sistem prediktif yang membutuhkan optimasi parameter regresi logistik dalam berbagai aplikasi, seperti deteksi penyakit, analisis keuangan, dan klasifikasi teks.
   c. Mendukung implementasi metode optimasi dalam berbagai bidang yang membutuhkan analisis klasifikasi biner, seperti kecerdasan buatan, pengolahan citra, dan analisis data bisnis.

1.5 Batasan Masalah

Agar penelitian ini lebih terarah dan fokus, beberapa batasan masalah yang perlu diperhatikan adalah sebagai berikut:

1. Penelitian ini dibatasi pada pengolahan dan analisis data *Pima Indian Diabetes* untuk kasus klasifikasi biner.
2. Penelitian ini hanya difokuskan pada tahapan implementasi dan evaluasi kinerja hasil optimasi parameter.

1.6 Definisi Istilah

1. Klasifikasi Biner

   Klasifikasi biner adalah salah satu tugas utama dalam pembelajaran mesin yang bertujuan memprediksi salah satu dari dua kelas target (Han dkk., 2022).

2. Regresi Logistik

   Regresi logistik adalah salah satu metode statistik yang paling sering digunakan dalam kasus klasifikasi biner, di mana tujuan utama adalah memprediksi probabilitas suatu kejadian tergantung pada satu atau lebih variabel independent (Scott dkk., 1991).

3. Metode Levenberg-Marquardt

   Metode Levenberg-Marquardt (LM) menggabungkan keuntungan dari metode gradient descent dan Gauss-Newton, yang membuatnya sangat efisien dalam mengatasi masalah optimasi non-linear. Algoritma ini efisien dan menunjukkan konvergensi yang stabil, khususnya dalam analisis data runtun waktu (Sumarauw dkk., 2018).

4. Metode Gauss-Seidel

Metode *Gauss-Seidel* adalah teknik iteratif yang digunakan untuk menyelesaikan sistem persamaan linear. Metode ini merupakan modifikasi dari metode Jacobi, di mana nilai-nilai solusi yang baru dihitung langsung digunakan dalam perhitungan nilai-nilai berikutnya dalam satu iterasi, bukan menunggu hingga seluruh vektor selesai dihitung (Burden dkk., 2015).

BAB II

KAJIAN TEORI

2.1 Teori Pendukung

2.1.1 Klasifikasi Biner

Klasifikasi biner adalah salah satu tugas utama dalam pembelajaran mesin yang bertujuan memprediksi salah satu dari dua kelas target (Han dkk., 2022). Model klasifikasi biner digunakan secara luas dalam berbagai bidang, seperti deteksi spam, prediksi penyakit, dan analisis sentimen.

Dalam tugas klasifikasi, regresi logistik adalah salah satu algoritma yang paling sering digunakan karena kemampuannya menangani keluaran biner secara langsung. Selain itu, regresi logistik memiliki interpretabilitas yang tinggi, yang penting dalam aplikasi-aplikasi yang memerlukan transparansi model (Scott dkk., 1991).

Untuk menilai kinerja model klasifikasi biner, berbagai metrik evaluasi digunakan, seperti akurasi, presisi, *recall*, *F1-score*, dan AUC-ROC. Akurasi dihitung dengan membandingkan jumlah prediksi benar terhadap total prediksi, sementara precision mengukur seberapa banyak prediksi positif yang benar. *Recall*, atau sensitivitas, menunjukkan seberapa baik model dalam mendeteksi sampel positif, sedangkan *F1-score* merupakan harmoni antara presisi dan *recall*.

Sementara itu, AUC-ROC digunakan untuk mengukur kemampuan model dalam membedakan antara dua kelas. Dengan metode yang beragam dan metrik evaluasi yang tepat, klasifikasi biner menjadi teknik yang banyak diterapkan dalam berbagai bidang, seperti deteksi *spam*, diagnosis medis, dan analisis sentimen.

2.1.2 Regresi Logistik

Regresi logistik adalah metode statistik yang digunakan untuk memodelkan hubungan antara sekumpulan variabel independen $x_1, x_2, \cdots, x_n$, dengan variabel dependen y. Tujuan utama regresi logistik adalah untuk memprediksi probabilitas bahwa suatu kejadian terjadi berdasarkan variabel independen yang diberikan. Teknik ini sering digunakan dalam klasifikasi biner, di mana variabel dependen y hanya dapat mengambil dua nilai, yaitu 0 dan 1.

Dalam regresi logistik, kita diberikan dataset pelatihan D, yang terdiri dari n sampel dengan fitur $\boldsymbol{x}_i \in \mathbb{R}^{n+1}$ dan label kelas $y_i \in \{0, 1\}$. Menurut Jamhuri dkk. (2022) untuk mempermudah perhitungan, kita menambahkan sebuah fitur tambahan $x_{i,0} = 1$, sehingga setiap titik data direpresentasikan sebagai vektor yang diperbesar:

$$\boldsymbol{x}_i = (1, x_{i,1}, x_{i,2}, \cdots, x_{i,n}) \in \mathbb{R}^{n+1}$$

dengan vektor parameter regresi logistik dinyatakan sebagai:

$$\boldsymbol{a} = (a_0, a_1, \dots, a_n)$$

Fungsi logit dalam regresi logistik dihitung sebagai kombinasi linear dari fitur dengan bobot parameter:

$$z_i = a_0 + a_1 x_{i,1} + \cdots + a_n x_{i,n} = \boldsymbol{a} \cdot \boldsymbol{x}_i \tag{2.1}$$

Selanjutnya, fungsi aktivasi sigmoid digunakan untuk mengonversi z_i menjadi probabilitas:

$$\sigma(z_i) = \frac{1}{1 + \exp\,(-z_i)} \tag{2.2}$$

Fungsi biaya dalam regresi logistik untuk klasifikasi biner dirumuskan sebagai berikut:

$$L = -\frac{1}{m}\sum_{i=1}^{m}[y_i \log\big(\sigma(z_i)\big) + (1 - y_i)\ \log\,(1 - \sigma(z_i))] \tag{2.3}$$

yang bertujuan untuk diminimalkan agar model dapat memprediksi probabilitas dengan lebih akurat.

Salah satu metode yang sering digunakan untuk meminimalkan fungsi biaya dalam regresi logistik adalah *penurunan gradien* (*gradient descent*). Dalam metode ini, parameter a_j diperbarui secara iteratif berdasarkan gradien fungsi biaya:

$$a_j^{(k+1)} = a_j^{(k)} - \alpha\,\frac{\partial L(\boldsymbol{a})}{\partial a_j} \tag{2.4}$$

dengan α sebagai laju pembelajaran (learning rate). Gradien dari fungsi biaya $J(\boldsymbol{a})$ terhadap a_j diberikan oleh:

$$\frac{\partial L(\boldsymbol{a})}{\partial a_j} = -\sum_{i=1}^{m}[\,y_i\,\frac{\partial \log\big(\sigma(z_i)\big)}{\partial a_j} + (1 - y_i)\ \frac{\partial \log\big(1 - \sigma(z_i)\big)}{\partial a_j}] \tag{2.5}$$

Untuk $j = 0, 1, \ldots, n$.

Untuk menyelesaikan turunan Persamaan (2.5), kita gunakan:

$$\frac{\partial \log\big(\sigma(z_i)\big)}{\partial a_j} = \frac{1}{\sigma(z_i)} \cdot \sigma(z_i)\big(1 - \sigma(z_i)\big) \cdot x_{i,j} = \big(1 - \sigma(z_i)\big)x_{i,j} \tag{2.6}$$

$$\frac{\partial \log\big(1 - \sigma(z_i)\big)}{\partial a_j} = \frac{1}{1 - \sigma(z_i)} \cdot (-\sigma(z_i)\big(1 - \sigma(z_i)\big)) \cdot x_{i,j} = -\sigma(z_i)x_{i,j} \tag{2.7}$$

Dengan mensubstitusi hasil turunan Persamaan (2.6) dan Persamaan (2.7) ke dalam Persamaan (2.5), diperoleh:

$$\frac{\partial L(\boldsymbol{a})}{\partial a_j} = -\sum_{i=1}^{m} (y_i - \sigma(z_i))x_{i,j}, \qquad j = 0, 1, \dots, n. \tag{2.8}$$

Menggunakan aturan pembaruan dalam metode penurunan gradien pada Persamaan (2.4), diperoleh:

$$a_j^{(k+1)} = a_j^{(k)} - \alpha \sum_{i=1}^{m} (y_i - \sigma(z_i))x_{i,j}, \qquad j = 0, 1, \dots, n. \tag{2.9}$$

Regresi logistik banyak digunakan dalam berbagai bidang, termasuk kedokteran, pemasaran, dan ilmu sosial, untuk memprediksi variabel biner seperti keputusan "ya/tidak", "benar/salah", atau "positif/negatif". Model ini sangat populer karena kesederhanaannya dan kemampuannya untuk memberikan interpretasi probabilistik dari hasil klasifikasi.

Metode ini memperbarui parameter regresi logistik dengan mengarahkan perubahan parameter ke arah yang mengurangi fungsi biaya. Dengan iterasi yang cukup, metode ini akan menemukan nilai parameter yang optimal untuk memprediksi probabilitas kelas dengan baik.

2.1.3 Metode *Levenberg-Marquardt*

Levenberg-*Marquardt* (LM) menggabungkan keuntungan dari metode *gradient descent* dan *Gauss-Newton*, yang membuatnya sangat efisien dalam mengatasi masalah optimasi non-linear. Algoritma ini efisien dan menunjukkan konvergensi yang stabil, khususnya dalam analisis data runtun waktu (Sumarauw dkk., 2018).

Menurut Jamhuri dkk. (2022) dalam konteks regresi logistik, pembaruan parameter $\boldsymbol{a}$ dengan cara berikut:

$$\boldsymbol{a}^{(k+1)} = \boldsymbol{a}^{(k)} + \Delta\boldsymbol{a} \tag{2.10}$$

Untuk meminimalkan fungsi biaya dengan pendekatan *Gauss-Newton*, diperlukan penyusunan sistem persamaan linear yang kemudian diselesaikan terhadap vektor $\boldsymbol{a}$ pada setiap iterasinya. Fungsi *loss* untuk dapat didefinisikan sebagai berikut:

$$L_i = -\big(y_i \log\big(\sigma(z_i)\big) + (1 - y_i)\log\big(1 - \sigma(z_i)\big)\big) \tag{2.11}$$

Selanjutnya, lakukan linearisasi terhadap $\log\big(\sigma(z_i)\big)$ dan $\log\big(1 - \sigma(z_i)\big)$ dalam G_i dengan menggunakan perluasan deret Taylor orde pertama. Misalkan $p(\boldsymbol{a}) = \log\big(\sigma(z_i)\big)$ dan $q(\boldsymbol{a}) = \log\big(1 - \sigma(z_i)\big)$, maka bentuk perluasan Taylor dari fungsi-fungsi ini adalah:

$$\begin{aligned} p(\boldsymbol{a} + \Delta\boldsymbol{a}) &\approx p(\boldsymbol{a}) + \nabla\, p(\boldsymbol{a}) \cdot \Delta\boldsymbol{a} \\ q(\boldsymbol{a} + \Delta\boldsymbol{a}) &\approx q(\boldsymbol{a}) + \nabla\, q(\boldsymbol{a}) \cdot \Delta\boldsymbol{a} \end{aligned} \tag{2.12}$$

di mana gradien p dan q didefinisikan sebagai $\nabla\, p(\boldsymbol{a}) = \left(\frac{\partial p}{\partial a_0}, \frac{\partial p}{\partial a_1}, \dots, \frac{\partial p}{\partial a_n}\right)$, dan $\nabla\, q(\boldsymbol{a}) = \left(\frac{\partial q}{\partial a_0}, \frac{\partial q}{\partial a_1}, \dots, \frac{\partial q}{\partial a_n}\right)$ sedangkan $\Delta\boldsymbol{a} = (\Delta a_0, \Delta a_1, \dots, \Delta a_n)$. Sebagaimana yang didefinisikan dalam Persamaan (2.12), turunan p dan q terhadap a_j adalah

$$\begin{aligned} &\frac{\partial}{\partial a_j} \log \sigma(z_i) = \big(1 - \sigma(z_i)\big)x_{i,j}, \\ &\frac{\partial}{\partial a_j} \log\big(1 - \sigma(z_i)\big) = -\sigma(z_i)x_{i,j}, \quad \forall j = 0, 1, \dots, n \end{aligned} \tag{2.13}$$

Dalam rangka melinearisasi fungsi *loss* L_i, digunakan fungsi-fungsi aproksimasi p dan q beserta turunan-turunannya, sehingga

$$L_i \approx -y_i(p(\boldsymbol{a}) + \nabla\, p(\boldsymbol{a}) \cdot \Delta\boldsymbol{a}) - (1 - y_i)(\, q(\boldsymbol{a}) + \nabla\, q(\boldsymbol{a}) \cdot \Delta\boldsymbol{a})$$

Kemudian dapat ditulis sebagai

$$y_i(\nabla p(\boldsymbol{a}) \cdot \Delta\boldsymbol{a}) + (1 - y_i)(\nabla q(\boldsymbol{a}) \cdot \Delta\boldsymbol{a}) = -y_i p(\boldsymbol{a}) - (1 - y_i)\, q(\boldsymbol{a})$$

Di mana ruas kanan dari persamaan sudah diketahui dan ruas kirinya belum diketahui. Dengan mengumpulkan $\Delta\boldsymbol{a}$ pada setiap suku di ruas kiri, kita memperoleh

$$\left(y_i \nabla p(\boldsymbol{a}) - y_i \nabla q(\boldsymbol{a}) + \nabla q(\boldsymbol{a})\right) \cdot \Delta\boldsymbol{a} = -(y_i p(\boldsymbol{a}) + (1 - y_i) q(\boldsymbol{a}) \quad (2.14)$$

Persamaan $\nabla p(\boldsymbol{a}) - \nabla q(\boldsymbol{a}) = \left(1, x_{i,1}, x_{i,2}, \cdot\cdot\cdot, x_{i,n}\right) = \boldsymbol{x}_i$ menunjukkan bahwa selisih antara gradien dari dua fungsi, yaitu $p(\boldsymbol{a})$ dan $q(\boldsymbol{a})$, dapat direpresentasikan sebagai vektor fitur $\boldsymbol{x}_i$. Vektor ini terdiri dari fitur-fitur data sampel ke-i, ditambah dengan komponen bias (nilai 1 di awal), sehingga membentuk vektor berdimensi $n + 1$. Kemudian, dengan mengalikan persamaan tersebut dengan label target y_i, diperoleh

$$\mathrm{y_i}\left(\nabla p(\boldsymbol{a}) - \nabla q(\boldsymbol{a})\right) = y_i \boldsymbol{x}_i \quad (2.15)$$

Dari Persamaan (2.15), kita dapat menyederhanakan ruas kanan yaitu sebagai berikut

$$\mathrm{y_i}\left(\nabla p(\boldsymbol{a}) - \nabla q(\boldsymbol{a})\right) + \nabla q(\boldsymbol{a}) = y_i \boldsymbol{x}_i - \sigma(z_i)\boldsymbol{x}_i = \left(y_i - \sigma(z_i)\right)\boldsymbol{x}_i$$

Kemudian terapkan ke dalam persamaan yang telah dilinearisasi, kita peroleh

$$\left(y_i - \sigma(z_i)\right)\boldsymbol{x}_i \cdot \Delta\boldsymbol{a} = -\left(y_i \log\left(\sigma(z_i)\right) + (1 - y_i) \log\left(1 - \sigma(z_i)\right)\right)$$

