

**OPTIMASI REGRESI LOGISTIK MENGGUNAKAN
GAUSS-NEWTON DAN DEKOMPOSISI QR UNTUK
PENYELESAIAN KLASIFIKASI BINER**

SKRIPSI

**OLEH:
Alvin Aura Einst
NIM. 210601110012**



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

**OPTIMASI REGRESI LOGISTIK MENGGUNAKAN
GAUSS-NEWTON DAN DEKOMPOSISI QR 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:
Alvin Aura Einst
NIM. 210601110012**

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

OPTIMASI REGRESI LOGISTIK MENGGUNAKAN GAUSS-NEWTON DAN DEKOMPOSISI QR UNTUK PENYELESAIAN KLASIFIKASI BINER

SKRIPSI

Oleh
Alvin Aura Einst
NIM. 210601110012

Telah Disetujui Untuk Diuji

Malang, 11 Desember 2025

Dosen Pembimbing I

Dosen Pembimbing II



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



Dr. Ach. Nachichuddin, M.A.
NIP. 19730705 200003 1 002

Mengetahui,
Ketua Program Studi Matematika



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

OPTIMASI REGRESI LOGISTIK MENGGUNAKAN GAUSS-NEWTON DAN DEKOMPOSISI QR UNTUK PENYELESAIAN KLASIFIKASI BINER


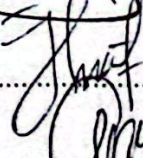


SKRIPSI

Oleh
Alvin Aura Einst
NIM. 210601110012

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

Tanggal 22 Desember 2025

Ketua Penguji	: Hisyam Fahmi, M.Kom.
Anggota Penguji 1	: Juhari, M.Si.
Anggota Penguji 2	: Dr. Mohammad Jamhuri, M.Si.
Anggota Penguji 3	: Dr. Ach. Nachichuddin, M.A.


.....

.....

.....

.....

Mengetahui,
Ketua Program Studi Matematika



Dr. Fachrudin Rozi, M.Si.
NIM. 272008011012

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini:

Nama : Alvin Aura Einst
NIM : 210601110012
Program Studi : Matematika
Fakultas : Sains dan Teknologi
Judul Skripsi : Optimasi Regresi Logistik Menggunakan Gauss-Newton
dan Dekomposisi Qr untuk Penyelesaian Klasifikasi Biner

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya sendiri. Bukan merupakan pengambilan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan dan pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila di kemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 22 Desember 2025

Yang Membuat Pernyataan



Alvin Aura Einst

NIM.210601110012

MOTO

“Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya”
(QS. Al-Baqarah : 286)

"Hatiku tenang karena mengetahui bahwa apa yang melewatkanmu tidak akan pernah menjadi takdirku, dan apa yang ditakdirkan untukmu tidak akan pernah melewatkanmu”
(Umar bin Khattab)

Setetes keringat ibuku yang keluar, ada seribu langkahku untuk maju

PERSEMBAHAN

Dengan penuh rasa syukur kepada Allah SWT, karya ini kupersembahkan untuk Ibu, perempuan luar biasa yang menjadi alasan saya kuat sampai hari ini. Dalam setiap lelah, Ibu hadir memberi pelukan; dalam setiap takut, Ibu hadir memberi keyakinan; dan dalam setiap jatuh, Ibu hadir memberi doa. Terima kasih, Ibu, untuk cinta tanpa batas yang tak pernah pudar. Semoga Ibu selalu sehat dan diberi umur panjang agar saya bisa terus membanggakan Ibu. Untuk diri yang tak menyerah, walau dunia berkali-kali menawarkan pelarian yang menipu. Terima kasih karena terus menjaga hati dan setia pada jalan Tuhan. Saat segala kepedihan menguji, engkau memilih untuk tetap berdiri. Hari ini adalah penghargaan atas ketegaranmu yang tak pernah hilang.

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

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 Gauss-Newton dan Dekomposisi QR 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 keinspirasian dari Rasulullah senantiasa mengiringi langkah-langkah peneliti dalam menggapai kesuksesan dan kebermanfaatan.

Skripsi ini disusun sebagai salah satu syarat untuk menyelesaikan studi Strata1 (S-1) 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., selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dr. H. Agus Mulyono, M.Si., selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang yang telah memberikan ilmu, wawasan, saran, solusi, dan nasihat yang sangat membantu untuk menyelesaikan skripsi ini.
3. Dr. Fachrur Rozi, M.Si., selaku Ketua Program Studi Matematika, Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Bapak Hisyam Fahmi, M.Kom., selaku Ketua penguji yang telah memberikan ilmu, saran dan nasihat yang sangat membantu untuk menyelesaikan skripsi ini.
5. Bapak Juhari, M.Si., selaku Anggota Penguji 1 yang telah memberikan ilmu, saran dan nasihat yang sangat membantu untuk menyelesaikan skripsi ini.

6. Bapak Dr. Mohammad Jamhuri, M.Si, selaku dosen pembimbing I yang telah memberikan ilmu, saran dan nasihat yang sangat membantu untuk menyelesaikan skripsi ini.
7. Bapak Ach. Nasichuddin, M.A, selaku dosen pembimbing II yang telah memberikan ilmu, saran dan nasihat yang sangat membantu untuk menyelesaikan skripsi ini.
8. Seluruh dosen Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.
9. Pintu surgaku, separuh nyawaku, dan sosok yang penulis jadikan panutan yaitu Ibu tercinta Siswati, orang yang sangat amat berjasa dalam hidup saya. Orang yang selalu mengusahakan anak pertamanya di tengah kondisi keluarga yang sering kali sulit, Ibu tetap berusaha sekuat tenaga agar saya, sebagai anak pertama, dapat mengejar pendidikan setinggi-tingginya. Terima kasih atas setiap pengorbanan, kerja keras, dan doa yang tidak pernah berhenti Ibu berikan. Apa yang saya capai hari ini adalah hadiah dari keteguhan dan cinta Ibu yang tidak ternilai.
10. Seluruh keluarga tercinta kakek, nenek, adik laki-laki penulis, om, tante dan seluruh sepupu penulis yang selalu memberi dukungan moral maupun materi, doa, motivasi, serta kasih sayang yang luar biasa.
11. Kepada orang yang tak kalah penting kehadirannya, Yaya, Bintang, Fara, Aristya selaku teman terdekat penulis di Kota Tuban yang menjadi tempat berbagi cerita keluh kesah dan memberikan dukungan untuk penulis.
12. Sherly, Lorensa, Yusma, Dhifa, Luthfia, Silla selaku teman terdekat penulis pada masa perkuliahan. Terimakasih telah menjadi rumah kedua bagi penulis selama melewati kehidupan di perantauan.
13. Seluruh teman pendakian Gunung Rinjani, Lawu, Buthak, Arjuno, Penanggungan, Budug Asu, Kawi, Tanggung, Kelud, Pundak, Lorokan, Jabal, terimakasih sudah menjadi bagian dari perjalanan ini. Langkah boleh tertatih, nafas boleh terengah, tapi kebersamaan kalian membuat setiap perjalanan terasa lebih ringan.