Di mana $L_i = -\left(y_i \log\left(\sigma(z_i)\right) + (1 - y_i) \log\left(1 - \sigma(z_i)\right)\right)$, maka:

$$\left(y_i - \sigma(z_i)\right)\boldsymbol{x}_i \cdot \Delta\boldsymbol{a} = L_i \quad (2.16)$$

Dengan mensubstitusikan i ke dalam persamaan (2.16) untuk $i = 1, 2, \dots, m$, kita memperoleh sekumpulan sistem persamaan linier sebanyak m

$$\left(y_1 - \sigma(z_1)\right)\boldsymbol{x}_1 \cdot \Delta\boldsymbol{a} = L_1$$

$$\left(y_2 - \sigma(z_2)\right)\boldsymbol{x}_2 \cdot \Delta\boldsymbol{a} = L_2$$

$$\vdots$$

$$\left(y_m - \sigma(z_m)\right)\boldsymbol{x}_m \cdot \Delta\boldsymbol{a} = L_m$$

Sistem tersebut dapat dituliskan dalam bentuk persamaan matriks

$$\begin{bmatrix} \left(y_1 - \sigma(z_1)\right) & \left(y_1 - \sigma(z_1)\right)x_{1,1} & \cdots & \left(y_1 - \sigma(z_1)\right)x_{1,n} \\ \left(y_2 - \sigma(z_2)\right) & \left(y_2 - \sigma(z_2)\right)x_{2,1} & \cdots & \left(y_2 - \sigma(z_2)\right)x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \left(y_m - \sigma(z_m)\right) & \left(y_m - \sigma(z_m)\right)x_{m,1} & \cdots & \left(y_m - \sigma(z_m)\right)x_{m,n} \end{bmatrix} \begin{bmatrix} \Delta a_0 \\ \Delta a_1 \\ \vdots \\ \Delta a_n \end{bmatrix} = \begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_m \end{bmatrix}$$

dengan:

Matriks jacobian linierisasi:

$$\boldsymbol{J} = \begin{pmatrix} \left(y_1 - \sigma(z_1)\right) & \left(y_1 - \sigma(z_1)\right)x_{1,1} & \cdots & \left(y_1 - \sigma(z_1)\right)x_{1,n} \\ \left(y_2 - \sigma(z_2)\right) & \left(y_2 - \sigma(z_2)\right)x_{2,1} & \cdots & \left(y_2 - \sigma(z_2)\right)x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \left(y_m - \sigma(z_m)\right) & \left(y_m - \sigma(z_m)\right)x_{m,1} & \cdots & \left(y_m - \sigma(z_m)\right)x_{m,n} \end{pmatrix}$$

Vektor perubahan parameter:

$$\boldsymbol{\eta} = \begin{pmatrix} \Delta a_0 \\ \Delta a_1 \\ \vdots \\ \Delta a_n \end{pmatrix}$$

Vektor fungsi biaya per observasi:

$$\boldsymbol{B} = \begin{pmatrix} L_1 \\ L_2 \\ \vdots \\ L_m \end{pmatrix}$$

Jika $m = n$, maka sistem memiliki satu solusi unik. Namun, jika $m < n$, sistem akan memiliki banyak solusi. Sementara itu, untuk $m > n$, sistem tidak memiliki solusi eksak dan akan mengarah pada penyelesaian dengan pendekatan kuadrat terkecil. Dari pendekatan kuadrat terkecil inilah muncul sistem normal:

$$\boldsymbol{J}^T\boldsymbol{J} = \boldsymbol{J}^T\boldsymbol{B} \tag{2.17}$$

Dengan $\boldsymbol{\eta} = \begin{pmatrix} \Delta a_0 \\ \Delta a_1 \\ \vdots \\ \Delta a_n \end{pmatrix}$, yang merupakan dasar dari metode *Gauss–Newton*. Namun, metode ini bisa tidak stabil jika matriks $\boldsymbol{J}^T\boldsymbol{J}$ bersifat hampir singular. Untuk mengatasi hal ini, *metode LM* memperkenalkan konsep *regularisasi* dengan menambahkan parameter λ pada diagonal matriks Hessian aproksimasi, menjadi:

$$(\boldsymbol{J}^T\boldsymbol{J} + \lambda\boldsymbol{I})\boldsymbol{\eta} = \boldsymbol{J}^T\boldsymbol{B} \tag{2.18}$$

Keterangan:

$\boldsymbol{J}$: Matriks Jacobian.

λ : Parameter *damping* yang mengatur keseimbangan antara metode *Gauss Newton* dan gradient descent.

$\boldsymbol{\eta}$: Vektor perubahan parameter.

$\boldsymbol{B}$: Vektor target.

2.1.4 Metode *Gauss Seidel*

Levenberg-Marquardt menghasilkan *sistem linear* sebagai hasil dari proses optimasi:

$$(\boldsymbol{J}^T\boldsymbol{J} + \lambda\boldsymbol{I})\boldsymbol{\eta} = \boldsymbol{J}^T\boldsymbol{B}$$

Misalkan kita ubah bentuk ini menjadi:

$$\boldsymbol{M\eta} = \boldsymbol{c} \tag{2.19}$$

dengan:

$\boldsymbol{M} = \boldsymbol{J}^T\boldsymbol{J} + \lambda\boldsymbol{I}$: Matriks $(n+1) \times (n+1)$

$\boldsymbol{c} = \boldsymbol{J}^T\boldsymbol{B}$: Vektor $(n+1) \times 1$

$\boldsymbol{\eta}$: Vektor parameter yang ingin dicari.

Matriks M dapat dinyatakan sebagai:

$$M = L + D + U$$

$$(L + D + U)\eta = c$$

$$(L + D)\eta_{k+1} = -U\eta_k + c$$

$$\eta_{k+1} = D^{-1}(c - U\eta_k - L\eta_{k+1}) \text{ untuk } k = 0, 1, 2, \ldots$$

Metode *GS* digunakan untuk menyelesaikan sistem linear $\boldsymbol{M\eta} = \boldsymbol{c}$ secara *iteratif.*

Langkah pertama yaitu menuliskan

$$\sum_{i=0}^{n} \boldsymbol{M}_{j,i}\, \Delta a_i = c_j \qquad \text{untuk } j = 0, 1, \ldots, n \tag{2.20}$$

Kemudian, kita pecah komponen yang diketahui dan tidak diketahui untuk pembaruan secara iteratif: (Burden dkk., 2015)

$$\Delta a_j^{(k+1)} = \frac{1}{\boldsymbol{M}_{j,j}} \left(c_j - \sum_{i=0}^{j-1} \boldsymbol{M}_{j,i} \Delta a_i^{(k+1)} - \sum_{i=j+1}^{n} \boldsymbol{M}_{j,i} \Delta a_i^{(k)} \right) \tag{2.21}$$

Untuk memantau proses konvergensi, iterasi dihentikan ketika akurasi pelatihan telah mencapai tingkat yang ditentukan yaitu apabila sudah mencapai 0,75 (75%). Proses iterasi ini digunakan sebagai kriteria tercapainya konvergensi model.

2.2 Kajian Integrasi Optimasi Regresi Logistik dalam Islam

Dalam Islam, ilmu pengetahuan merupakan sarana untuk mencapai kesejahteraan manusia dan menegakkan keadilan. Salah satu bentuk pengembangan ilmu di era modern adalah optimasi regresi logistik menggunakan pendekatan LM dan GS, yang bertujuan untuk meningkatkan akurasi dalam klasifikasi biner.

Konsep optimasi regresi logistik ini sejalan dengan prinsip-prinsip Islam salah satunya yaitu bersungguh-sungguh dan giat dalam melakukan suatu pekerjaan.

Al-Qur’an memberikan panduan agar manusia senantiasa berupaya mencari solusi terbaik dalam berbagai aspek kehidupan, termasuk dalam ranah sains dan teknologi. Oleh karena itu, setiap pengembangan ilmu harus dilandasi oleh nilai-nilai yang bersumber dari Al-Qur’an dan Hadits. Hal ini mencakup pula bidang-bidang seperti matematika dan statistik, yang memiliki peran penting dalam mewujudkan keadilan dan kesejahteraan bersama.

Bersyukur atas kesempatan untuk bekerja dan berusaha merupakan sikap yang seharusnya dimiliki setiap individu. Kemampuan untuk menjalankan aktivitas kerja dan memperoleh penghasilan bukanlah hal yang dapat dimiliki oleh semua orang, melainkan merupakan nikmat dan anugerah dari Allah SWT yang patut disyukuri. Oleh karena itu, dalam menjalankan pekerjaan, sudah semestinya dilakukan dengan sebaik-baiknya, disertai kesungguhan, tanggung jawab, dan integritas. Bekerja dengan sungguh-sungguh tidak hanya mencerminkan etos kerja yang baik, tetapi juga merupakan bentuk ibadah apabila dilakukan dengan niat yang benar dan sesuai dengan nilai-nilai kebaikan. Sikap ini menjadi landasan penting dalam berbagai aktivitas, termasuk dalam kegiatan penelitian ilmiah, di mana ketekunan, ketelitian, dan dedikasi menjadi kunci utama dalam menghasilkan karya yang bermanfaat dan berkualitas.

وَسَلَّمَ :إِنّ اللَّهَ تَعَالى يُحِبّ إِذَا عَمِلَ أَحَدُكُمْ عَمَلاً أَنْ يُتْقِنَهُ (رواه الطبرني والبيهقي)

Artinya: “Dari Aisyah r.a., sesungguhnya Rasulullah s.a.w. bersabda: “Sesungguhnya Allah mencintai seseorang yang apabila bekerja, mengerjakannya secara itqan (professional)”.” (HR. Thabrani) (pa-sungairaya.go.id, 2024).

Menurut Buya H. Muhammad Alfis Chaniago, apabila seseorang diberikan amanah dalam bentuk pekerjaan, maka sudah sepatutnya ia melaksanakannya dengan sebaik mungkin. Pekerjaan tersebut hendaknya dilakukan dengan sepenuh hati, penuh tanggung jawab, dan tidak mengecewakan pihak yang telah memberikan kepercayaan. Sikap ini mencerminkan integritas dan kesungguhan dalam menjalankan tugas yang diamanahkan (pa-sungairaya.go.id, 2024).

Dalam konteks penelitian ilmiah, nilai kesungguhan ini sangat penting karena proses penelitian membutuhkan ketekunan, kehati-hatian, dan tanggung jawab agar menghasilkan hasil yang bermanfaat dan berkualitas. Penelitian tentang optimasi regresi logistik menggunakan pendekatan LM dan GS untuk menyelesaikan masalah klasifikasi biner merupakan salah satu bentuk penerapan nilai tersebut.

Proses optimasi memerlukan ketelitian, keuletan, dan dedikasi tinggi karena bertujuan untuk memperoleh solusi yang paling akurat dan efisien. Pemilihan metode LM, yang unggul dalam menangani permasalahan non-linear, serta metode GS, yang efisien dalam menyelesaikan sistem persamaan linear, merupakan wujud dari kesungguhan peneliti dalam memilih pendekatan terbaik demi meningkatkan kinerja model klasifikasi. Dengan demikian, integrasi metode-metode ini tidak hanya mencerminkan kemampuan teknis, tetapi juga mencerminkan sikap itqan dan kesungguhan dalam bekerja sebagaimana dianjurkan dalam ajaran Islam.

BAB III

METODE PENELITIAN

3.1 Jenis Penelitian

Penelitian ini merupakan penelitian kuantitatif eksperimental yang bertujuan mengoptimalkan model klasifikasi biner menggunakan regresi logistik. Pendekatan optimasi yang diterapkan melibatkan dua metode iteratif, yaitu LM dan GS, yang kemudian dibandingkan dengan metode optimasi lainnya seperti *gradient descent* dan Adam. Penelitian ini akan menguji kinerja metode-metode ini dalam memecahkan masalah klasifikasi biner pada dataset *Pima Indian Diabetes*.

3.2 Data dan Sumber Data

Dataset yang digunakan dalam penelitian ini diperoleh dari *Kaggle* yaitu *Pima Indian Diabetes* Dataset: Dataset ini mengandung data medis terkait diabetes dengan total 768 data dan 8 fitur numerik. Klasifikasi biner dilakukan untuk memprediksi apakah seorang pasien menderita diabetes atau tidak. Pemilihan dataset ini didasarkan pada beberapa pertimbangan. Dataset ini banyak digunakan dalam penelitian klasifikasi biner, sehingga memudahkan proses pembandingan hasil antara berbagai metode optimasi seperti *Gradient Descent*, Adam, *Levenberg–Marquardt* dan *Gauss Seidel*. Selain itu, dataset ini merupakan data nyata yang sering digunakan sebagai *benchmark* dalam penelitian *machine learning* untuk deteksi dini penyakit diabetes, sehingga hasil yang diperoleh dari penelitian ini dapat dibandingkan secara objektif dengan hasil penelitian sebelumnya.

3.3 Tahapan Penelitian

Tahapan-tahapan yang akan dilakukan dalam menyelesaikan penelitian ini, antara lain:

3.3.1 Persiapan Data

Sebelum digunakan dalam analisis, data harus dipersiapkan dan dibersihkan melalui langkah-langkah berikut:

1. Standardisasi Fitur: Setiap fitur akan distandardisasi menggunakan metode *Z-Score Standardization* agar memiliki rata-rata 0 dan simpangan baku 1, menggunakan rumus:

$$x'_{i,j} = \frac{x_{i,j} - \mu_j}{\sigma_j} \tag{3.1}$$

dengan:

- $x'_{i,j}$: nilai setelah standardisasi
- $x_{i,j}$: nilai fitur ke-j pada data ke-i
- μ_j : rata-rata fitur ke-j
- σ_j : simpangan baku fitur ke-j

Adapun rumus rata-rata dan simpangan baku untuk setiap fitur yaitu sebagai berikut.

Rumus Rata-rata

$$\mu_j = \frac{\sum_{i=1}^{n} x_{i,j}}{n}$$

Rumus Simpangan Baku

$$\sigma_j = \sqrt{\frac{\sum_{i=1}^{n} \left(x_{i,j} - \mu_j\right)^2}{n-1}}$$

Menurut (Bishop, 2006), penyamaan skala fitur merupakan bagian dari tahap awal prapemrosesan data yang bertujuan untuk menjaga kestabilan proses optimasi dan perhitungan numerik dalam pembelajaran model. Pada penelitian ini, penyamaan skala fitur dilakukan menggunakan metode standardisasi, sehingga membantu proses optimasi mencapai konvergensi yang lebih cepat.

2. Split Data: Dataset yang telah distandardisasi dibagi menjadi beberapa bagian, yaitu data latih sebesar 70% , data validasi sebesar 15%, dan data uji sebesar 15%.

3.3.2 Implementasi Regresi Logistik

1. Menghitung nilai z_i

   Nilai z_i diperoleh dari kombinasi linear antara bobot dan fitur sesuai Persamaan (2.1)

2. Menghitung fungsi aktivasi sigmoid

   Nilai probabilitas dihitung dengan rumus sesuai dengan Persamaan (2.2)

3. Menghitung fungsi *loss*

   Menghitung error untuk setiap sampel data menggunakan rumus pada Persamaan (2.11)

3.3.3 Implementasi Metode *Levenberg-Marquardt*

Metode *Levenberg–Marquardt (LM)* digunakan untuk mengoptimalkan parameter regresi logistik melalui kombinasi pendekatan *Gradient Descent* dan *Gauss Newton*. Tahapan implementasinya sebagai berikut:

1. Menghitung ∇L_i

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

$$\frac{\partial L_i}{\partial a_j} = (y_i - \sigma(z_i))x_{i,j} \tag{3.2}$$

2. Membangun matriks Jacobian
3. Menghitung matriks *Hessian Aproximation*

$$\boldsymbol{M} = \boldsymbol{J}^T\boldsymbol{J} + \lambda\boldsymbol{I}$$

3.3.4 Implementasi Metode *Gauss-Seidel*

Metode *Gauss–Seidel (GS)* digunakan untuk menyelesaikan sistem persamaan linear yang muncul pada proses optimasi regresi logistik hasil dari metode *Levenberg–Marquardt.* Pendekatan ini bersifat iteratif dan memperbarui setiap parameter secara berurutan hingga mencapai konvergensi. Langkah-langkah implementasinya adalah sebagai berikut:

1. Menyusun Sistem Persamaan Linear
2. Proses iterasi

   Proses iterasi pada metode *Gauss Seidel* dilakukan untuk menghitung nilai baru setiap variabel secara bertahap dengan memanfaatkan hasil perhitungan variabel-variabel sebelumnya sesuai dengan Persamaan (2.21).

3. Kriteria konvergensi

Iterasi dihentikan ketika akurasi pelatihan telah mencapai tingkat yang ditentukan yaitu apabila sudah mencapai 0,75 (75%). Proses iterasi ini digunakan sebagai kriteria tercapainya konvergensi model.

3.3.5 Evaluasi Model

Evaluasi model dilakukan menggunakan beberapa metrik performa:

1. Akurasi: Persentase prediksi yang benar dari total data.

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

2. Presisi: Rasio *true positive* terhadap total prediksi positif.

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

3. *Recall*: Rasio *true positive* terhadap total kasus sebenarnya yang positif.

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

4. *F1-Score*: Harmonik antara presisi dan *recall* untuk menyeimbangkan keduanya.

$$F1 = 2 \times \frac{presisi \times recall}{presisi + recall}$$

Adapun definisi dari masing-masing istilah evaluasi model adalah sebagai berikut:

TP (True Positive) :Kasus di mana model memprediksi *positif* dan kenyataannya juga *positif.*

TN (True Negative) :Kasus di mana model memprediksi *negatif* dan kenyataannya juga *negatif.*

FP (False Positive) :Kasus di mana model memprediksi *positif* namun kenyataannya *negatif*.

FN (False Negative) :Kasus di mana model memprediksi *negatif* namun kenyataannya *positif*.

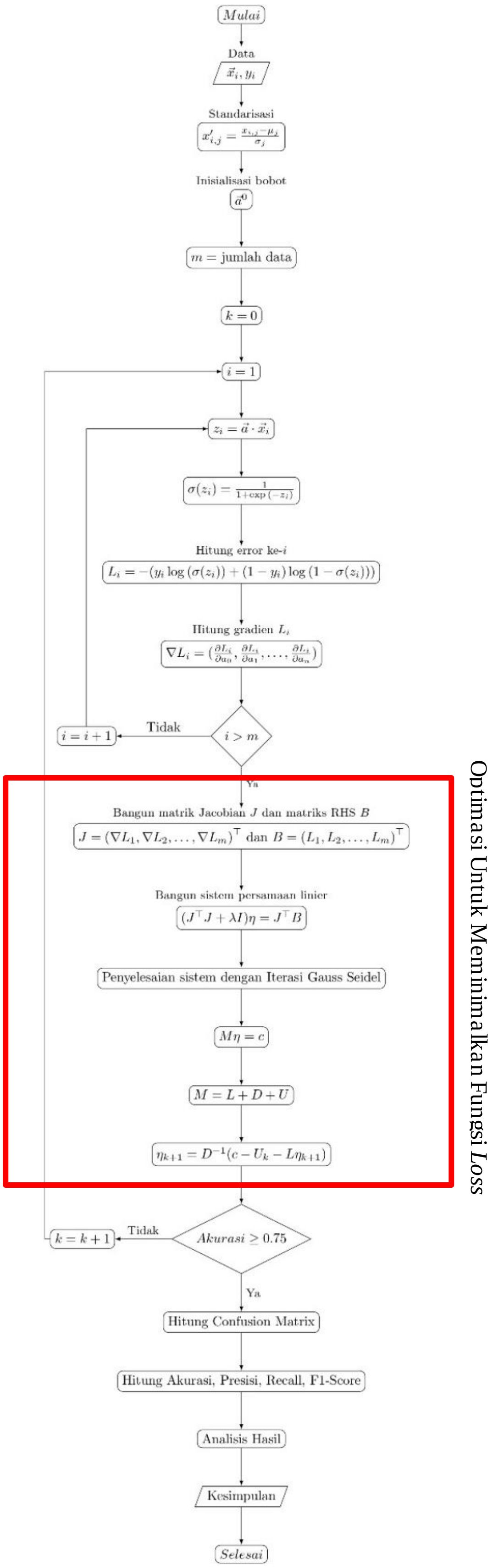
Selain itu, model akan dibandingkan dengan benchmark dari optimasi tradisional, seperti:

1. *Gradient Descent*: Algoritma optimasi paling dasar, dengan pembaruan parameter berdasarkan turunan pertama dari fungsi biaya.
2. Adam (*Adaptive Moment Estimation*): Algoritma optimasi yang menggunakan moment pertama dan kedua dari gradien untuk mempercepat konvergensi.

Untuk menggambarkan alur lengkap proses pelatihan model, penelitian ini dilengkapi dengan sebuah *flowchart* seperti ditunjukkan pada Gambar 3.1. *Flowchart* tersebut disusun dengan mempertimbangkan arahan pembimbing serta mengacu pada berbagai referensi literatur yang relevan.

3.3.6 Perbandingan dengan Penelitian Sebelumnya

Hasil optimasi dengan metode LM dan GS akan dibandingkan dengan penelitian sebelumnya yang menggunakan optimasi lain seperti *backpropagation* atau metode *conjugate gradient*. Hasil yang akan dibandingkan mencakup kecepatan konvergensi dan akurasi klasifikasi.



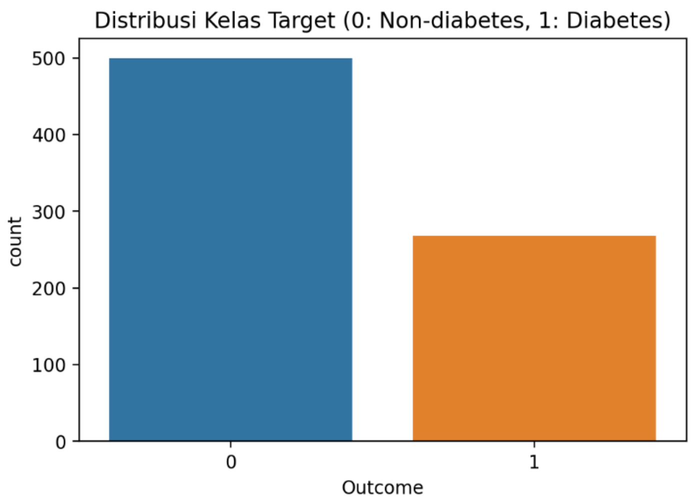
Gambar 3.1 *Flowchart* Proses Optimasi Regresi Logistik Menggunakan Metode Levenberg–Marquardt yang Memuat Gauss–Seidel.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Deskripsi Data

Data yang digunakan dalam penelitian ini adalah *Pima Indian Diabetes* Dataset yang bersumber dari repositori publik *Kaggle*. Dataset ini terdiri dari 768 baris data dengan delapan variabel independen (fitur) dan satu variabel dependen (label). Fitur-fitur tersebut merupakan fitur numerik yang meliputi jumlah kehamilan yang pernah dialami (*Pregnancies*), kadar glukosa plasma setelah dua jam tes (*Glucose*), tekanan darah diastolik (*BloodPressure*), ketebalan lipatan kulit trisep (*SkinThickness*), kadar insulin serum setelah dua jam (Insulin), indeks massa tubuh atau *Body Mass Index* (BMI), skor riwayat keluarga terhadap risiko diabetes (*DiabetesPedigreeFunction*), dan usia responden (*Age*). Sementara itu, variabel *Outcome* digunakan sebagai label dengan dua kategori, yaitu 1 untuk responden yang terdeteksi menderita diabetes dan 0 untuk responden yang tidak menderita diabetes.



Gambar 4.1 Distribusi Kelas Non-Diabetes dan Diabetes pada Data *Pima Indian Diabetes*

4.2 Hasil

Subbab ini menyajikan seluruh hasil yang diperoleh dari proses eksperimen menggunakan regresi logistik dengan beberapa metode optimasi. Penyusunan hasil ini memberikan gambaran menyeluruh mengenai kemampuan masing-masing metode dalam menyelesaikan klasifikasi biner.

4.2.1 Implementasi Metode Secara Manual

Pada subbab ini disajikan implementasi metode secara manual sebagai dasar untuk memahami proses perhitungan dalam regresi logistik dan algoritma optimasi yang digunakan, yaitu *Levenberg Marquardt* dan *Gauss Seidel*.

4.2.1.1 Pra Pemrosesan Data

Pada tahap pra pemrosesan data, dilakukan serangkaian langkah awal untuk memastikan bahwa data yang digunakan dalam pelatihan model berada dalam kondisi yang baik, konsisten, dan siap diproses oleh algoritma optimasi. Tahap ini penting karena kualitas hasil pelatihan sangat dipengaruhi oleh kualitas data masukan. Beberapa proses yang dilakukan meliputi standardisasi fitur dan pembagian data. Dengan adanya pra pemrosesan yang tepat, model regresi logistik dan metode optimasi yang digunakan dapat bekerja secara lebih stabil dan efisien.

Standardisasi fitur merupakan tahap penting dalam pra-pemrosesan data untuk memastikan bahwa setiap fitur numerik memiliki skala yang sebanding dan berdistribusi secara proporsional terhadap rata-ratanya. Proses standardisasi dilakukan menggunakan metode *Z-Score Standardization*, yaitu dengan mentransformasikan setiap nilai fitur agar memiliki rata-rata 0 dan simpangan baku 1.

Proses standardisasi diterapkan pada seluruh fitur numerik dalam *Pima Indian Diabetes Dataset* setelah data dibersihkan dari nilai tidak valid. Standardisasi dilakukan sebelum proses pembagian data menjadi data latih, sehingga model memperoleh input yang konsisten dan stabil selama proses pelatihan. Dengan transformasi ini, algoritma optimasi seperti *Levenberg-Marquardt* dan *Gauss-Seidel* dapat bekerja lebih efisien serta mencapai konvergensi yang lebih cepat.

Perhitungan nilai standardisasi untuk setiap data dilakukan menggunakan rumus *Z-Score Standardization* pada Persamaan (3.1).

Perhitungan standardisasi fitur $x_{i,j}$, berdasarkan data pada Tabel 4.1 untuk $i = 1, 2, \ldots, 6$ dan fitur ke- $j = 1$ dan 2 yaitu sebagai berikut:

Tabel 4.1 Sampel 6 Data Seimbang dan 2 fitur dengan Korelasi Tinggi

i	Fitur		Target
	Glucose	BMI	
1	97	38,1	0
2	78	33,7	0
3	111	29,5	0
4	144	31,6	1
5	138	36,1	1
6	171	45,4	1

$$X = \begin{bmatrix} 97 & 38{,}1 \\ 78 & 33{,}7 \\ 111 & 29{,}5 \\ 144 & 31{,}6 \\ 138 & 36{,}1 \\ 171 & 45{,}4 \end{bmatrix}$$

$$y = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Selanjutnya, yang dilakukan adalah menghitung nilai rata-rata (μ_j) dan simpangan baku (σ_j) untuk setiap fitur.

Untuk $j = 1$, maka

$$\mu_1 = \frac{97 + 78 + 111 + 144 + 138 + 171}{6} = 123{,}17$$

$$\sigma_1 = \sqrt{\frac{\sum_{i=1}^{6}\left(x_{i,1} - \mu_1\right)^2}{n-1}} = 35{,}18$$

Untuk $j = 2$, maka

$$\mu_2 = \frac{38{,}1 + 33{,}7 + 29{,}5 + 31{,}6 + 36{,}1 + 45{,}4}{6} = 35{,}73$$

$$\sigma_2 = \sqrt{\frac{\sum_{i=1}^{6}\left(x_{i,2} - \mu_2\right)^2}{n-1}} = 5{,}39$$

Berdasarkan Persamaan (3.1), maka perhitungan standardisasi untuk $x_{i,j}$. Adapun proses standardisasi untuk $x_{i,1}$ dan standardisasi untuk $x_{i,2}$ yaitu sebagai berikut:

1. Standardisasi untuk $x_{i,1}$

   a. Untuk $i = 1,\ j = 1$

   $$x'_{1,1} = \frac{97 - 123{,}17}{35{,}18} = -0{,}743$$

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

   $$x'_{2,1} = \frac{78 - 123{,}17}{35{,}18} = -1{,}282$$

c. Untuk $i = 3,\ j = 1$

$$x'_{3,1} = \frac{111 - 123{,}17}{35{,}18} = -0{,}346$$

Perhitungan nilai standardisasi untuk $x_{i,1}$ dengan $i = 4, 5,$ dan 6 dilakukan menggunakan prosedur yang sama seperti pada perhitungan sebelumnya.

2. Standardisasi untuk $x_{i,2}$

a. Untuk $i = 1,\ j = 2$

$$x'_{1,2} = \frac{38{,}1 - 35{,}73}{5{,}39} = 0{,}441$$

b. Untuk $i = 2,\ j = 2$

$$x'_{2,2} = \frac{33{,}7 - 35{,}73}{5{,}39} = -0{,}377$$

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

$$x'_{3,2} = \frac{29{,}5 - 35{,}73}{5{,}39} = -1{,}155$$

Perhitungan nilai standardisasi untuk $x_{i,2}$ dengan $i = 4, 5,$ dan 6 dilakukan menggunakan prosedur yang sama seperti pada perhitungan sebelumnya.

3. Hasil Standardisasi

Setelah dilakukan proses standardisasi data menggunakan metode *Z-Score Standardization*, yaitu berdasarkan nilai rata-rata dan simpangan baku masing-masing fitur, hasil standardisasi dari data pada Tabel 4.1 yaitu sebagai berikut.

$$X' = \begin{bmatrix} -0{,}743 & 0{,}441 \\ -1{,}282 & -0{,}377 \\ -0{,}346 & -1{,}155 \\ 0{,}591 & -0{,}765 \\ 0{,}421 & 0{,}069 \\ 1{,}358 & 1{,}791 \end{bmatrix}$$

Data hasil standardisasi akan digunakan sebagai input dalam proses pelatihan model regresi logistik, yang pembahasannya akan diuraikan pada subbab selanjutnya.

4.2.1.2 Inisialisasi Bobot

Inisialisasi bobot merupakan langkah awal dalam proses pelatihan model, di mana setiap bobot diberi nilai awal sebelum proses iteratif dilakukan guna memperoleh bobot yang optimal. Pemilihan nilai bobot awal dapat dilakukan secara acak, biasanya bobot awal dipilih dalam rentang *[0,1]* untuk menghindari nilai aktivasi yang terlalu besar. Pada contoh ini, bobot awal ditetapkan sebesar $\boldsymbol{a} = (0{,}1;\ 0{,}2;\ 0{,}3)$, dengan vektor fitur didefinisikan sebagai $\boldsymbol{x}_i = (1, x_{i,1}, x_{i,2})$. Nilai z_i dihitung menggunakan kombinasi linear antara bobot dan fitur sesuai Persamaan (2.1).