14. Untukmu, terimakasih sudah hadir dan memberi warna baru. Meskipun langkah ini masih awal, aku menghargai setiap pertemuan. Semoga langkah kecil ini membawa cerita baik kedepanya.
15. Teman-teman Program Studi Matematika angkatan 2021 yang selalu mendukung satu sama lain dalam proses penyelesaian skripsi ini.
16. Terakhir, saya ingin menyampaikan terima kasih yang tulus kepada diri saya sendiri, Alvin Aura Einst, erima kasih atas segala kerja keras dan semangatnya sehingga mampu bertahan dan terus melangkah sejauh ini. Terima kasih pada raga dan jiwa yang masih tetap tegar dan ikhlas menjalani semuanya hingga sekarang. Terima kasih telah percaya pada keraguan dan kelelahan dalam setiap proses ini, meskipun jalannya terasa begitu berat. Saya bangga pada diri saya sendiri! Kedepannya untuk raga yang tetap kuat, hati yang selalu tegar, Mari bekerjasama untuk lebih berkembang lagi menjadi pribadi yang lebih baik dari hari ke hari.

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. Wassalamu'alaikum Warahmatullahi Wabarakatuh.

Malang, 22 Desember 2025

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGANTAR.....	ii
HALAMAN PERSETUJUAN.....	iii
HALAMAN PENGESAHAN.....	iv
PERNYATAAN KEASLIAN TULISAN	v
MOTO	vi
HALAMAN PERSEMBAHAN.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	xi
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR.....	xiv
DAFTAR SIMBOL	xv
ABSTRAK	xvi
ABSTRACT	xvii
مستخلص البحث.....	xviii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	7
1.3 Tujuan Penelitian.....	7
1.4 Manfaat Penelitian.....	7
1.5 Batasan Masalah.....	8
1.6 Definisi Istilah	9
BAB II KAJIAN TEORI	11
2.1 Kajian Teori	11
2.1.1 Klasifikasi Biner	11
2.1.2 Regresi Logistik.....	12
2.1.3 Metode Gauss-Newton	21
2.1.4 Metode Dekomposisi QR.....	24
2.2 Keutamaan Orang Berilmu.....	29
2.3 Kajian Topik dan Teori Pendukung	32
BAB III METODE PENELITIAN	34
3.1 Jenis Penelitian	34
3.2 Data dan Sumber Data.....	34
3.3 Tahapan Penelitian.....	34
3.3.1 Persiapan Data	35
3.3.2 Penerapan Model Regresi Logistik Biner.....	35
3.3.3 Penerapan Metode Gauss-Newton.....	36
3.3.4 Penerapan Metode Dekomposisi QR.....	36
3.3.5 Evaluasi Performa Metode Optimasi.....	37
BAB IV HASIL DAN PEMBAHASAN.....	41
4.1 Deskripsi Dataset.....	41
4.1.1 Sumber Data dan Karakteristik Data	41
4.1.2 Fitur dan Target yang Digunakan	41
4.1.3 Statistik Deskriptif dan Ketidakseimbangan Kelas	44

4.1.4	Pembagian Data	45
4.2	Hasil.....	46
4.2.1	Studi Perhitungan Manual (Contoh Kecil)	47
4.2.2	Eksperimen pada Seluruh Data.....	75
4.3	Pembahasan	98
4.3.1	Analisis Stabilitas Numerik dengan Dekomposisi QR.....	98
4.3.2	Perbandingan dengan Penelitian Sebelumnya	100
4.3.3	Penerapan Metode Gauss-Newton dan Dekomposisi QR dalam Pandangan Islam	105
BAB V	KESIMPULAN.....	108
5.1	Kesimpulan.....	108
5.2	Saran	109
DAFTAR PUSTAKA		110
LAMPIRAN.....		114
RIWAYAT HIDUP		123

DAFTAR TABEL

Tabel 4.1	Hasil Korelasi Pearson untuk Seluruh Data terhadap Survived.....	42
Tabel 4.2	Variabel Respon dan Prediktor	43
Tabel 4.3	Hasil Data Splitting	46
Tabel 4.4	Data Sampel dengan 6 Data dengan 3 Fitur.....	47
Tabel 4.5	Hasil Data Setelah Normalisasi	50
Tabel 4.6	Hasil Perhitungan Nilai z_i	54
Tabel 4.7	Hasil Perhitungan Nilai σz_i	55
Tabel 4.8	Hasil Perhitungan Nilai \mathcal{L}_i	56
Tabel 4.9	Hasil Pengecekan Missing Value	76
Tabel 4.10	Hasil Normalisasi Fitur Sex	77
Tabel 4.11	Hasil Normalisasi Fitur Embarked	79
Tabel 4.12	Hasil Loss dan Akurasi untuk Data Train dan Val untuk 3 Optimasi	84
Tabel 4.13	Hasil Confusion Matrix untuk Data Uji	86
Tabel 4.14	Hasil Perhitungan Confusion Matriks.....	90
Tabel 4.15	Hasil Komparasi Data Uji untuk Tiga Metode	96
Tabel 4.16	Perbandingan Hasil Penelitian dengan Penelitian Sebelumnya.....	101

DAFTAR GAMBAR

Gambar 4.1	Distribusi Kelas Target.....	44
Gambar 4.2	Kurva Pembelajaran <i>Loss</i> untuk Tiga Optimasi.....	83
Gambar 4.3	Kurva Pembelajaran Akurasi untuk Tiga Optimasi	83

DAFTAR SIMBOL

Makna dari masing-masing simbol yang digunakan pada penelitian ini dapat dilihat sebagai berikut:

m	: Jumlah sampel/observasi dalam dataset
n	: Jumlah fitur/variabel prediktor
i	: Indeks sampel
j	: Indeks fitur
k	: Indeks iterasi dalam algoritma optimasi
X	: Matrix fitur
Y	: Vektor label/target
θ	: Vektor parameter model
b	: Bias
$\Delta\theta$: Perubahan/pembaruan parameter
z_i	: Skor linear untuk sampel ke- i
$\sigma(z_i)$: Fungsi sigmoid
\mathcal{L}_i	: Fungsi loss untuk sampel ke- i
r	: Vektor residual
J	: Matrix Jacobian
Q	: Matrix ortogonal dalam dekomposisi QR
R	: Matrix segitiga atas dalam dekomposisi QR
I	: Matrix identitas
G	: Vektor sisi kanan sistem persamaan
s	: Vektor hasil proyeksi
ε	: Toleransi konvergensi

ABSTRAK

Aura Einst, Alvin. 2025. Optimasi Regresi Logistik Menggunakan Gauss Newton dan Dekomposisi QR untuk Penyelesaian Klasifikasi Biner. Skripsi. Jurusan Matematika Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (1) Dr. Mohammad Jamhuri, M.Si. (2) Ach. Nashichuddin, M.A.