1. Menghitung z_i

   a. Untuk $i = 1$

   Maka $\boldsymbol{x}_1 = (1;\ -0{,}743;\ 0{,}441)$

   Selanjutnya z_1 dapat dihitung sebagai berikut

   $$\begin{aligned} z_1 &= \boldsymbol{a} \cdot \boldsymbol{x}_1 \\ &= 0{,}1 + 0{,}2 \times (-0{,}743) + 0{,}3 \times 0{,}441 \\ &= 0{,}084 \end{aligned}$$

   b. Untuk $i = 2$

   Maka $\boldsymbol{x}_2 = (1;\ -1{,}282;\ -0{,}377)$

   Selanjutnya z_2 dapat dihitung sebagai berikut

   $$z_2 = \boldsymbol{a} \cdot \boldsymbol{x}_2$$

$$= 0{,}1 + 0{,}2 \times (-1{,}282) + 0{,}3 \times (-0{,}377)$$

$$= -0{,}269$$

c. Untuk $i = 3$

Maka $\boldsymbol{x}_3 = (1;\ -0{,}346;\ -1{,}155)$

Selanjutnya z_3 dapat dihitung sebagai berikut

$$z_3 = \boldsymbol{a} \cdot \boldsymbol{x}_3$$

$$= a_0 + a_1 \times x_{31} + a_2 \times x_{32}$$

$$= 0{,}1 + 0{,}2 \times (-0{,}346) + 0{,}3 \times (-1{,}155)$$

$$= -0{,}316$$

Perhitungan nilai z_i untuk $i = 4, 5,$ dan 6 dilakukan menggunakan prosedur yang sama seperti pada perhitungan sebelumnya. Hasil perhitungan nilai z_i untuk 6 data ditunjukkan pada Tabel 4.2.

Tabel 4.2 Hasil Perhitungan Nilai z_i

i	Nilai z_i
1	0,084
2	−0,269
3	−0,316
4	−0,011
5	0,205
6	0,909

2. Menghitung $\sigma(z_i)$

a. Untuk $i = 1$

$$\sigma(z_1) = \frac{1}{1 + \exp\,(-z_1)}$$

$$= \frac{1}{1 + \exp(-0{,}084)}$$

$$= 0{,}521$$

b. Untuk $i = 2$

$$\sigma(z_2) = \frac{1}{1 + \exp\,(-z_2)}$$

$$= \frac{1}{1 + \exp(0{,}269)}$$

$$= 0{,}433$$

c. Untuk $i = 3$

$$\sigma(z_3) = \frac{1}{1 + \exp\,(-z_3)}$$

$$= \frac{1}{1 + \exp(0{,}316)}$$

$$= 0{,}422$$

Perhitungan nilai $\sigma(z_i)$ untuk $i = 4, 5,$ dan 6 dilakukan menggunakan prosedur yang sama seperti pada perhitungan sebelumnya. Hasil perhitungan nilai $\sigma(z_i)$ untuk 6 data ditunjukkan pada Tabel 4.3.

Tabel 4.3 Hasil Perhitungan Nilai $\sigma(z_i)$

i	Nilai $\sigma(z_i)$
1	0,521
2	0,433
3	0,422
4	0,497
5	0,551
6	0,712

4.2.1.3 Perhitungan Fungsi *Loss*

Setelah memperoleh nilai $\sigma(z_i)$, langkah selanjutnya adalah menghitung error untuk setiap sampel data dengan menggunakan rumus pada Persamaan (2.11).

Menghitung L_i dengan $i = 1, 2, \dots, 6$

1. Untuk $i = 1$

$$L_1 = -\big(y_1 \log\big(\sigma(z_1)\big) + (1 - y_1) \log\big(1 - \sigma(z_1)\big)\big)$$

$$= -(0 \log(0{,}521) + (1 - 0) \log(1 - 0{,}521))$$

$$= 0{,}737$$

2. Untuk $i = 2$

$$L_2 = -\big(y_2 \log\big(\sigma(z_2)\big) + (1 - y_2) \log\big(1 - \sigma(z_2)\big)\big)$$

$$= -(0 \log(0{,}433) + (1 - 0) \log(1 - 0{,}433))$$

$$= 0{,}568$$

3. Untuk $i = 3$

$$L_3 = -\big(y_3 \log\big(\sigma(z_3)\big) + (1 - y_3) \log\big(1 - \sigma(z_3)\big)\big)$$

$$= -(0 \log(0{,}422) + (1 - 0) \log(1 - 0{,}422))$$

$$= 0{,}548$$

Perhitungan nilai L_i untuk $i = 4, 5,$ dan 6 dilakukan menggunakan prosedur yang sama seperti pada perhitungan sebelumnya. Hasil perhitungan nilai L_i untuk 6 data ditunjukkan pada Tabel 4.4.

Tabel 4.4 Hasil Perhitungan Nilai L_i

i	Nilai L_i
1	0,737
2	0,568
3	0,548
4	0,714
5	0,805
6	0,335

4.2.1.4 Implementasi Metode *Levenberg Marquardt* pada Model Regresi Logistik

Perhitungan ∇L_i dilakukan dengan menurunkan fungsi *loss* terhadap masing-masing parameter bobot. Gradien sendiri merupakan vektor yang berisi turunan parsial dari fungsi *loss* terhadap setiap bobot, sehingga menggambarkan seberapa besar dan ke arah mana perubahan bobot akan memengaruhi nilai *loss*.

Pada data ke-i, hasil turunan ini menghasilkan nilai gradien untuk setiap fitur, yang kemudian digunakan pada tahap optimasi untuk memperbarui bobot agar *loss* semakin kecil. Dengan demikian, untuk setiap data, nilai gradien ∇L_i diperoleh melalui turunan fungsi *loss* terhadap bobot sebagaimana ditunjukkan pada Persamaan (3.2).

1. Menghitung $\nabla L_1 = \left(\frac{\partial L_1}{\partial a_0}, \frac{\partial L_1}{\partial a_1}, \frac{\partial L_1}{\partial a_2}\right)$,

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

$$\frac{\partial L_1}{\partial a_0} = \big(y_1 - \sigma(z_1)\big)x_{1,0}$$
$$= (0 - 0{,}521)1$$
$$= -0{,}521$$

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

$$\frac{\partial L_1}{\partial a_1} = \big(y_1 - \sigma(z_1)\big)x_{1,1}$$
$$= (0 - 0{,}521)(-0{,}743)$$
$$= 0{,}387$$

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

$$\frac{\partial L_1}{\partial a_2} = \left(y_1 - \sigma(z_1)\right)x_{1,2}$$

$$= (0 - 0{,}521)0{,}441$$

$$= -0{,}230$$

Hasil perhitungan nilai ∇L_1 yaitu sebagai berikut.

$$\nabla L_1 = (-0{,}521;\ 0{,}387;\ -0{,}230)$$

2. Menghitung $\nabla L_2 = \left(\frac{\partial L_2}{\partial a_0}, \frac{\partial L_2}{\partial a_1}, \frac{\partial L_2}{\partial a_2}\right)$

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

$$\frac{\partial L_2}{\partial a_0} = (y_2 - \sigma(z_2))x_{2,0}$$

$$= (0 - 0{,}433)1$$

$$= -0{,}433$$

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

$$\frac{\partial L_2}{\partial a_1} = \left(y_2 - \sigma(z_2)\right)x_{2,1}$$

$$= (0 - 0{,}434)(-1{,}282)$$

$$= 0{,}556$$

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

$$\frac{\partial L_2}{\partial a_2} = \left(y_2 - \sigma(z_2)\right)x_{2,2}$$

$$= (0 - 0{,}434)(-0{,}377)$$

$$= 0{,}164$$

Hasil perhitungan nilai ∇L_2 yaitu sebagai berikut

$$\nabla L_2 = (-0{,}433;\ 0{,}556;\ 0{,}164)$$

3. Menghitung $\nabla L_3 = \left(\frac{\partial L_3}{\partial a_0}, \frac{\partial L_3}{\partial a_1}, \frac{\partial L_3}{\partial a_2}\right)$

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

$$\frac{\partial L_3}{\partial a_0} = \left(y_3 - \sigma(z_3)\right)x_{3,0}$$

$$= (0 - 0{,}422)1$$

$$= -0{,}422$$

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

$$\frac{\partial L_3}{\partial a_1} = \left(y_3 - \sigma(z_3)\right)x_{31}$$

$$= (0 - 0{,}422)(-0{,}346)$$

$$= 0{,}146$$

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

$$\frac{\partial L_3}{\partial a_2} = \left(y_3 - \sigma(z_3)\right)x_{32}$$

$$= (0 - 0{,}422)(-1{,}155)$$

$$= 0{,}487$$

Hasil perhitungan nilai ∇L_3 yaitu sebagai berikut

$$\nabla L_3 = (-0{,}422;\ 0{,}146;\ 0{,}487)$$

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.

$$\nabla L_1 = (-0{,}521;\ 0{,}387;\ -0{,}230) \qquad \nabla L_2 = (-0{,}433;\ 0{,}556;\ 0{,}164)$$
$$\nabla L_3 = (-0{,}422;\ 0{,}146;\ 0{,}487) \qquad \nabla L_4 = (0{,}503; 0{,}297;\ -0{,}385)$$
$$\nabla L_5 = (0{,}449; 0{,}189; 0{,}031) \qquad \nabla L_6 = (0{,}287; 0{,}390; 0{,}514)$$

4.2.1.5 Membangun Matriks Jacobian dan Matriks RHS (*Right Hand Side*)

Setelah menghitung gradien untuk setiap parameter bobot, langkah selanjutnya dalam proses optimasi adalah menyusun Matriks Jacobian dan Matriks RHS (*Right Hand Side*). Matriks Jacobian dibangun dari turunan *loss* terhadap setiap parameter model. Nilai turunan ini pada dasarnya merupakan komponen gradien per sampel. Dengan demikian, Jacobian dapat dipandang sebagai kumpulan gradien yang disusun dalam bentuk matriks. Sedangkan Matriks RHS digunakan untuk membentuk sistem persamaan linear yang akan diselesaikan dalam proses pembaruan parameter. Kedua matriks ini merupakan komponen penting dalam metode optimasi numerik seperti *Gauss-Newton* maupun *Levenberg-Marquardt*, yang bertujuan untuk mempercepat konvergensi menuju solusi optimal. Penyusunan kedua matriks ini dilakukan berdasarkan data yang tersedia dan hasilnya akan digunakan pada tahap iterasi pembaruan bobot. Setelah dilakukan perhitungan dengan data yang tersedia, Matriks Jacobian yang dihasilkan ditunjukkan sebagai berikut:

$$\boldsymbol{J} = \begin{bmatrix} -0{,}521 & 0{,}387 & -0{,}230 \\ -0{,}433 & 0{,}556 & 0{,}164 \\ -0{,}422 & 0{,}146 & 0{,}487 \\ 0{,}503 & 0{,}297 & -0{,}385 \\ 0{,}449 & 0{,}189 & 0{,}031 \\ 0{,}287 & 0{,}390 & 0{,}514 \end{bmatrix}$$

Tahapan selanjutnya setelah membentuk Matriks Jacobian adalah menyusun Matriks *Right Hand Side* (RHS), yang diperoleh dari hasil kalkulasi terhadap fungsi *loss* sebagaimana ditunjukkan pada persamaan berikut:

$$\boldsymbol{B} = \begin{bmatrix} 0,737 \\ 0,568 \\ 0,548 \\ 0,714 \\ 0,805 \\ 0,335 \end{bmatrix}$$

Setelah proses pembentukan Matriks Jacobian dan Matriks RHS (B) selesai, tahap selanjutnya adalah mentranspos Matriks Jacobian guna memperoleh hasil perhitungan $\boldsymbol{J}^T\boldsymbol{J}$ dan $\boldsymbol{J}^T\boldsymbol{B}$, yang berperan penting dalam sistem persamaan linier pada metode optimasi.

$$\boldsymbol{J}^T = \begin{bmatrix} -0,521 & -0,433 & -0,422 & 0,503 & 0,449 & 0,287 \\ 0,387 & 0,556 & 0,146 & 0,297 & 0,189 & 0,390 \\ -0,230 & 0,164 & 0,487 & -0,385 & 0,031 & 0,514 \end{bmatrix}$$

Setelah diperoleh Matriks Jacobian dan Matriks RHS (D), kemudian menghitung $\boldsymbol{J}^T\boldsymbol{J}$

$$\boldsymbol{J}^T\boldsymbol{J} = \begin{bmatrix} 1,174 & -0,157 & -0,188 \\ -0,157 & 0,755 & 0,165 \\ -0,188 & 0,165 & 0,730 \end{bmatrix}$$

Setelah menghitung $\boldsymbol{J}^T\boldsymbol{J}$ kemudian menghitung $\boldsymbol{J}^T\boldsymbol{B}$

$\boldsymbol{J}^T\boldsymbol{B}$

$$= \begin{bmatrix} -0,521 & -0,433 & -0,422 & 0,503 & 0,449 & 0,287 \\ 0,387 & 0,556 & 0,146 & 0,297 & 0,189 & 0,390 \\ -0,230 & 0,164 & 0,487 & -0,385 & 0,031 & 0,514 \end{bmatrix} \begin{bmatrix} 0,737 \\ 0,568 \\ 0,548 \\ 0,714 \\ 0,805 \\ 0,335 \end{bmatrix}$$

$$= \begin{bmatrix} -0,044 \\ 1,176 \\ 0,113 \end{bmatrix}$$

Kemudian menghitung matriks *Hessian Aproximation*

$$\boldsymbol{M} = \boldsymbol{J}^T\boldsymbol{J} + \lambda\boldsymbol{I}$$

$$= \begin{bmatrix} 1{,}174 & -0{,}157 & -0{,}188 \\ -0{,}157 & 0{,}755 & 0{,}165 \\ -0{,}188 & 0{,}165 & 0{,}730 \end{bmatrix} + 0{,}1 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1{,}274 & -0{,}157 & -0{,}188 \\ -0{,}157 & 0{,}855 & 0{,}165 \\ -0{,}188 & 0{,}165 & 0{,}830 \end{bmatrix}$$

Matriks M ini digunakan untuk menentukan arah pembaruan parameter. Pada metode *Gauss–Newton*, Hessian aproksimasi adalah $\boldsymbol{J}^T\boldsymbol{J}$, sedangkan pada *Levenberg–Marquardt* diberikan tambahan $\lambda\boldsymbol{I}$ untuk meningkatkan kestabilan numerik.

4.2.1.6 Implementasi Metode *Gauss Seidel* untuk Penyelesaian SPL

Setelah diperoleh matriks *Hessian Aproximation* dan $\boldsymbol{J}^T\boldsymbol{B}$, yang disimbolkan sebagai $\boldsymbol{c}$, tahap selanjutnya adalah menyelesaikan sistem persamaan linear dengan rumus seperti pada Persamaan (2.19).

$$\boldsymbol{M\eta} = \boldsymbol{c}$$

$$\begin{bmatrix} 1{,}274 & -0{,}157 & -0{,}188 \\ -0{,}157 & 0{,}855 & 0{,}165 \\ -0{,}188 & 0{,}165 & 0{,}830 \end{bmatrix} \begin{bmatrix} \Delta a_0 \\ \Delta a_1 \\ \Delta a_2 \end{bmatrix} = \begin{bmatrix} -0{,}044 \\ 1{,}176 \\ 0{,}113 \end{bmatrix}$$

Pada konteks metode Levenberg–Marquardt, η merupakan vektor koreksi parameter, yaitu besarnya perubahan bobot yang akan ditambahkan pada parameter saat ini. Dengan demikian, penyelesaian SPL ini menjadi tahap penting untuk menghitung update parameter LM. Matriks $\boldsymbol{M}$ bersifat simetris, karena seluruh elemennya memenuhi $\boldsymbol{M}_{i,j} = \boldsymbol{M}_{j,i}$. Selain itu matriks $\boldsymbol{M}$ juga bersifat positif definit karena minor utamanya bernilai positif: minor orde pertama $1{,}274 > 0$, minor orde kedua $1{,}088 > 0$, dan determinan penuh $0{,}854 > 0$, sehingga metode *Gauss–Seidel* dapat digunakan pada sistem ini dengan rumus seperti ditunjukkan pada Persamaan (2.21). Berikut proses iterasi yang dilakukan.

1. Iterasi 1 $(k = 0,)$

   a. Menghitung $\Delta a_0^{(1)}$ $(j = 0)$

$$\Delta a_0^{(1)} = \frac{c_1 - \left(\boldsymbol{M}_{0,1}\Delta a_1^{(0)} + \boldsymbol{M}_{0,2}\Delta a_2^{(0)}\right)}{\boldsymbol{M}_{0,0}}$$

$$= \frac{(-0{,}044) - (-0{,}157 \times 0 + (-0{,}188) \times 0)}{1{,}274}$$

$$= \frac{-0{,}044}{1{,}274}$$

$$= -0{,}035$$

   b. Menghitung $\Delta a_1^{(1)}$ $(j = 1)$

$$\Delta a_1^{(1)} = \frac{1{,}176 - (((-0{,}157) \times (-0{,}035)) + (0{,}165 \times 0))}{0{,}855}$$

$$= \frac{1{,}176 - 0{,}005 - 0}{0{,}855}$$

$$= 1{,}369$$

   c. Menghitung $\Delta a_2^{(1)}$ $(j = 2)$

$$\Delta a_2^{(1)} = \frac{0{,}113 - (((-0{,}188) \times (-0{,}035)) + (0{,}165 \times 1{,}369))}{0{,}830}$$

$$= \frac{0{,}113 - 0{,}232}{0{,}830}$$

$$= -0{,}144$$

Hasil setelah iterasi 1:

$$\Delta a^{(1)} = [-0{,}035;\ 1{,}369;\ -0{,}144]^T$$

2. Iterasi 2 ($k = 1$)

a. Menghitung $\Delta a_0^{(2)}$ $j = 0$

$$\Delta a_0^{(2)}$$

$$= \frac{-0{,}044 - \big((-0{,}157) \times 1{,}369 + (-0{,}188) \times (-0{,}144)\big)}{1{,}274}$$

$$= \frac{-0{,}044 - (-0{,}188)}{1{,}274}$$

$$= 0{,}112$$

b. Menghitung $\Delta a_1^{(2)}$ $j = 1$

$$\Delta a_1^{(2)} = \frac{1{,}176 - ((-0{,}157) \times 0{,}112 + 0{,}165 \times (-0{,}144))}{0{,}855}$$

$$= \frac{1{,}176 - (-0{,}017) - (-0{,}024)}{0{,}855}$$

$$= 1{,}424$$

c. Menghitung $\Delta a_2^{(2)}$ $j = 2$

$$\Delta a_2^{(2)} = \frac{0{,}113 - ((-0{,}188) \times 0{,}112 + 0{,}165 \times 1{,}424)}{0{,}830}$$

$$= \frac{0{,}113 - 0{,}214}{0{,}830}$$

$$= -0{,}122$$

Hasil setelah iterasi 2:

$$\Delta a^{(2)} = [0{,}112;\ 1{,}424;\ -0{,}122]^T$$

3. Iterasi 3 ($k = 2$)

a. Menghitung $\Delta a_0^{(3)}$ $j = 0$

$$\Delta a_0^{(3)}$$

$$= \frac{-0{,}044 - (((-0{,}157) \times 1{,}424) + ((-0{,}188) \times (-0{,}122)))}{1{,}274}$$

$$= \frac{-0{,}044 - (-0{,}200)}{1{,}274}$$

$$= 0{,}123$$

b. Menghitung $\Delta a_1^{(3)}$ $j = 1$

$$\Delta a_1^{(3)} = \frac{1{,}176 - \left(((-0{,}157) \times 0{,}123) + (0{,}165 \times (-0{,}122))\right)}{0{,}855}$$

$$= \frac{1{,}176 - (-0{,}019) - (-0{,}020)}{0{,}855}$$

$$= 1{,}421$$

c. Menghitung $\Delta a_2^{(3)}$ $j = 2$

$$\Delta a_2^{(3)} = \frac{0{,}113 - ((-0{,}188) \times 0{,}123 + 0{,}165 \times 1{,}421)}{0{,}830}$$

$$= \frac{-0{,}101 - (-0{,}062)}{0{,}830}$$

$$= -0{,}119$$

Hasil setelah iterasi 3:

$$\Delta a^{(3)} = [0{,}123;\ 1{,}421;\ -0{,}119]^T$$

4.2.2 Implementasi Metode Secara Komputasional

4.2.2.1 Pra Pemrosesan Data

Tahapan pra-pemrosesan data dilakukan untuk memastikan bahwa dataset *Pima Indian Diabetes* siap digunakan dalam proses pelatihan model regresi logistik. Proses ini dilakukan secara komputasional menggunakan bahasa pemrograman *Python*.

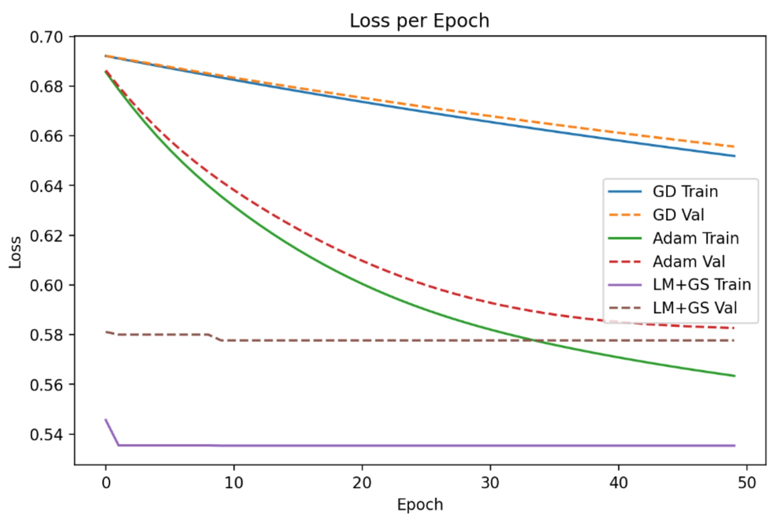
Tahap awal yang dilakukan adalah melakukan standardisasi terhadap seluruh fitur pada dataset agar setiap variabel memiliki skala yang seragam. Proses standardisasi dilakukan menggunakan *StandardScaler* dari pustaka *Scikit-learn*, sehingga setiap fitur memiliki nilai rata-rata (*mean*) 0 dan deviasi standar (*standard deviation*) 1. Standardisasi ini bertujuan untuk mencegah dominasi fitur tertentu akibat perbedaan skala serta membantu proses pembelajaran model berjalan lebih stabil.

Setelah proses standardisasi selesai, dataset kemudian dibagi menjadi tiga bagian, yaitu data latih sebesar 70% dengan jumlah data 538, data validasi sebesar 15% dengan jumlah data 115, dan data uji sebesar 15% dengan jumlah data 115. Pembagian data dilakukan secara acak menggunakan fungsi *train_test_split*. Pembagian ini bertujuan agar model dapat dievaluasi menggunakan data yang tidak pernah digunakan selama proses pelatihan, sehingga kemampuan generalisasi model terhadap data baru dapat diukur dengan lebih baik.

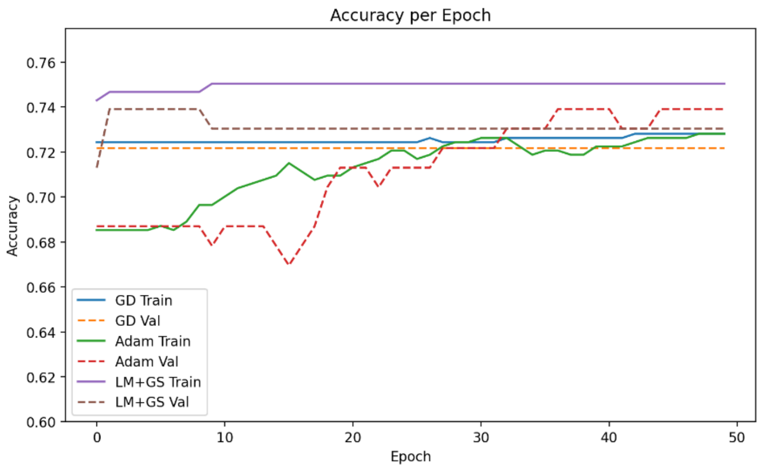
4.2.2.2 Kurva Pembelajaran

Analisis kurva pembelajaran dilakukan untuk memahami bagaimana model mempelajari pola dari data selama proses pelatihan, serta untuk menilai kestabilan proses optimasi dan mendeteksi potensi terjadinya *underfitting* maupun *overfitting*. Kurva pembelajaran menyajikan perubahan nilai *loss* dan akurasi pada data latih dan data validasi terhadap jumlah *epoch*. Pada penelitian ini, kurva pembelajaran digunakan untuk membandingkan performa tiga metode optimasi, yaitu *Levenberg–Marquardt* dengan *Gauss–Seidel* (LMGS), *Gradient Descent* (GD), dan *Adam Optimizer* (Adam).

Proses pelatihan model pada masing-masing metode optimasi dilakukan dengan pengaturan parameter (*hyperparameter*) yang sama agar hasil perbandingan menjadi objektif. Jumlah *epoch* yang digunakan adalah 50 *epoch*, di mana setiap *epoch* merepresentasikan satu siklus pembaruan bobot terhadap seluruh data pelatihan. Dataset dibagi menjadi 70% data latih, 15% data validasi, dan 15% data uji, tanpa menggunakan *mini-batch* (seluruh data pelatihan digunakan sekaligus pada setiap iterasi). Penggunaan *full-batch* dipilih karena metode *Levenberg–Marquardt* membutuhkan perhitungan matriks Jacobian dan aproksimasi Hessian yang stabil pada setiap iterasi. Jika *mini-batch* digunakan, nilai matriks $\boldsymbol{J}^T\boldsymbol{J}$ dan $\boldsymbol{J}^T\boldsymbol{B}$ akan berubah-ubah pada setiap *batch* sehingga mengganggu kestabilan pembentukan sistem persamaan linear yang harus diselesaikan dengan *Gauss–Seidel*. Sehingga penggunaan *full-batch* memberikan konvergensi yang lebih baik dan hasil yang lebih konsisten.



Gambar 4.2 Perbandingan *Loss* Data *Training* dan *Validation*



Gambar 4.3 Perbandingan Akurasi Data *Training* dan *Validation*

Berdasarkan Gambar 4.2 dan Gambar 4.3, terlihat bahwa metode LMGS menunjukkan performa yang lebih unggul dibandingkan metode optimasi dasar seperti *Gradient Descent* (GD) dan Adam. Pada metode GD, penurunan *loss* berlangsung lambat dan hampir linear, sementara akurasi pelatihan dan validasi cenderung stagnan di sekitar nilai 0,72. Kondisi ini menunjukkan bahwa GD memiliki kemampuan pembelajaran yang kurang optimal dan mendekati kondisi *underfitting*.

Metode Adam menunjukkan penurunan *loss* yang lebih cepat dibandingkan GD, namun akurasi data latih maupun validasi masih mengalami fluktuasi pada awal pelatihan dan baru stabil menjelang akhir *epoch* di sekitar nilai 0,73. Hal ini mengindikasikan bahwa Adam lebih adaptif, tetapi kestabilan pembelajarannya belum sepenuhnya optimal.

Berbeda dengan kedua metode tersebut, LMGS memberikan hasil yang jauh lebih stabil dan efektif. *Loss* data latih mengalami penurunan tajam pada awal pelatihan hingga mencapai nilai rendah sekitar 0,54, kemudian tetap stabil hingga akhir *epoch*. *Validation loss* juga berada pada nilai rendah dan tidak menunjukkan peningkatan berarti, sehingga mengindikasikan tidak terjadi *overfitting*. Selain itu, akurasi latih dan validasi berada pada kisaran yang tinggi dan stabil, yaitu 0,74–0,75, tanpa fluktuasi signifikan. Hal ini menunjukkan bahwa metode LMGS memiliki proses konvergensi yang cepat, stabil, serta kemampuan generalisasi yang baik.

Secara keseluruhan, kurva pembelajaran menunjukkan bahwa LMGS merupakan metode yang paling stabil dan konsisten dibandingkan GD dan Adam. Tidak ditemukan indikasi *underfitting* maupun *overfitting* pada LMGS. Secara teoretis, *underfitting* biasanya muncul ketika model gagal mempelajari pola dasar data, yang tampak dari train *loss* dan val *loss* yang sama-sama tinggi serta akurasi yang rendah. Sementara itu, *overfitting* terjadi ketika model terlalu fokus pada data pelatihan hingga menghasilkan train *loss* yang sangat rendah, tetapi val *loss* justru lebih tinggi atau cenderung naik, disertai gap akurasi yang besar antara pelatihan dan validasi. Berdasarkan grafik, metode LMGS menunjukkan performa yang paling baik dibandingkan GD dan Adam. Train *loss* LMGS langsung berada pada nilai yang jauh lebih rendah dan stabil sejak awal, sedangkan GD menurun lambat dan Adam masih mengalami fluktuasi. Meskipun validasi *loss* LMGS tidak serendah train *loss*-nya, kurvanya tetap stabil dan tidak menunjukkan pola kenaikan yang menandakan *overfitting*. Selain itu, akurasi pelatihan LMGS merupakan yang tertinggi dan akurasi validasinya konsisten berada sedikit di atas atau setara dengan Adam dan GD. Dengan demikian, LMGS dapat dikatakan sebagai metode optimasi yang paling efektif pada kasus ini.

4.2.2.3 Kinerja Model

Selain perbandingan berdasarkan kurva pembelajaran, dilakukan pula analisis terhadap kecepatan konvergensi dari masing-masing metode optimasi. Analisis ini bertujuan untuk mengetahui efisiensi waktu dan jumlah *epoch* yang dibutuhkan setiap metode dalam mencapai akurasi pelatihan tertentu. Hasil pengujian perbandingan waktu dan jumlah *epoch* hingga model mencapai akurasi pelatihan $\geq$ 0,75 ditunjukkan pada Tabel 4.5.