Kata kunci: Regresi logistik, Gauss-Newton, Dekomposisi QR, optimasi numerik, klasifikasi biner, Titanic Dataset. nilai-nilai islam.

Penelitian ini bertujuan untuk mengoptimalkan regresi logistik pada masalah klasifikasi biner dengan menerapkan metode Gauss-Newton yang dipadukan dengan Dekomposisi QR sebagai teknik penyelesaian sistem persamaan linear dalam proses pembaruan parameter. Optimasi ini dilakukan untuk meningkatkan stabilitas numerik serta mempercepat konvergensi dalam estimasi parameter regresi logistik. Analisis dilakukan melalui beberapa tahap, yaitu pra-pemrosesan Titanic Dataset, pembentukan model regresi logistik, perhitungan residual dan matriks Jacobian, penerapan metode Gauss-Newton dengan penyelesaian langkah pembaruan menggunakan Dekomposisi QR, serta evaluasi performa model. Hasil penelitian menunjukkan bahwa kombinasi Gauss-Newton dan Dekomposisi QR menghasilkan model dengan performa yang lebih baik dibandingkan metode berbasis gradien seperti SGD dan Adam. Pada data uji, model mencapai akurasi sebesar 83%, presisi 84%, recall 69%, dan F1-score 76%. Selain itu, metode ini jauh lebih efisien secara komputasi, ditunjukkan dengan konvergensi yang dicapai hanya dalam satu epoch dengan waktu pelatihan sekitar 0,02 detik, sementara Adam dan SGD membutuhkan puluhan hingga hampir seratus epoch. Temuan ini menunjukkan bahwa metode Gauss-Newton dengan Dekomposisi QR mampu meningkatkan akurasi sekaligus memberikan efisiensi komputasi yang signifikan dalam optimasi regresi logistik untuk klasifikasi keselamatan penumpang Titanic.

ABSTRACT

Aura Einst, Alvin. 2025. Logistic Regression Optimization Using Gauss Newton and QR Decomposition for Binary Classification Solutions. Thesis. Department of Mathematics, Faculty of Science and Technology, Universitas Maulana Malik Ibrahim Malang. Supervisors: (1) Dr. Mohammad Jamhuri, M.Si. (2) Ach. Nashichuddin, M.A.

Keywords: Logistic regression, Gauss-Newton, QR decomposition, numerical optimization, binary classification, Titanic Dataset.

This study aims to optimize logistic regression on binary classification problems by applying the Gauss-Newton method combined with QR Decomposition as a technique for solving linear equation systems in the parameter update process. This optimization is carried out to improve numerical stability and accelerate convergence in the estimation of logistics regression parameters. The analysis was carried out through several stages, namely pre-processing of the Titanic Dataset, the formation of a logistic regression model, residual calculations and Jacobian matrices, the application of the Gauss-Newton method by completing the update step using QR Decomposition, and evaluation of model performance. The results showed that the combination of Gauss-Newton and QR Decomposition resulted in a model with better performance than gradient-based methods such as SGD and Adam. In the test data, the model achieved an accuracy of 83%, precision of 84%, recall of 69%, and an F1-score of 76%. In addition, this method is much more computationally efficient, demonstrated by the convergence achieved in just one epoch with a training time of about 0.02 seconds, while Adam and SGD require tens to almost a hundred epochs. These findings show that the Gauss-Newton method with QR Decomposition is able to improve accuracy while providing significant computational efficiency in the optimization of logistic regression for the classification of Titanic passenger safety.

مستخلص البحث

أورا أينست، ألفين. 2025. تحسين الانحدار اللوجستي باستخدام جاوس نيوتن وتحليل QR لحلول التصنيف الثنائية. البحث العلمي. قسم الرياضيات، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية، مالانغ. المشرفون: (1) الدكتور. محمد جمهوري، الماجستير. (2) أحمد ناصح الدين ، الماجستير. الكلمات الأساسية: الانحدار اللوجستي، غاوس-نيوتن، تحليل QR، التحسين العددي، التصنيف الثنائي، مجموعة بيانات تاييتانيك.

هدفت هذه الدراسة إلى تحسين الانحدار اللوجستي في مسائل التصنيف الثنائي من خلال تطبيق طريقة غاوس-نيوتن مع تحليل QR كتقنية حل أنظمة المعادلات الخطية في عملية تحديث المعلمات. تنفيذ هذا التحسين لتحسين الاستقرار العددي وتسريع التقارب في تقدير معايير الانحدار اللوجستي. تم إجراء التحليل عبر عدة مراحل، وهي المعالجة المسبقة لمجموعة بيانات تاييتانيك، وتشكيل نموذج الانحدار اللوجستي، والحسابات المتبقية ومصفوفات جاكوبيان، وتطبيق طريقة غاوس-نيوتن من خلال إكمال خطوة التحديث باستخدام تفكيك QR، وتقييم أداء النموذج. أظهرت النتائج أن الجمع بين غاوس-نيوتن وتحليل QR أدى إلى نموذج بأداء أفضل من الطرق القائمة على التدرج مثل SGD وAdam. في بيانات الاختبار، حقق النموذج دقة 83%، ودقة 84%، واستدعاء 69%، ودرجة F1 بلغت 76%، بالإضافة إلى ذلك، هذه الطريقة أكثر كفاءة حسابية، كما يتضح من التقارب الذي تحقق في عصر واحد فقط بزمّن تدريب حوالي 0.02 ثانية، بينما يتطلب آدم وSGD عشرات إلى ما يقرب من مئة عصر. تظهر هذه النتائج أن طريقة غاوس-نيوتن مع تحليل QR قادرة على تحسين الدقة مع توفير كفاءة حسابية كبيرة في تحسين الانحدار اللوجستي لتصنيف سلامة ركاب التيتانيك.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sejak awal peradaban, manusia telah memanfaatkan bermacam-macam alat untuk menyelesaikan pekerjaan dengan lebih efisien. Berkat inovasi dan kecerdasan yang dimiliki, manusia telah menciptakan berbagai mesin untuk memudahkan kehidupan. Mesin-mesin ini membantu dalam memenuhi berbagai kebutuhan, seperti transportasi, industri, komputasi, dan salah satunya adalah *machine learning* (Mahesh, 2020). *Machine Learning* memiliki tiga jenis utama, yaitu *supervised learning*, *unsupervised learning*, dan *reinforcement learning*, yang masing-masing memiliki pendekatan berbeda dalam mempelajari pola dari data untuk menghasilkan keputusan yang tepat (Raschka & Mirjalili, 2017).