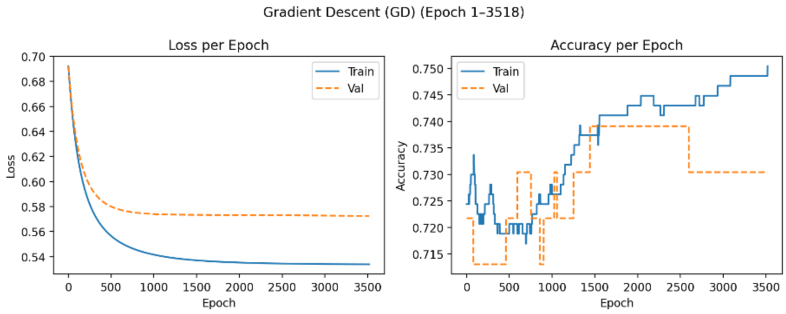
Tabel 4.5 Perbandingan Jumlah *Epoch* dan Waktu Hingga Akurasi Pelatihan $\geq$ 0,75

Metode	*Epoch*	*Time* (ms)
LMGS	10	5,76 ms
Adam	227	59,38 ms
GD	3518	865,91 ms

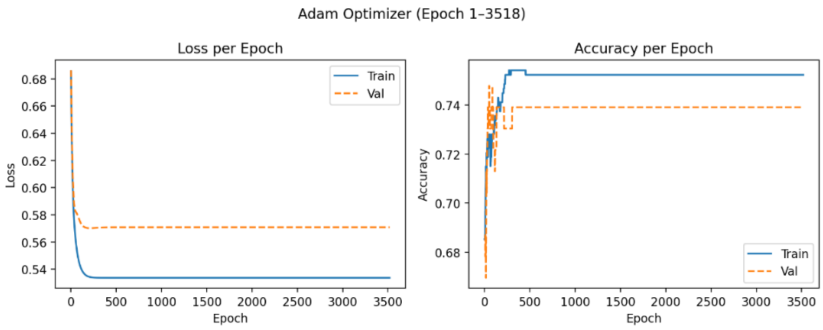
Dari hasil pada Tabel 4.5, terlihat bahwa metode LMGS memiliki kecepatan konvergensi yang jauh lebih tinggi dibandingkan dengan dua metode dasar lainnya. Metode LMGS hanya membutuhkan 10 *epoch* dengan waktu komputasi sekitar 5,76 milidetik untuk mencapai akurasi pelatihan $\geq$ 0,75. Sebaliknya, metode Adam memerlukan 227 *epoch* dengan waktu sekitar 59,38 milidetik, sedangkan GD membutuhkan waktu yang paling lama, yaitu 3518 *epoch* dan 865,91 milidetik.

Hasil ini menunjukkan bahwa LMGS mampu mempercepat proses konvergensi secara signifikan, karena modifikasi *Levenberg–Marquardt* yang dikombinasikan dengan iterasi *Gauss–Seidel* mampu menyesuaikan parameter dengan lebih efisien pada setiap langkah iterasi. Dengan demikian, metode LMGS tidak hanya menghasilkan akurasi lebih tinggi pada data uji, tetapi juga mencapai titik konvergensi dalam waktu pelatihan yang jauh lebih singkat dibandingkan *optimizer* dasar seperti Adam dan GD.

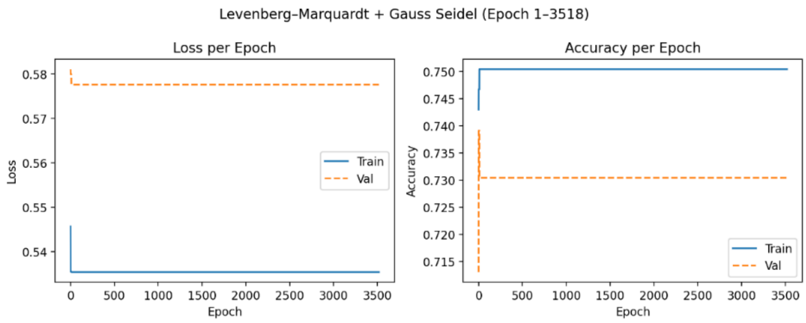
Untuk melengkapi analisis numerik pada Tabel 4.5, ditambahkan pula visualisasi berupa kurva pembelajaran (*learning curves*) untuk masing-masing metode optimasi, yaitu GD, Adam, dan LMGS. Pada Gambar 4.4 sampai 4.6 memberikan gambaran yang lebih jelas mengenai dinamika perubahan *loss* dan akurasi sepanjang proses pelatihan.



Gambar 4.4 Grafik *Gradient Descent*



Gambar 4.5 Grafik Adam



Gambar 4.6 Grafik LMGS

Berdasarkan Gambar 4.4, GD menunjukkan penurunan *training loss* yang berlangsung sangat lambat dan membutuhkan hingga ribuan epoch untuk mencapai kondisi relatif stabil. Validasi *loss* cenderung berhenti menurun lebih awal dan berada di atas *training loss*, yang mengindikasikan adanya keterbatasan kemampuan generalisasi meskipun tidak terjadi overfitting yang ekstrem. Pola akurasi juga meningkat secara bertahap, dengan fluktuasi pada fase awal dan konvergensi yang lambat menuju nilai akhir sekitar 0,75 untuk data latih dan sedikit lebih rendah untuk data validasi. Hal ini menegaskan bahwa GD memiliki konvergensi yang stabil tetapi kurang efisien secara waktu komputasi.

Pada Gambar 4.5, terlihat penurunan *loss* pada Adam jauh lebih cepat dibandingkan GD. Baik *training loss* maupun validasi *loss* mencapai nilai stabil hanya dalam beberapa ratus epoch awal. Akurasi pelatihan meningkat dengan cepat, diikuti oleh akurasi validasi yang relatif stabil namun sedikit lebih rendah. Perbedaan kecil antara kurva latih dan validasi menunjukkan generalisasi yang cukup baik, meskipun peningkatan performa berhenti lebih dini. Ini mencerminkan karakteristik Adam yang adaptif dan efisien, tetapi berpotensi terjebak pada solusi suboptimal ketika kurva sudah mendatar.

Sementara itu pada Gambar 4.6, LMGS menunjukkan konvergensi yang paling cepat dan stabil. *Training loss* langsung berada pada nilai terendah sejak epoch awal dan hampir tidak berubah hingga akhir, sedangkan validasi *loss* juga stabil dengan selisih yang relatif kecil terhadap *training loss*. Akurasi pelatihan mencapai nilai tertinggi secara sangat cepat dan tetap konsisten, disertai akurasi validasi yang lebih baik dibandingkan GD dan Adam. Pola ini mengindikasikan bahwa metode LMGS mampu memanfaatkan informasi orde kedua secara efektif, sehingga menghasilkan konvergensi cepat dan performa generalisasi yang lebih baik. Dengan demikian, grafik kurva pembelajaran dan data numerik keduanya mengonfirmasi bahwa LMGS merupakan metode yang paling efisien dalam hal kecepatan konvergensi sekaligus menghasilkan kinerja model yang lebih stabil.

Efisiensi dalam hal kecepatan konvergensi sekaligus menghasilkan kinerja model yang lebih stabil dapat diamati melalui perbandingan nilai *loss* dan akurasi pada beberapa epoch sebagaimana disajikan pada Tabel 4.6 dan Tabel 4.7.

Tabel 4.6 Perbandingan Nilai *Loss* Data *Training* dan Validasi pada Beberapa *Epoch* untuk Metode GD, Adam, dan LMGS

Epoch	GD Train *Loss*	GD Val *Loss*	Adam Train *Loss*	Adam Val *Loss*	LMGS Train *Loss*	LMGS Val *Loss*
1	0,690	0,695	0,688	0,690	0,545	0,580
10	0,660	0,630	0,580	0,570	0,540	0,580
227	0,560	0,580	0,535	0,570	0,535	0,578
3518	0,535	0,575	0,535	0,570	0,535	0,578

Tabel 4.7 Perbandingan Nilai Akurasi Data Training dan Validasi pada Beberapa Epoch untuk Metode GD, Adam, dan LMGS

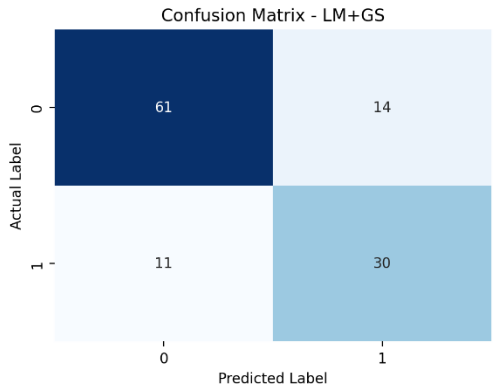
Epoch	GD Train Akurasi	GD Val Akurasi	Adam Train Akurasi	Adam Val Akurasi	LMGS Train Akurasi	LMGS Val Akurasi
1	0,720	0,710	0,670	0,665	0,750	0,715
10	0,720	0,720	0,740	0,730	0,750	0,730
227	0,730	0,730	0,750	0,740	0,750	0,730
3518	0,750	0,730	0,750	0,740	0,750	0,730

Berdasarkan Tabel 4.6, metode LMGS menunjukkan kinerja paling baik karena menghasilkan nilai *loss* yang paling rendah sejak *epoch* awal dan tetap stabil hingga akhir pelatihan. Metode Adam mengalami penurunan *loss* yang cukup cepat, namun nilai loss-nya masih berada di atas LMGS pada sebagian besar *epoch* yang diamati. Sementara itu, GD memiliki nilai *loss* yang paling tinggi dan memerlukan jumlah *epoch* yang lebih banyak untuk mencapai nilai yang mendekati metode lainnya. Hal ini menunjukkan bahwa LMGS lebih efektif dalam menurunkan nilai *loss* dibandingkan dua metode lainnya.

Berdasarkan Tabel 4.7, metode LMGS menghasilkan nilai akurasi yang tinggi sejak *epoch* awal dan mempertahankannya secara konsisten hingga akhir pelatihan. Metode Adam menunjukkan peningkatan akurasi yang cukup cepat, namun masih berada di bawah atau setara dengan LMGS pada *epoch-epoch* yang diamati. Sebaliknya, GD mengalami peningkatan akurasi secara bertahap dan membutuhkan *epoch* yang lebih banyak untuk mencapai nilai yang setara. Dengan demikian, LMGS dapat disimpulkan sebagai metode yang paling unggul karena mampu mencapai akurasi tinggi secara lebih cepat dan stabil.

4.2.2.4 Evaluasi Model

Hasil *confusion matrix* dari model klasifikasi pada data *testing* dengan metode LMGS ditunjukkan pada Gambar 4.7.



Gambar 4.7 *Confusion Matriks* (0: Non Diabetes-Negatif, 1: Diabetes-Positif)

Hasil *confusion matrix* memberikan gambaran mengenai sejauh mana model mampu mengklasifikasikan data uji dengan benar. Pada kasus ini, label positif mengacu pada data pasien yang terindikasi diabetes, sedangkan label negatif menunjukkan pasien yang tidak mengidap diabetes. Sebanyak 30 data positif berhasil diprediksi dengan benar (*True Positive* = 30), sedangkan 11 data positif lainnya salah diprediksi sebagai negatif (*False Negative* = 11). Sementara itu, terdapat 14 data negatif yang keliru diprediksi sebagai positif (*False Positive* = 14), dan 61 data negatif berhasil dikenali dengan benar (*True Negative* = 61). Dari hasil tersebut, diperoleh metrik evaluasi:

1. Nilai Akurasi

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

$$= \frac{30 + 61}{30 + 61 + 14 + 11}$$

$$= 0{,}784$$

2. Nilai Presisi

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

$$= \frac{30}{30 + 14}$$

$$= 0{,}681$$

3. Nilai *Recall*

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

$$= \frac{30}{30 + 11}$$

$$= 0{,}731$$

4. Nilai *F1-Score*

$$F1\text{-}Score = 2 \times \frac{\text{presisi} \times recall}{\text{presisi} + recall}$$

$$= 2 \times \frac{0{,}681 \times 0{,}731}{0{,}681 + 0{,}731}$$

$$= 0{,}705$$

Untuk mengetahui kinerja metode LMGS secara lebih menyeluruh, dilakukan perbandingan dengan dua metode optimasi dasar, yaitu *Gradient Descent* (GD) dan Adam. Hasil perbandingan tercantum pada Tabel 4.8.

Tabel 4.8 Perbandingan Performa dengan Menggunakan Metode GS, Adam, dan LMGS pada Dataset *Pima Indian Diabetes*

Metode	Akurasi	Presisi	*Recall*	*F1-Score*
GD	0,758	0,644	0,707	0,674
Adam	0,758	0,644	0,707	0,674
LMGS	0,784	0,681	0,731	0,705

Perbandingan hasil numerik antara ketiga metode dapat dilihat pada Tabel 4.8. Nilai akurasi metode LMGS mencapai 0,784, lebih tinggi dibandingkan GD dan Adam yang sama-sama sebesar 0,758. Selain itu, nilai presisi dan *recall* pada LMGS juga meningkat, nilai presisi sebesar 0,681 menunjukkan bahwa dari seluruh prediksi positif (diabetes) yang dihasilkan model, sebanyak 68,1% merupakan prediksi yang benar mengenai pasien yang benar-benar mengidap diabetes. Sementara itu, nilai *recall* sebesar 0,731 mengindikasikan bahwa dari seluruh pasien yang benar-benar positif diabetes, model berhasil mengenali 73,1% di antaranya dengan tepat. Nilai *F1-Score* pada LMGS sebesar 0,705, lebih unggul dibandingkan dua metode dasar yang hanya mencapai 0,674, menunjukkan keseimbangan yang baik antara presisi dan *recall*.

Pada evaluasi ini, perhitungan presisi, *recall*, dan *F1-Score* tidak menggunakan nilai rata-rata antar kelas, karena model dikembangkan dengan pendekatan *binary classification* yang berfokus pada *single-class evaluation*. Dalam konteks diagnosis diabetes, kelas positif merupakan kelas utama yang harus dikenali secara akurat, karena kesalahan dalam mendeteksi pasien yang benar-benar mengidap diabetes dapat menimbulkan konsekuensi yang lebih serius dibandingkan kesalahan pada kelas negatif. Jika nilai rata-rata digunakan, performa model pada kelas positif dapat tersamarkan oleh kinerja pada kelas negatif, sehingga tidak mencerminkan kemampuan sebenarnya dalam mendeteksi pasien diabetes. Oleh karena itu, fokus evaluasi diarahkan pada kelas positif agar performa model dalam mengidentifikasi pasien yang mengidap diabetes dapat dinilai secara lebih spesifik dan akurat.

Secara keseluruhan, dapat disimpulkan bahwa metode *Levenberg–Marquardt* dengan *Gauss Seidel* (LMGS) memberikan performa terbaik dibandingkan metode optimasi dasar *Gradient Descent* dan Adam. Metode ini mampu menghasilkan *loss* yang lebih rendah, akurasi yang lebih tinggi, serta stabilitas pelatihan yang lebih baik. Dengan demikian, penerapan metode LMGS terbukti lebih efektif dan efisien dalam meningkatkan performa model pada dataset *Pima Indian Diabetes* dibandingkan kedua metode optimasi dasar tersebut.

4.3 Pembahasan

Bagian ini menguraikan hasil pelatihan model serta perbandingan performa antar metode optimasi. Pembahasan difokuskan pada pola *loss* dan akurasi yang muncul, kemudian dilanjutkan dengan penilaian efektivitas setiap metode berdasarkan hasil pengujian.

4.3.1 Analisis Hasil

Hasil eksperimen menunjukkan bahwa metode LMGS memiliki performa klasifikasi yang lebih baik dibandingkan GD dan Adam. Dari kurva pembelajaran terlihat bahwa LMGS menurunkan *loss* secara tajam dan stabil sejak iterasi awal hingga akhir *epoch*, sementara GD menurun lambat dan Adam mengalami fluktuasi awal. Kecepatan konvergensi LMGS jauh lebih tinggi, hanya memerlukan 10 *epoch* untuk mencapai akurasi pelatihan $\geq$ 0,75, dibandingkan Adam (227 *epoch*) dan GD (3518 *epoch*). Evaluasi kinerja menggunakan *confusion matrix* menunjukkan bahwa LMGS mampu mengklasifikasikan data positif dengan lebih tepat, menghasilkan akurasi 0,784, presisi 0,681, *recall* 0,731, dan *F1-Score* 0,705, lebih tinggi dibandingkan GD dan Adam yang hanya mencapai akurasi 0,758 dan *F1-Score* 0,674. Hasil ini menegaskan bahwa LMGS tidak hanya mempercepat proses konvergensi, tetapi juga meningkatkan kemampuan generalisasi model dalam klasifikasi biner, sehingga metode ini lebih efektif dan andal untuk penerapan regresi logistik.

Secara keseluruhan, berdasarkan seluruh tahapan yang telah dilakukan, menunjukkan bahwa kombinasi *metode Levenberg–Marquardt* dan *Gauss–Seidel (LMGS)* terbukti efektif dalam meningkatkan performa model regresi logistik pada kasus prediksi diabetes menggunakan dataset *Pima Indian Diabetes*. Pendekatan ini mampu menghasilkan akurasi tinggi, konvergensi cepat, serta stabilitas komputasi yang baik dibandingkan dengan metode optimasi konvensional seperti *Gradient Descent* dan Adam.

4.3.2 Perbandingan dengan Penelitian Sebelumnya

Evaluasi kinerja model dalam penelitian ini dilakukan dengan membandingkannya terhadap hasil-hasil penelitian terdahulu yang menggunakan dataset *Pima Indian Diabetes*. Ringkasan perbandingan akurasi, presisi, *recall*, *F1-Score*, dan beberapa metrik lain disajikan pada Tabel 4.9 sesuai dengan metode dan studi terkait.

Tabel 4.9 Perbandingan dengan Penelitian Sebelumnya

Author	Metode	Akurasi	Presisi	*Recall*	*F1-Score*
Penelitian ini	LMGS	0,784	0,681	0,731	0,705
(Arrohman & Fatah, 2024)	KNN	0,701	0,591	0,481	-
(Kumar dkk., 2021)	SVM	0,665	0,681	0,665	0,671
	NB	0,736	0,745	0,736	0,739

Pada penelitian ini, metode LMGS menunjukkan performa yang cukup baik dengan akurasi 0,784, presisi 0,681, *recall* 0,731, dan *F1-Score* 0,705. Nilai *recall* yang relatif tinggi menunjukkan bahwa model mampu mendeteksi sebagian besar data positif dengan baik, meskipun akurasinya belum setinggi beberapa metode pembanding. Hasil ini menandakan bahwa LMGS bekerja secara seimbang antara ketepatan prediksi dan kemampuan mengenali kasus positif, sehingga masih relevan digunakan dalam konteks medis yang membutuhkan sensitivitas tinggi.

Jika dibandingkan dengan penelitian Arrohman & Fatah, (2024) yang menggunakan metode KNN dengan akurasi 0,701, presisi 0,591, dan *recall* 0,481, metode LMGS terbukti memberikan peningkatan yang signifikan di seluruh metrik evaluasi. Hal ini menunjukkan bahwa pendekatan LMGS lebih stabil dan efisien dalam menjaga keseimbangan antara akurasi, presisi, dan *recall* dibandingkan metode dasar seperti KNN. Dengan demikian, metode LMGS terbukti lebih unggul dibandingkan dengan KNN.

Selain itu, jika dibandingkan dengan penelitian (Kumar dkk., 2021) yang menggunakan metode SVM dan *Naïve Bayes*, performa LMGS juga menunjukkan hasil yang baik. Metode SVM hanya memperoleh akurasi 0,665 dan *F1-Score* 0,671, sehingga masih lebih rendah dibandingkan LMGS. Metode *Naïve Bayes* memiliki akurasi 0,736 dan *F1-Score* 0,739, namun metode ini memiliki asumsi bahwa setiap fitur saling independen, yang sering tidak sesuai dengan karakteristik data medis (Bishop, 2006). Hal ini membuat performanya bisa kurang stabil. Sebaliknya, LMGS mampu menangani hubungan antar fitur dengan lebih baik melalui pendekatan optimasi berbasis gradien dan Hessian, sehingga memberikan hasil yang lebih konsisten.

Secara keseluruhan, LMGS tidak hanya lebih baik dari metode sederhana seperti KNN, tetapi juga mampu bersaing dengan metode lain seperti SVM dan *Naïve Bayes*. Keunggulan LMGS terlihat dari keseimbangan nilai *recall* dan *F1-*

Score, yang penting dalam bidang medis karena berkaitan dengan kemampuan model dalam mendeteksi kasus positif dengan lebih baik.

4.4 Kajian terkait Sikap Profesional terhadap Hasil Penelitian dalam Optimasi

Dalam penelitian ini, penulis tidak hanya berfokus pada penerapan metode optimasi secara matematis, tetapi juga melihat bagaimana nilai-nilai etika kerja dapat tercermin dalam proses penelitian. Salah satu nilai tersebut adalah sifat profesionalitas atau *itqan*, yang berasal dari hadits yang menyatakan bahwa "Sesungguhnya Allah mencintai seseorang yang apabila bekerja, mengerjakannya secara *itqan* (professional)". (HR. Thabrani) (pa-sungairaya.go.id, 2024). Hadits tersebut menyebutkan bahwa Allah mencintai seseorang yang ketika bekerja, ia melakukannya dengan baik dan profesional. Nilai ini relevan untuk menjadi landasan sikap dalam menjalankan rangkaian proses penelitian, termasuk pada tahap optimasi.

Sikap profesional tersebut tercermin dalam pelaksanaan optimasi menggunakan metode LMGS. Penerapan metode ini dilakukan melalui dua pendekatan, yaitu perhitungan manual untuk memastikan kebenaran konsep matematis dan implementasi komputasional agar metode dapat dijalankan secara efisien pada dataset. Kedua pendekatan ini menuntut ketelitian, karena setiap bagian dari metode harus dipastikan berjalan sesuai prosedur. Selain itu, proses iteratif LMGS dijalankan dengan sungguh-sungguh sampai model mencapai hasil yang stabil, ditandai dengan penurunan *loss* dan peningkatan akurasi secara konsisten. Penggunaan *Gauss Seidel* di dalam metode turut membantu mempercepat proses pencarian solusi sehingga hasil optimasi lebih terarah. Penelitian ini juga membandingkan performa LMGS dengan metode lain seperti *Gradient Descent* dan Adam untuk memastikan bahwa metode yang digunakan benar-benar memberikan hasil terbaik.

Jika dibandingkan dengan penelitian sebelumnya, metode LMGS berhasil mengoptimalkan regresi logistik secara lebih efisien dibandingkan metode lain seperti KNN, SVM, dan NB, karena memiliki keseimbangan yang lebih baik antara akurasi, presisi, *recall* dan *F1-Score*.

Secara keseluruhan, sikap profesional, teliti, serius, dan bersungguh-sungguh dalam menerapkan LMGS baik melalui perhitungan manual maupun implementasi komputasional berpengaruh nyata terhadap kualitas hasil penelitian. Hal ini terlihat dari nilai *loss* yang lebih rendah, proses konvergensi yang lebih cepat, dan akurasi yang lebih tinggi dibandingkan beberapa metode dasar lainnya. Dengan demikian, nilai *itqan* yang terdapat dalam hadits tersebut tercermin dalam keseluruhan proses optimasi dan memperkuat keberhasilan penggunaan metode LMGS dalam penelitian ini, sehingga temuan ini semakin menegaskan pentingnya nilai *itqan* dalam memastikan pelaksanaan optimasi berjalan maksimal.

BAB V
PENUTUP

5.1 Kesimpulan

Berdasarkan rumusan masalah dan hasil pembahasan pada bab sebelumnya, diperoleh beberapa kesimpulan penting sebagai berikut:

1. Optimalisasi regresi logistik menggunakan pendekatan *Levenberg–Marquardt* dan *Gauss–Seidel* (LMGS) terbukti efektif dalam penyelesaian masalah klasifikasi biner karena kombinasi kedua metode ini mampu mempercepat proses konvergensi serta menghasilkan nilai *loss* lebih rendah dan akurasi lebih tinggi dibandingkan metode optimasi lain seperti *Gradient Descent* dan Adam, dengan proses pelatihan yang lebih stabil dan cepat mencapai titik optimal. LMGS hanya membutuhkan 10 *epoch* dengan waktu komputasi 5,76 milidetik untuk mencapai akurasi pelatihan $\geq 0{,}75$ jauh lebih cepat dibandingkan Adam (227 *epoch*) dan GD (3518 *epoch*). Dengan demikian, pendekatan LMGS terbukti lebih efisien, stabil, dan akurat dalam proses optimisasi parameter regresi logistik untuk kasus klasifikasi biner. Dengan demikian, pendekatan LMGS efektif dalam mengoptimalkan regresi logistik untuk penyelesaian klasifikasi biner, khususnya pada kasus prediksi diabetes menggunakan dataset *Pima Indian Diabetes*.
2. Performa metode LMGS dalam klasifikasi biner menggunakan regresi logistik menunjukkan hasil terbaik di antara metode pembanding. Nilai akurasi mencapai 0,784, presisi 0,681, *recall* 0,731, dan *F1-score* 0,705, yang berarti model memiliki keseimbangan yang baik antara ketepatan prediksi dan kemampuan mengenali data positif. Jika dibandingkan dengan penelitian

sebelumnya, metode *LMGS* berhasil mengoptimalkan regresi logistik secara lebih efisien dibandingkan metode lain seperti KNN, SVM, dan NB, karena memiliki keseimbangan yang lebih baik antara akurasi, presisi, *recall* dan *F1-Score*.

5.2 Saran untuk Penelitian Lanjutan

Berdasarkan hasil penelitian, beberapa saran untuk pengembangan penelitian selanjutnya adalah sebagai berikut:

1. Penelitian selanjutnya disarankan untuk mengombinasikan metode LMGS dengan algoritma pembelajaran modern seperti *ensemble learning* (misalnya Random Forest, XGBoost, atau LightGBM) guna memperoleh hasil yang lebih akurat dan stabil.
2. Penelitian selanjutnya disarankan untuk mengaplikasikan metode LMGS pada dataset dengan karakteristik berbeda (misalnya jumlah fitur yang lebih banyak, data berukuran besar, atau distribusi kelas yang tidak seimbang) untuk menilai konsistensi dan generalisasi metode ini.
3. Penelitian selanjutnya dapat membandingkan LMGS dengan algoritma optimasi lain seperti *BFGS***,** *RMSProp*, atau bahkan metode hybrid, agar diperoleh pemahaman yang lebih luas mengenai keunggulan relatif masing-masing pendekatan, baik dari sisi akurasi, presisi, *recall*, dan *F1-score*.

DAFTAR PUSTAKA

Arrohman, S., & Fatah, Z. (2024). Prediksi Diabetes Menggunakan Algoritma Klasifikasi K-Nearest Neighbors (K-NN) pada Perempuan Indian Pima. *Gudang Jurnal Multidisiplin Ilmu , 2*(10), 220–226.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. New York, NY: Springer.

Burden, Richard L., Faires, & J. Douglas et al. (2015). *Numerical Analysis*. Cengage Learning

Damanik, M. Z., Mawadda, M. A., & Novita, D. (2024). Ayat al-quran dan hadis hakikat ilmu. *At-Tarbiyah: Jurnal Penelitian Dan Pendidikan Agama Islam*, *2*, 601–608.

Hagan, Martin T., Menhaj, & Mohammad B. et al. (1994). Training Feedforward Networks with the Marquardt Algorithm. *IEEE Transactions on Neural Networks, 5*(6), 989–993. https://doi.org/10.1109/72.329697

Han, J., Kamber, M., & Pei, J. (2022). *Data Mining Concepts and Techniques (4th ed)*. Morgan Kaufmann.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *Springer Series in Statistics*. Springer.

Indriani, V. (2011). Perbandingan Metode Gauss Newton Dan Metode Marquardt Dalam Menaksir Parameter Binary Logistic Regression (Doctoral dissertation, Universitas Brawijaya).

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

Kementerian Agama. (2022). Qur'an Kemenag

Khan, N., Gaurav, D., & Kandl, T. (2013). Performance evaluation of Levenberg-Marquardt technique in error reduction for diabetes condition classification. *Procedia Computer Science*, *18*, 2629–2637. https://doi.org/10.1016/j.procs.2013.05.455

Koshy, Soumya Sara, & Jani Anbarasi, L. et al. (2024). LMHistNet: Levenberg-Marquardt Based Deep Neural Network for Classification of Breast Cancer Histopathological Images. *IEEE Access*, *12*(March), 52051–52066. https://doi.org/10.1109/ACCESS.2024.3385011

Kumar, S., Kumar, S., Kumar, K., & Abhisekh, P. A. (2021). *Prediction of Diabetes in Females of Pima Indian Heritage : A Complete Supervised Learning*

Approach. *12*(10), 3074–3084.

Levenberg, K. (1944). *A METHOD FOR THE SOLUTION OF CERTAIN NON-LINEAR*. *1*, 164–168.

Mohamad, N., Zaini, F., Johari, A., Yassin, I., & Zabidi, A. (2010). *Comparison between Levenberg-Marquardt and Scaled Conjugate Gradient training algorithms for Breast Cancer Diagnosis using MLP*. *Proceedings of the 6th International Colloquium on Signal Processing & its Applications (CSPA)*, 1–7. https://doi.org/10.1109/cspa.2010.5545325

Pa-sungairaya.go.id. (2024). Mahkamah Agung Republik Indonesia Pengadilan Agama Sungai Raya . Diambil dari https://www.pa-sungairaya.go.id/wp/hadits-keutamaan-bekerja-dengan-baik-sungguh-sungguh-dan-profesional/

Scott, A. J., Hosmer, D. W., & Lemeshow, S. (1991). Applied Logistic Regression. *Biometrics*, *47*(4), 1632. https://doi.org/10.2307/2532419

Sumarauw, S J A, Manado, & Universitas Negeri et al. (2018). Algoritma Pelatihan Levenberg-Marquardt Backpropagation Artificial Neural Network Untuk Data Time Series. *Frontiers: Jurnal Sains Dan Teknologi*, *1*, 213–222. https://doi.org/10.36412/frontiers/001035e1/agustus201801.10