Klasifikasi biner merupakan salah satu tugas dalam *supervised learning*, yaitu ketika label *output*-nya hanya terdiri dari dua kelas biner. Salah satu algoritma yang paling umum digunakan untuk menyelesaikan permasalahan klasifikasi biner adalah regresi logistik. Metode regresi logistik sangat efektif dalam menangani masalah klasifikasi biner, dengan tujuan utama untuk menentukan apakah suatu data point termasuk dalam suatu kategori tertentu atau tidak (Raschka & Mirjalili, 2017).

Adapun regresi logistik adalah metode pendekatan statistik yang berfungsi untuk memodelkan hubungan fungsional antara variabel dependen berskala kategorik dengan satu atau lebih variabel independen yang berskala kontinu maupun kategorik (Hosmer Jr dkk., 2013). Dalam regresi logistik, estimasi parameter model merupakan langkah penting untuk mendapatkan model yang

akurat dalam memprediksi kelas biner. Parameter-parameter ini, yang berupa koefisien variabel prediktor, tidak dapat dihitung secara langsung dengan rumus tertutup karena fungsi *log-likelihood* regresi logistik bersifat non-linier. Oleh karena itu, diperlukan metode optimasi iteratif untuk menemukan nilai parameter yang memaksimalkan fungsi *log-likelihood* tersebut.

Berbagai metode optimasi yang umum digunakan untuk estimasi parameter regresi logistik adalah metode klasik seperti Newton-Raphson, Algoritma Genetika, dan Gradient Descent. Meskipun metode-metode tersebut efektif, tetapi memiliki keterbatasan. Metode klasik seperti Newton-Raphson memanfaatkan turunan pertama dan kedua untuk mempercepat konvergensi, namun memerlukan perhitungan matrix Hessian yang kompleks dan mahal secara komputasi, terutama pada dataset berukuran besar (Raschka & Mirjalili, 2017). Algoritma genetika (GA) menawarkan solusi pencarian global tanpa memerlukan turunan, tetapi membutuhkan waktu komputasi yang lebih lama dan konvergensi lambat (Raschka & Mirjalili, 2017). Metode *Stochastic Gradient Descent* (SGD) dan variannya efisien untuk dataset besar dan toleran terhadap noise, namun cenderung lambat konvergensinya dan sensitif terhadap pemilihan *learning rate* (Hutagalung & Sya'ban, 2024). Sebagai alternatif yang menjembatani efisiensi dan stabilitas, metode Gauss-Newton dipilih karena mampu mengaproksimasi matrix Hessian menggunakan Jacobian sehingga mengurangi beban komputasi dan memiliki konvergensi cepat pada masalah non-linier *least squares* seperti regresi logistik (Saleh, 2010).

Metode Gauss-Newton dikembangkan melalui proses pendefinisian fungsi *error* berdasarkan dataset pelatihan, kemudian aproksimasi menggunakan

pendekatan deret Taylor orde pertama. Melalui hasil persamaan aproksimasi yang diperoleh, dilakukan estimasi terhadap besar kesalahan pada masing-masing prediktor. Proses ini menghasilkan sistem persamaan linier yang memuat koefisien bobot sebagai variabel bebas. Untuk memperoleh solusi optimal, diterapkan metode *least squares* yang bertujuan memperbaiki nilai koefisien bobot dalam jaringan (Jamhuri, dkk., 2022).

Namun, metode ini berpotensi mengalami masalah singularitas pada matrix Jacobian yang dapat menghambat proses inversi dalam iterasi optimasi. Untuk mengatasi hal tersebut, digunakan metode numerik tambahan yang mampu meningkatkan stabilitas perhitungan, salah satunya adalah Dekomposisi QR. Di samping itu, Dekomposisi QR merupakan teknik yang akan memecah matrix menjadi matrix orthogonal dan segitiga atas, sehingga sistem persamaan linier yang muncul pada proses estimasi parameter regresi logistik dapat diselesaikan dengan lebih efisien dan stabil. Dengan struktur matrix segitiga atas, pencarian solusi parameter menjadi lebih mudah tanpa perlu membalik matrix besar secara langsung (Jamhuri, dkk., 2025). Dekomposisi QR digunakan sebagai metode pelengkap untuk menstabilkan penyelesaian sistem persamaan linier dalam iterasi Gauss-Newton, sekaligus mengatasi multikolinearitas dan meningkatkan stabilitas numerik (Golub & Van Loan, 2013).

Penelitian terdahulu menunjukkan keberhasilan penerapan Gauss-Newton dalam berbagai konteks klasifikasi biner, penelitian yang dilakukan oleh (Dora dkk., 2017) membahas implementasi algoritma Gauss-Newton dalam proses klasifikasi kanker payudara. Dalam studi tersebut, digunakan pendekatan Gauss-Newton *Re-Weighted Bootstrap Aggregating* (GNRBA) yang bertujuan untuk

mengoptimalkan bobot pada data pelatihan. Optimalisasi bobot ini berperan penting dalam meningkatkan kinerja klasifikasi, khususnya dari segi akurasi. Hasil eksperimen menunjukkan bahwa metode GNRBA mampu memberikan performa yang lebih unggul dibandingkan dengan pendekatan konvensional, terutama dalam menangani data medis yang bersifat kompleks dan memiliki variabilitas tinggi.

Kemudian pada penelitian (Jamhuri dkk., 2022) mengevaluasi efektivitas metode Gauss-Newton dalam meningkatkan performa regresi logistik biner, khususnya dalam konteks optimasi parameter model. Sebagai metode optimasi, metode ini mampu meningkatkan efisiensi proses konvergensi regresi logistik, terutama dalam konteks permasalahan non-linier. Jurnal ini memiliki relevansi yang tinggi, karena mengilustrasikan bagaimana metode Gauss-Newton dapat meningkatkan kinerja model regresi logistik, sehingga menjadi dasar bagi penerapan metode ini dalam penelitian skripsi.

Selain itu, penelitian oleh (Reza dkk., 2021) untuk memprediksi kemungkinan keselamatan penumpang kapal Titanic berdasarkan sejumlah variabel prediktor, antara lain jenis kelamin, usia, susunan keluarga, dan kelas tiket digunakan regresi logistik biner. Analisis tersebut dilakukan menggunakan EVIEWS untuk mengevaluasi kontribusi setiap variabel prediktor, serta mengintegrasikan *Generalized Linear Model* (GLM) guna mengembangkan fungsi regresi logistik biner. Berdasarkan hasil penelitian, metode ini berhasil dalam memprediksi variabel biner, namun masih menggunakan model standar tanpa teknik optimasi tambahan, sehingga membuka peluang penerapan metode optimasi yang lebih efisien dan akurat. Oleh karena itu, penelitian ini akan melakukan penerapan metode optimasi lebih lanjut, seperti Gauss-Newton dan dekomposisi

QR guna meningkatkan akurasi dan kinerja model.