LAMPIRAN

Lampiran 1 Data *Pima Indian Diabetes*

Lampiran 2 *Script Phyton*

```
# ==========================================
# Logistic Regression from scratch + GD, Adam, LM+GS (improved)
# with Train/Validation/Test results & plots
# ==========================================
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score
from sklearn.metrics import confusion_matrix

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

# ---------------------------------------------------
# A. Setup
# ---------------------------------------------------
RNG_SEED = 42
np.random.seed(RNG_SEED)

# ---------------------------------------------------
# B. Load dataset (Pima Indians Diabetes)
# ---------------------------------------------------
df = pd.read_csv("/kaggle/input/pima-indians-diabetes-
database/diabetes.csv")
X = df.drop("Outcome", axis=1).values
y = df["Outcome"].values.reshape(-1, 1)

# Cek distribusi kelas target
plt.figure(figsize=(6, 4))
sns.countplot(x='Outcome', data=df)
plt.title('Distribusi Kelas Target (0: Non-diabetes, 1:
Diabetes)')
plt.show()
```

```
# Standarisasi
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split: 70% train, 15% val, 15% test
X_train, X_temp, y_train, y_temp = train_test_split(
    X_scaled, y, test_size=0.3, stratify=y,
random_state=RNG_SEED
)
X_val, X_test, y_val, y_test = train_test_split(
    X_temp, y_temp, test_size=0.5, stratify=y_temp,
random_state=RNG_SEED
)

# -------------------------------------------------
# C. Fungsi bantu
# -------------------------------------------------
def sigmoid(z):
    return 1 / (1 + np.exp(-z))

def bce_loss(a, X, y, lambda_reg=0.0):
    p = sigmoid(X @ a)
    eps = 1e-8
    loss = -np.mean(y * np.log(p + eps) + (1 - y) * np.log(1 - p
+ eps))
    loss += lambda_reg * np.sum(a**2) / (2 * len(y))
    return loss, p

def metrics_from_prob(y_true, p):
    y_pred = (p >= 0.5).astype(int)
    acc = np.mean(y_pred == y_true)
    TP = np.sum((y_true == 1) & (y_pred == 1))
    TN = np.sum((y_true == 0) & (y_pred == 0))
    FP = np.sum((y_true == 0) & (y_pred == 1))
    FN = np.sum((y_true == 1) & (y_pred == 0))
    return acc, TP, TN, FP, FN

# -------------------------------------------------
# D. Gradient Descent
# -------------------------------------------------

def fit_gd(X, y, X_val, y_val, lr=0.01, epochs=50):
    a = np.zeros((X.shape[1], 1))
    hist = {"train_loss": [], "val_loss": [], "train_acc": [],
"val_acc": []}

    for epoch in range(epochs):
        p = sigmoid(X @ a)
        grad = X.T @ (p - y) / len(y)
        a -= lr * grad

        Ltr, p_tr = bce_loss(a, X, y)
        Lva, p_va = bce_loss(a, X_val, y_val)
        acc_tr, *_ = metrics_from_prob(y, p_tr)
        acc_va, *_ = metrics_from_prob(y_val, p_va)

        hist["train_loss"].append(Ltr)
        hist["val_loss"].append(Lva)
```

```
        hist["train_acc"].append(acc_tr)
        hist["val_acc"].append(acc_va)


    return a, hist

# --------------------------------------------------
# E. Adam Optimizer
# --------------------------------------------------
def fit_adam(X, y, X_val, y_val, lr=0.01, beta1=0.9,
beta2=0.999, eps=1e-8, epochs=50):
    a = np.zeros((X.shape[1], 1))
    m, v = np.zeros_like(a), np.zeros_like(a)
    hist = {"train_loss": [], "val_loss": [], "train_acc": [],
"val_acc": []}

    for t in range(1, epochs + 1):
        p = sigmoid(X @ a)
        grad = X.T @ (p - y) / len(y)
        m = beta1 * m + (1 - beta1) * grad
        v = beta2 * v + (1 - beta2) * (grad ** 2)
        m_hat = m / (1 - beta1 ** t)
        v_hat = v / (1 - beta2 ** t)
        a -= lr * m_hat / (np.sqrt(v_hat) + eps)

        Ltr, p_tr = bce_loss(a, X, y)
        Lva, p_va = bce_loss(a, X_val, y_val)
        acc_tr, *_ = metrics_from_prob(y, p_tr)
        acc_va, *_ = metrics_from_prob(y_val, p_va)

        hist["train_loss"].append(Ltr)
        hist["val_loss"].append(Lva)
        hist["train_acc"].append(acc_tr)
        hist["val_acc"].append(acc_va)
    return a, hist

# --------------------------------------------------
# F. Levenberg-Marquardt + Gauss-Seidel (improved, anti-
overfitting)
# --------------------------------------------------
def fit_lm_gs_with_history(
    X, y, X_val, y_val,
    lam0=1.0, max_outer=50,
    tol=1e-6, lambda_reg=0.01
):
    a = np.zeros((X.shape[1], 1))
    lam = lam0
    hist = {"train_loss": [], "val_loss": [], "train_acc": [],
"val_acc": []}

    best_val_loss = np.inf
    bad_epochs = 0  # untuk early stopping

    for k in range(max_outer):
        # === Forward pass
        p = sigmoid(X @ a)
        r = y - p
        J = X * (p * (1 - p))
        H = J.T @ J + lam * np.eye(X.shape[1])
```

```
        g = J.T @ r

        # === Gauss-Seidel iteratif untuk Δa
        delta_a = np.zeros_like(a)
        for j in range(len(a)):
            s1 = np.dot(H[j, :j], delta_a[:j])
            s2 = np.dot(H[j, j + 1:], delta_a[j + 1:])
            delta_a[j] = (g[j] - s1 - s2) / H[j, j]

        a_new = a + delta_a

        # === Evaluasi loss dgn regularisasi
        L_old, _ = bce_loss(a, X, y, lambda_reg=lambda_reg)
        L_new, _ = bce_loss(a_new, X, y, lambda_reg=lambda_reg)

        # === Update parameter dan λ adaptif
        if L_new < L_old:
            a = a_new
            lam = max(lam / 2, 1e-3)
        else:
            lam = min(lam * 2, 1e3)

        # === Simpan histori train/val
        Ltr, p_tr = bce_loss(a, X, y, lambda_reg=lambda_reg)
        Lva, p_va = bce_loss(a, X_val, y_val,
lambda_reg=lambda_reg)
        acc_tr, *_ = metrics_from_prob(y, p_tr)
        acc_va, *_ = metrics_from_prob(y_val, p_va)
        hist["train_loss"].append(Ltr)
        hist["val_loss"].append(Lva)
        hist["train_acc"].append(acc_tr)
        hist["val_acc"].append(acc_va)


    return a, hist

# -------------------------------------------------
# G. Train all models
# -------------------------------------------------
a_gd, hist_gd = fit_gd(X_train, y_train, X_val, y_val, lr=0.01,
epochs=50)
a_adam, hist_adam = fit_adam(X_train, y_train, X_val, y_val,
lr=0.01, epochs=50)
a_lm, hist_lm = fit_lm_gs_with_history(X_train, y_train, X_val,
y_val, lam0=1.0, max_outer=50)

# -------------------------------------------------
# H. Plot Training & Validation history
# -------------------------------------------------
plt.figure(figsize=(8, 5))
plt.plot(hist_gd["train_loss"], label="GD Train")
plt.plot(hist_gd["val_loss"], "--", label="GD Val")
plt.plot(hist_adam["train_loss"], label="Adam Train")
plt.plot(hist_adam["val_loss"], "--", label="Adam Val")
plt.plot(hist_lm["train_loss"], label="LM+GS Train")
plt.plot(hist_lm["val_loss"], "--", label="LM+GS Val")
plt.title("Loss per Epoch")
plt.xlabel("Epoch")
plt.ylabel("Loss")
```

```
plt.legend()

plt.figure(figsize=(8, 5))
plt.plot(hist_gd["train_acc"], label="GD Train")
plt.plot(hist_gd["val_acc"], "--", label="GD Val")
plt.plot(hist_adam["train_acc"], label="Adam Train")
plt.plot(hist_adam["val_acc"], "--", label="Adam Val")
plt.plot(hist_lm["train_acc"], label="LM+GS Train")
plt.plot(hist_lm["val_acc"], "--", label="LM+GS Val")
plt.title("Accuracy per Epoch")
plt.xlabel("Epoch")
plt.ylabel("Accuracy")
plt.legend()
plt.ylim(0.6, 0.775)
plt.tight_layout()
plt.show()

# ------------------------------------------------
# I. Evaluate on Test set
# ------------------------------------------------
for name, a in [("GD", a_gd), ("Adam", a_adam), ("LM+GS",
a_lm)]:
    loss_test, p_test = bce_loss(a, X_test, y_test)
    acc_test, *_ = metrics_from_prob(y_test, p_test)
    print(f"{name} → Test Accuracy: {acc_test:.4f} | Test Loss:
{loss_test:.4f}")

# ------------------------------------------------
# J. Evaluasi Metrik Lengkap (Accuracy, Precision, Recall, F1-
score)
# ------------------------------------------------

# Kumpulkan hasil prediksi probabilitas dari model
y_pred_gd = (sigmoid(X_test @ a_gd) >= 0.5).astype(int)
y_pred_adam = (sigmoid(X_test @ a_adam) >= 0.5).astype(int)
y_pred_lm = (sigmoid(X_test @ a_lm) >= 0.5).astype(int)

# Simpan dalam dictionary agar mudah di-loop
optimizers = {
    "GD": y_pred_gd,
    "Adam": y_pred_adam,
    "LM+GS": y_pred_lm
}

# Hitung metrik untuk tiap optimizer
print("=== Evaluasi Metrik pada Data Test ===")
for name, y_pred in optimizers.items():
    acc = accuracy_score(y_test, y_pred)
    prec = precision_score(y_test, y_pred, zero_division=0)
    rec = recall_score(y_test, y_pred, zero_division=0)
    f1 = f1_score(y_test, y_pred, zero_division=0)

    print(f"\n {name}")
    print(f"Accuracy : {acc:.4f}")
    print(f"Precision: {prec:.4f}")
    print(f"Recall   : {rec:.4f}")
    print(f"F1-score : {f1:.4f}")

# ------------------------------------------------
```

```
# K. Tabel Komparasi Metrik Semua Optimizer
# --------------------------------------------------

# Simpan hasil metrik ke dalam list of dict
results = []
for name, y_pred in optimizers.items():
    acc = accuracy_score(y_test, y_pred)
    prec = precision_score(y_test, y_pred, zero_division=0)
    rec = recall_score(y_test, y_pred, zero_division=0)
    f1 = f1_score(y_test, y_pred, zero_division=0)
    results.append({
        "Optimizer": name,
        "Accuracy": round(acc, 4),
        "Precision": round(prec, 4),
        "Recall": round(rec, 4),
        "F1-Score": round(f1, 4)
    })

# Buat DataFrame hasil evaluasi
df_metrics = pd.DataFrame(results)

# Urutkan berdasarkan akurasi tertinggi
df_metrics = df_metrics.sort_values(by="Accuracy",
ascending=False).reset_index(drop=True)

# Tampilkan tabel
print("\n Tabel Komparasi Metrik (Data Test):")
display(df_metrics)

# --------------------------------------------------
# L. Tampilkan Confusion Matrix Semua Optimizer
# --------------------------------------------------

for name, y_pred in optimizers.items():
    print(f"\n Confusion Matrix untuk {name}")

    # Hitung confusion matrix
    cm = confusion_matrix(y_test, y_pred)

    # Plot confusion matrix
    plt.figure(figsize=(5, 4))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
cbar=False)
    plt.title(f"Confusion Matrix - {name}")
    plt.xlabel("Predicted Label")
    plt.ylabel("Actual Label")
    plt.tight_layout()
    plt.show()

# --------------------------------------------------
# E. Rangkuman: epoch & waktu untuk mencapai target (train)
# --------------------------------------------------
rows = []
for name, H in [("GD", hist_gd), ("Adam", hist_adam), ("LM+GS",
hist_lm)]:
    rows.append({
        "Method": name,
        "Epoch_to_≥0.75": H["epoch_to_target"],
```

```
        "Time_to_≥0.75_ms": None if H["time_to_target_ms"] is
     None else round
(H["time_to_target_ms"], 2),
        "Reached?": bool(H["epoch_to_target"] is not None),
        "Avg_epoch_time_ms":
     round(float(np.mean(H["epoch_time_ms"])), 3) if l
en(H["epoch_time_ms"]) else None
    })
df_target = pd.DataFrame(rows).sort_values(by=["Reached?",
"Epoch_to_≥0.75"], ascending=[False, True])
print("\n=== Epoch & Waktu sampai akurasi TRAINING ≥ 0.75 ===")
print(df_target.to_string(index=False))
```

RIWAYAT HIDUP

Arida Nur Baizuriana, lahir di Wonogiri pada 4 Mei 2003. Penulis merupakan anak pertama dari Ayah Sarimo, S.E dan Ibu Darmini serta kakak dari Almh. Nadia Firna Nur Zuriana, Ardia Zaky Yudistira, Arniza Zahra Asilah, Azhariah Inara Nuha Zahira, Arfirza Irsyad Zahir Firdaus, Aziva Chelya Auristella. Penulis telah menempuh pendidikan dari sekolah dasar di SDN 1 Sempukerep yang lulus tahun 2015. Kemudian penulis melanjutkan pendidikan sekolah menengah pertama di SMPN 2 Sidoharjo dan lulus pada tahun 2018, selanjutnya penulis melanjutkan pendidikan sekolah menengah atas di SMAN 1 Sidoharjo dan lulus pada tahun 2021. Pada tahun yang sama, penulis melanjutkan pendidikan tinggi di Universitas Islam Negeri Maulana Malik Ibrahim Malang. Penulis juga telah melaksanakan program Kuliah Kerja Nyata yang dilaksanakan di Desa Sawahan, Turen pada tahun 2024 dan berperan aktif sebagai bendahara yang mengatur keuangan yang dibutuhkan selama kegiatan berlangsung. Penulis juga telah melaksanakan kegiatan Program Kerja Lapangan (PKL) di Kantor Pos Malang pada bulan Juni-Juli di bagian jasa keuangan dan marketing yang melaksanakan tugas seperti promosi produk dan layanan, dan membantu mengolah data penerima dana bantuan sosial. Selanjutnya selama masa PKL penulis juga ditempatkan pada bagian pembayaran pensiunan Asabari dan Taspen, dengan tugas yang dilaksanakan yaitu melayani pencairan dana pensiun, verifikasi data pensiun, meng-*input* transaksi, dan melakukan pelayanan administratif.

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	:	Arida Nur Baizuriana
NIM	:	210601110084
Fakultas / Jurusan	:	Sains dan Teknologi / Matematika
Judul Skripsi	:	Optimasi Regresi Logistik Menggunakan Pendekatan *Levenberg Marquardt* dan *Gauss Seidel* untuk Penyelesaian Klasifikasi Biner
Pembimbing I	:	Dr. Mohammad Jamhuri, M.Si.
Pembimbing II	:	Dr. Fachrur Rozi, M.Si.

No	Tanggal	Hal	Tanda Tangan
1.	16 April 2025	Konsultasi Bab I, II, dan III	1.
2.	22 April 2025	Konsultasi Bab I, II, dan III	2.
3.	29 April 2025	Konsultasi Bab I, II, dan III	3.
4.	5 Mei 2025	ACC Bab I, II, dan III	4.
5.	8 Mei 2025	Konsultasi Kajian Agama Bab I dan II	5.
6.	9 Mei 2025	Konsultasi Kajian Agama Bab I dan II	6.
7.	14 Mei 2025	Konsultasi Kajian Agama Bab I dan II	7.
8.	14 Mei 2025	ACC Kajian Agama Bab I dan II	8.
9.	14 Mei 2025	ACC Seminar Proposal	9.
10.	3 Oktober 2025	Konsultasi Revisi Seminar Proposal	10.
11.	9 Oktober 2025	Konsultasi Bab IV	11.
12.	4 November 2025	Konsultasi Bab IV	12.
13.	17 November 2025	Konsultasi Bab IV dan V	13.
14.	19 November 2025	ACC Bab IV dan V	14.
15.	20 November 2025	Konsultasi Kajian Agama Bab IV	15.

16.	21 November 2025	Konsultasi Kajian Agama Bab IV	16.
17.	24 November 2025	ACC Kajian Agama Bab IV	17.
18.	24 November2025	ACC Seminar Hasil	18.
19.	4 Desember 2025	Konsultasi Revisi Seminar Hasil	19.
20.	16 Desember 2025	Sidang Skripsi	20.
21	23 Desember 2025	ACC Keseluruhan	21.

Malang, 23 Desember 2025
Mengetahui,
Ketua Program Studi Matematika

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