Berdasarkan pertimbangan tersebut, kombinasi metode Gauss-Newton dan dekomposisi QR dipilih sebagai teknik optimasi karena menawarkan keunggulan ganda, Gauss-Newton mempercepat konvergensi pada masalah non-linear, sedangkan Dekomposisi QR meningkatkan stabilitas numerik dan mengatasi masalah singularitas matrix. Dengan demikian, kedua metode ini saling melengkapi dalam menghasilkan estimasi parameter yang optimal.

Titanic Dataset dipilih karena merupakan dataset publik yang sangat populer dan sering digunakan sebagai *benchmark* dalam penelitian klasifikasi biner, sehingga hasil penelitian ini dapat dengan mudah dibandingkan dengan studi-studi sebelumnya.

Dalam ilmu pengetahuan dan teknologi, manusia terus berusaha mengembangkan metode yang lebih baik untuk memahami dan menganalisis fenomena di sekitar mereka. Perkembangan kecerdasan buatan merupakan manifestasi dari kemampuan intelektual manusia yang dianugerahkan melalui akal dan pemahaman ilmiah. Dalam Al-Qur'an, tepatnya pada surat Al-Mujadilah ayat 11 (Kementerian Agama, 2017):

يَا أَيُّهَا الَّذِينَ آمَنُوا إِذَا قِيلَ لَكُمْ تَفَسَّحُوا فِي الْمَجَالِسِ فَافْسَحُوا يَفْسَحِ اللَّهُ لَكُمْ وَإِذَا قِيلَ انشُرُوا فَانْشُرُوا يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنْكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ وَاللَّهُ بِمَا تَعْمَلُونَ خَبِيرٌ (١)

Artinya: “Wahai orang-orang yang beriman, apabila dikatakan kepadamu “Berilah kelapangan di dalam majelis-majelis,” lapangkanlah, niscaya Allah akan memberi kelapangan untukmu. Apabila dikatakan, “Berdirilah,” (kamu) berdirilah. Allah niscaya akan mengangkat orang-orang yang beriman di antaramu dan orang-orang yang diberi ilmu beberapa derajat. Allah Mahateliti terhadap apa yang kamu kerjakan.” (QS. Al-Mujadilah:11).

Berdasarkan tafsir Ibnu Katsir pada buku Al-Asas (Kulle, 2016) ditegaskan bahwa Allah akan mengangkat derajat umat-Nya yang beriman dan berilmu, baik

umat-umat yang berada di dunia maupun di akhirat. Islam begitu menghargai serta menjunjung tinggi ilmu pengetahuan. Menuntut ilmu merupakan kewajiban bagi manusia sesuai dengan perintah Allah Swt. Islam memberikan kedudukan yang mulia bagi ilmu karena dengan pengetahuan, seseorang dapat memahami hal-hal yang sebelumnya tidak mereka ketahui. Ilmu juga berperan dalam mencapai kebahagiaan, baik dalam aspek keagamaan maupun kehidupan duniawi, dengan tetap menyadari berbagai tingkatan dalam alam semesta. Selain itu, ilmu membantu manusia membedakan antara kebenaran dan kesalahan, memperjelas pemahaman, memperluas wawasan, serta membuka peluang untuk memperoleh pengetahuan baru (Estuningtyas, 2018).

Dalam penelitian ini, proses optimasi parameter pada model regresi logistik untuk klasifikasi biner difokuskan melalui penerapan metode Gauss-Newton yang secara iteratif meminimalkan fungsi kuadrat residual guna mempercepat konvergensi dan mengatasi non-linieritas dalam model. Kemudian metode ini didukung oleh Dekomposisi QR yang berperan penting dalam menyelesaikan sistem persamaan linear secara numerik stabil, menghindari masalah singularitas dan meningkatkan efisiensi komputasi selama proses estimasi parameter. Kombinasi kedua metode tersebut diharapkan mampu menghasilkan estimasi parameter regresi logistik yang lebih akurat dan konvergen secara cepat, sehingga meningkatkan performa klasifikasi biner pada Titanic Dataset sebagai studi kasus. Dengan demikian, penerapan Gauss-Newton dan Dekomposisi QR tidak hanya memberikan solusi optimasi yang efektif, tetapi juga memperkuat pendekatan numerik dalam pengembangan model regresi logistik untuk aplikasi klasifikasi nyata.

1.2 Rumusan Masalah

Berdasarkan pemaparan latar belakang di atas, maka rumusan masalah dalam penelitian ini mencakup beberapa aspek berikut:

1. Bagaimana penerapan metode Gauss-Newton dan Dekomposisi QR dalam proses optimasi pada model regresi logistik?
2. Bagaimana performa dan waktu pelatihan metode Gauss-Newton dan Dekomposisi QR terhadap klasifikasi keselamatan penumpang kapal titanic?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah di atas, tujuan dari penelitian ini meliputi beberapa aspek berikut:

1. Menerapkan metode Gauss–Newton yang dipadukan dengan Dekomposisi QR untuk optimasi parameter pada regresi logistik, lengkap dengan skema praproses data dan kriteria konvergensi yang terukur.
2. Mengevaluasi performa dan waktu pelatihan dari metode Gauss-Newton dan Dekomposisi QR pada tugas klasifikasi keselamatan penumpang Titanic, menggunakan metrik akurasi, presisi, *recall*, *F1-Score* serta durasi pelatihan/iterasi, dan membandingkannya dengan metode pembandingan.

1.4 Manfaat Penelitian

Penelitian ini diharapkan memberikan manfaat baik secara teoritis maupun praktis yaitu:

1. Manfaat Teoritis

Penelitian ini memberikan kontribusi teoritis dalam pengembangan metode optimasi parameter model regresi logistik, khususnya dengan menggunakan pendekatan Gauss-Newton dan Dekomposisi QR. Hal ini memperkaya kajian akademik di bidang pemodelan matematis dan komputasi numerik dalam klasifikasi biner, serta dapat menjadi rujukan bagi penelitian lanjutan terkait efisiensi dan akurasi metode optimasi.

2. Manfaat Praktis

Secara praktis, penelitian ini dapat membantu peneliti dan praktisi dalam memilih metode optimasi yang lebih efisien dan akurat untuk membangun model klasifikasi berdasarkan data historis. Khususnya dalam konteks prediksi keselamatan penumpang, hasil penelitian ini dapat digunakan untuk mendukung pengambilan keputusan berbasis data dan pengembangan sistem prediktif yang lebih andal di bidang transportasi.

1.5 Batasan Masalah

Batasan masalah dirancang untuk memastikan bahwa cakupan pembahasan dan proses penyusunan penelitian tetap terarah guna mencapai tujuan yang telah ditetapkan, yaitu:

1. Penelitian ini hanya akan menggunakan 7 fitur atau variabel prediktor dari 14 fitur yang tersedia, yaitu : *sex*, *fare*, *Pclass*, *Embarktion*, *age*, *sibps*, *parch*, pemilihan fitur yang telah disebutkan didasarkan pada uji korelasi pearson.
2. Metode Gauss-Newton yang digunakan tidak dimodifikasi secara adaptif, melainkan menggunakan versi klasik dengan pendekatan iteratif

berdasarkan deret Taylor orde pertama tanpa regularisasi tambahan. Dekomposisi QR yang diterapkan terbatas pada pendekatan *Householder*, tidak mencakup metode alternatif lain.

1.6 Definisi Istilah

Penelitian ini mencakup sejumlah definisi yang menjadi dasar dalam pembahasannya. Berikut diberikan definisi-definisi untuk memberikan pemahaman yang lebih jelas:

- Regresi Logistik : Model statistik/pemelajaran mesin untuk memprediksi probabilitas kejadian biner (0/1) menggunakan fungsi logit/sigmoid.
- Klasifikasi Biner : Tugas klasifikasi dengan dua kelas saja (mis. selamat vs tidak selamat).
- Metode Gauss-Newton : Algoritma optimasi iteratif untuk *nonlinear least squares* yang memperkirakan kelengkungan tanpa Hessian penuh umumnya lebih cepat dan stabil daripada gradien murni untuk mencari parameter.
- Metode Dekomposisi QR : Faktorisasi matrix yang memisahkan sebuah matrix menjadi bagian ortonormal dan segitiga atas untuk menyelesaikan *least squares* secara lebih stabil.
- Iterasi : Satu langkah pembaruan dalam algoritma optimasi diulang hingga kriteria berhenti terpenuhi.
- Konvergensi : Keadaan ketika proses iteratif mendekati solusi.

- Ill-conditioned* : Situasi numerik ketika matrix memiliki angka kondisi tinggi, sehingga perhitungan invers/normal equation tidak stabil dan sensitif terhadap gangguan kecil.
- Early stopping* : Strategi menghentikan pelatihan lebih awal saat metrik validasi tak lagi membaik untuk menghindari *overfitting* dan menghemat waktu.
- Disminishing returns* : Fenomena manfaat tambahan menurun, penambahan epoch/iterasi memberi perbaikan metrik yang makin kecil.
- Overfitting* : Model terlalu menyesuaikan data latih sehingga kinerja pada data baru/validasi menurun.
- Underfitting* : Model terlalu sederhana atau pelatihan kurang sehingga gagal menangkap pola penting kinerja buruk baik di latih maupun validasi.
- Leakage* : Kebocoran informasi ketika fitur atau proses praproses memakai data yang seharusnya tidak tersedia saat prediksi nyata.

BAB II

KAJIAN TEORI

2.1 Kajian Teori

2.1.1 Klasifikasi Biner

Klasifikasi bertujuan untuk mempelajari pemetaan dari *input* x ke *output* y , di mana y termasuk dalam himpunan $y \in \{1, \dots, C\}$, dengan C adalah jumlah kelas. Dapat disebut klasifikasi biner jika $C = 2$ dalam hal ini sering mengasumsikan y termasuk dalam himpunan $\{0, 1\}$, lalu dapat disebut klasifikasi multi-kelas jika $C > 2$ (Murphy, 2012).

Klasifikasi biner adalah salah satu metode dalam klasifikasi yang bertujuan untuk memisahkan data ke dalam dua kategori *output* berdasarkan pola yang dipelajari dari data pelatihan. Dalam proses klasifikasi biner, sebuah model atau algoritma dibangun untuk mengidentifikasi apakah suatu data termasuk dalam salah satu dari dua kategori berdasarkan karakteristik atau variabel independen yang tersedia (Haddock, 2011).

Secara matematis, klasifikasi biner dapat direpresentasikan melalui pendekatan tertentu. Dalam pendekatan ini, suatu himpunan data berisi pasangan *input-output*, di mana *input* berupa vektor fitur dan *output* merepresentasikan label kelas. Tujuan dari klasifikasi biner adalah mengembangkan fungsi yang mampu memetakan vektor fitur ke dalam salah satu dari dua kelas yang tersedia. Fungsi tersebut didefinisikan sebagai berikut:

$$f: \mathbb{R}^n \rightarrow \{0,1\}$$

di mana \mathbb{R}^n merepresentasikan ruang fitur berdimensi n sedangkan $\{0,1\}$ menunjukkan bahwa fungsi f hanya menghasilkan dua kemungkinan ruang

output, yaitu 0 atau 1 yang masing-masing melambangkan dua kelas berbeda. Dalam hal ini, n adalah jumlah variabel bebas (independen) yang memengaruhi nilai variabel dependen Y .

Pada penelitian (Semendawai et al., 2024) disebutkan bahwa klasifikasi biner berperan penting dalam berbagai sektor dan industri, khususnya dalam analisis data serta proses pengambilan keputusan. Salah satu penerapannya dalam bidang keamanan siber, teknik klasifikasi biner diterapkan dalam deteksi *malware* dengan memanfaatkan algoritma pembelajaran mesin. Penelitian tersebut menunjukkan bahwa teknik seperti *K-Nearest Neighbour* dan *Decision Tree* dapat mencapai akurasi tinggi dalam mengidentifikasi serangan *shellcode*, yang menunjukkan ancaman serius terhadap keamanan sistem komputer. Lalu dalam (Abdurrahman dkk., 2014) menyatakan bahwa pada sektor kesehatan, metode klasifikasi biner berkontribusi dalam mengidentifikasi pasien yang terindikasi mengidap penyakit tertentu melalui penerapan regresi logistik. Algoritma QUEST (*Quick, Unbiased, and Efficient Statistical Tree*) meningkatkan ketepatan diagnosis Gauss-Newtonosis medis melalui optimalisasi pengolahan data input. Penelitian tersebut menunjukkan bahwa algoritma QUEST efektif untuk mendeteksi kelainan pada fungsi hati berdasarkan analisis parameter tes fungsi hati.

2.1.2 Regresi Logistik

Regresi logistik merupakan metode statistik yang digunakan untuk menganalisis hubungan antara satu atau lebih variabel independen terhadap variabel dependen yang bersifat biner, yaitu hanya memiliki dua kategori, seperti

“selamat” dan “tidak selamat” atau “positif” dan “negatif”. Berbeda dari regresi linier yang berfungsi untuk memprediksi variabel dependen berskala kontinu, regresi logistik dirancang untuk menghitung probabilitas terjadinya suatu peristiwa, di mana nilai *output* yang dihasilkan berada pada rentang $[0, 1]$ (Hosmer Jr dkk., 2013).

Metode regresi logistik secara luas dimanfaatkan untuk menyelesaikan berbagai permasalahan klasifikasi biner. Namun, implementasi regresi logistik pada dataset berukuran besar memerlukan optimasi yang efisien guna memastikan kinerja yang optimal dan menghindari masalah komputasi. Salah satu kelebihan regresi logistik terletak pada kemampuannya dalam mengolah variabel independen, baik yang bersifat kontinu maupun kategorikal, dalam proses pemodelan. Selain itu, model ini memiliki kemampuan untuk menghasilkan *output* dalam bentuk probabilitas, yang sangat berguna dalam berbagai aplikasi praktis, seperti prediksi risiko kesehatan, penilaian kelayakan kredit, analisis pemasaran, serta mendukung pengambilan keputusan di berbagai sektor lainnya (Agresti, 2012).

Regresi logistik memiliki beberapa asumsi penting yang harus diperhatikan agar hasil analisis tetap valid dan dapat diinterpretasikan dengan baik. Menurut (Schreiber-Gregory & Bader, 2018) beberapa asumsi utama dalam regresi logistik meliputi:

1. Tidak terdapat indikasi multikolinearitas yang berarti di antara variabel-variabel bebas.
2. Data tidak menunjukkan adanya outlier yang memengaruhi secara signifikan hasil estimasi model.

Dengan demikian, regresi logistik memungkinkan analisis hubungan antara variabel independen dan probabilitas terjadinya suatu peristiwa dalam bentuk model matematis yang dapat digunakan untuk pengambilan keputusan. Untuk merepresentasikan data dalam analisis ini, variabel independen dan dependen akan disusun dalam format matrix X dan Y yang memudahkan proses estimasi parameter model serta interpretasi hasil analisis. Data dalam bentuk matrix ini akan digunakan untuk mengidentifikasi pola dan hubungan yang signifikan dalam model regresi logistik:

1. Matrix fitur:

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix}$$

2. Matrix label:

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

Matrix X direpresentasikan sebagai kumpulan vektor baris yang dapat ditulis sebagai $X = (x_1, x_2, x_3, \dots, x_m)^T$ di mana setiap $x_i \in \mathbb{R}^n$ adalah vektor fitur dari sampel ke- i .

Sebagai ilustrasi, akan diberikan sebuah pasangan data yang akan digunakan sebagai contoh dalam proses pelatihan $(x_i, y_i), \forall i = 1, 2, \dots, m$ dengan $x_i \in \mathbb{R}^n$ dan $y_i \in \{1, 0\}$. Di mana dalam dataset tersebut, m merepresentasikan jumlah total sampel yang digunakan dalam proses pelatihan model. Setiap sampel terdiri dari pasangan data (x_i, y_i) , di mana x_i adalah vektor fitur yang berisi nilai-nilai variabel independen, dan y_i adalah label biner yang menunjukkan kategori

target. Dengan adanya m sampel, model dapat mempelajari pola hubungan antara variabel independen dan probabilitas terjadinya suatu peristiwa yang direpresentasikan oleh y_i . Semakin besar nilai m , semakin banyak informasi yang tersedia untuk melatih model, sehingga diharapkan model dapat menghasilkan prediksi yang lebih akurat.

Berdasarkan penjelasan sebelumnya, arsitektur dari regresi logistik dapat didefinisikan sebagai berikut:

1. Input Vektor (vektor fitur)

Lapisan input merepresentasikan sampel data secara matematis, di mana setiap sampel x_i diwujudkan dalam bentuk vektor yang memuat nilai-nilai dari fitur-fitur relevan sebagai variabel prediktor. Vektor *input* ini kemudian akan digunakan dalam perhitungan regresi logistik untuk menghasilkan prediksi probabilitas kelas dari sampel data tersebut. Gunakan x_i, y_i sebagai prediktor dan label secara berturut-turut.

2. Koefisien bobot

Koefisien bobot $\mathbf{w} = (w_1, w_2, \dots, w_n)^T$ merupakan vektor parameter yang menyatakan pengaruh masing-masing fitur terhadap probabilitas target. Bias b merupakan konstanta tambahan yang berperan sebagai intersep model. Selama pelatihan model, nilai bobot, termasuk bias, diperbarui secara iteratif menggunakan algoritma optimasi, seperti Gauss-Newton untuk meminimalkan fungsi *loss*.

3. Fungsi transfer

Target linear dari model diperoleh dari kombinasi linear antara bobot dan fitur yang disebut z_i :

$$z_i = \mathbf{w}^T \mathbf{x}_i + b \quad (2.1)$$

Persamaan diatas dapat dituliskan secara eksplisit dalam bentuk skalar sebagai berikut:

$$z_i = w_1 x_{i,1} + w_2 x_{i,2} + \dots + w_n x_{i,n} + b$$

Nilai z_i merupakan hasil dari kombinasi linear antara kontribusi setiap fitur-fitur *input* dengan bobotnya. Nilai ini belum dalam bentuk probabilitas, sehingga masih berupa nilai kontinu.

4. Output layer

Tahapan output *layer* bertugas mengkonversi nilai linier z_i menjadi nilai probabilitas menggunakan fungsi aktivasi sigmoid:

$$P(\mathbf{y}_i = 1 | \mathbf{x}_i) = \sigma(z_i) = \frac{1}{1 + \exp(-z_i)} \quad (2.2)$$

fungsi sigmoid memetakan z_i ke dalam interval $[0, 1]$, sehingga dapat ditafsirkan sebagai probabilitas bahwa \mathbf{x}_i termasuk ke dalam kelas positif.

5. Prediksi

Proses prediksi berfungsi mengonversi target berupa probabilitas ke dalam bentuk label kelas diskrit, yang kemudian dapat dimanfaatkan dalam analisis data atau proses pengambilan keputusan berbasis klasifikasi.

$$\mathbf{y}_i = \begin{cases} 1, & \sigma(z_i) \geq 0,5 \\ 0, & \sigma(z_i) \leq 0,5 \end{cases}$$

di mana:

\mathbf{y}_i = label kelas untuk sampel \mathbf{x}_i , yaitu hasil klasifikasi

berdasarkan nilai probabilitas dari fungsi sigmoid,

$\sigma(z_i) \geq 0,5$, maka $\mathbf{y}_i = 1$ mempresentasikan kelas positif,

$\sigma(z_i) \leq 0,5$, maka $\mathbf{y}_i = 0$ mempresentasikan kelas negatif.

Dalam ranah *machine learning*, konsep *loss function* memegang peranan krusial dalam mengevaluasi performa sebuah model. Fungsi ini menjadi tolak seberapa jauh hasil prediksi terhadap label aktual. Berdasarkan (Wali, 2022) untuk suatu data *log loss function* individu didefinisikan secara matematis sebagai berikut:

$$\mathcal{L}_i = -[y_i \log(\sigma(z_i)) + (1-y_i)\log(1-\sigma(z_i))] \quad (2.3)$$

Untuk mengevaluasi performa model secara keseluruhan, digunakan rata-rata fungsi *log-loss* dari semua sampel, yang dikenal sebagai *binary cross-entropy loss*. Fungsi ini menjadi fungsi objektif yang ingin diminimalkan selama pelatihan, dan bergantung pada parameter model $\theta = [\mathbf{w}, \mathbf{b}]$. Dalam hal ini, berdasarkan penelitian (Jamhuri dkk., 2022) secara matematis, total *loss* ini didefinisikan sebagai:

$$\mathcal{L}(\theta) = \frac{1}{m} \sum_{i=1}^m \mathcal{L}_i \quad (2.4)$$

atau dapat dinyatakan sebagai:

$$\mathcal{L}(\theta) = -\frac{1}{m} \sum_{i=1}^m [y_i \log(\sigma(z_i)) + (1-y_i)\log(1-\sigma(z_i))] \quad (2.5)$$

Proses pelatihan model ini melibatkan empat tahapan utama, yaitu *feedforward*, perhitungan loss, *backward*, dan prediksi dengan penjelasan yang lebih rinci sebagai berikut:

Tahapan yang pertama yaitu *Feedforward*. Pada tahapan ini bertujuan untuk menghasilkan *output* atau prediksi awal dari model berdasarkan *input* yang diberikan. Matrix $X \in \mathbb{R}^{m \times n}$ di mana m merupakan jumlah data dan n adalah jumlah fitur, dimasukkan kedalam model. Model kemudian menghasilkan nilai

linier z_i dan mengonversinya ke probabilitas dengan fungsi sigmoid berdasarkan persamaan (2.2) yang merupakan fungsi aktivasi sigmoid.

Tahapan yang kedua yaitu perhitungan fungsi *loss*. Fungsi *loss* dihitung menggunakan *log-likelihood* yang telah dirumuskan dalam persamaan (2.3) untuk mengukur kesalahan prediksi terhadap nilai aktual. Nilai *loss* digunakan sebagai dasar dalam perhitungan gradien. Fungsi *loss* ini akan menghasilkan nilai numerik yang merepresentasikan seberapa buruk kinerja model. Semakin besar nilai *loss*, semakin besar kesalahan model. Nilai *loss* yang diperoleh akan dimanfaatkan pada tahap berikutnya, yaitu proses *backward*.

Pada tahapan *Backward* (optimasi) bertujuan untuk memperbarui parameter model dengan menurunkan fungsi *loss* terhadap setiap bobot dan bias. Tujuannya adalah untuk meminimalkan *loss*, sehingga model dapat membuat prediksi yang lebih akurat. Proses *backward* melibatkan perhitungan gradien dari fungsi *loss* $\nabla \mathcal{L}(\theta)$ untuk meminimalkan $\mathcal{L}(\theta)$. Gradienya dinyatakan sebagai:

$$\nabla \mathcal{L}(\theta) = \left(\frac{\partial \mathcal{L}}{\partial \theta_0}, \frac{\partial \mathcal{L}}{\partial \theta_1}, \frac{\partial \mathcal{L}}{\partial \theta_2}, \dots, \frac{\partial \mathcal{L}}{\partial \theta_n} \right) \quad (2.6)$$

Agar bisa melakukan pembaruan parameter, langkah pertama yang dilakukan dalam banyak metode optimasi adalah mencari turunan dari fungsi *log loss* dalam klasifikasi biner terhadap masing-masing parameter dapat ditentukan menggunakan persamaan (2.3). Sebelum masuk ke Gauss-Newton, tinjau gradien biasa dulu. Dalam hal ini, menurunkan total *loss* terhadap setiap bobot θ . Dengan pendekatan ini, bobot dan bias diperbarui iteratif hingga model mencapai konvergensi, di mana fungsi *loss* mendekati nilai minimum, memastikan performa klasifikasi yang optimal.

Untuk mempermudah perhitungan turunan parsial, digunakan substitusi $c_1 = \log \sigma(z_i)$ dan $c_2 = \log(1 - \sigma(z_i))$, dan turunan dari c_1 dan c_2 terhadap w_j adalah sebagai berikut:

Maka turunan c_1 terhadap $\log \sigma(z_i)$ terhadap θ_j :

$$\begin{aligned} \frac{\partial c_1}{\partial \theta_j} &= \frac{\partial \log \sigma(z_i)}{\partial \sigma(z_i)} \frac{\partial \sigma(z_i)}{\partial z_i} \frac{\partial z_i}{\partial \theta_j} = \frac{1}{\sigma(z_i)} \sigma(z_i)(1 - \sigma(z_i))x_{i,j} \\ &= (1 - \sigma(z_i)) \end{aligned} \quad (2.7)$$

Dan turunan c_2 terhadap $(1 - \sigma(z_i))$ terhadap θ_j :

$$\begin{aligned} \frac{\partial c_2}{\partial \theta_j} &= \frac{\partial \log(1 - \sigma(z_i))}{\partial(1 - \sigma(z_i))} \frac{\partial(1 - \sigma(z_i))}{\partial z_i} \frac{\partial z_i}{\partial w_j} \\ &= \frac{1}{1 - \sigma(z_i)} (-\sigma(z_i)(1 - \sigma(z_i))x_{i,j}) \\ &= -\sigma(z_i) \end{aligned} \quad (2.8)$$

Kemudian substitusi $\frac{\partial c_1}{\partial \theta_j}$ dan $\frac{\partial c_2}{\partial \theta_j}$ yaitu persamaan (2.7) dan (2.8) kedalam persamaan (2.5). Substitusi ini bertujuan untuk menghitung turunan parsial dalam kasus klasifikasi biner dengan fungsi *loss log-likelihood* sehingga menjadi:

$$\frac{\partial \mathcal{L}_\theta}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^m (y_i - \sigma(z_i))x_{i,j} \quad (2.9)$$

untuk $\forall j = 1, \dots, n$.

Tahapan yang terakhir yaitu Prediksi. Model yang telah dilatih akan mengklasifikasikan data masukan σ berdasarkan probabilitas yang dihasilkan oleh persamaan sigmoid. Metode regresi logistik secara luas dimanfaatkan untuk menyelesaikan berbagai permasalahan klasifikasi biner. Namun, implementasi regresi logistik pada dataset berukuran besar memerlukan optimasi yang efisien guna memastikan kinerja yang optimal dan menghindari masalah komputasi.

Untuk menerapkan metode Gauss-Newton, fungsi *log-loss* didekati menggunakan pendekatan nonlinear *least squares* dengan mendefinisikan residual:

$$r_i = y_i - \sigma(z_i) \quad (2.10)$$

sehingga:

$$r = y - \sigma \quad (2.11)$$

dengan $y \in \mathbb{R}^m$ sebagai vektor label asli, dan $\sigma = [\sigma(z_1), \dots, \sigma(z_m)]^T$ sebagai vektor probabilitas prediksi model.

Pendefinisian residu ini memungkinkan untuk mereformulasi masalah optimasi dari *log-loss* menjadi pendekatan *nonlinear least squares*. Dengan bentuk ini, metode Gauss-Newton dapat diterapkan untuk memperkirakan parameter secara iteratif.

Untuk menerapkan Gauss-Newton, diperlukan turunan parsial dari setiap residu r_i terhadap parameter θ , yang membentuk matrix Jacobian J . Komponen Jacobian diberikan sebagai berikut:

$$\begin{aligned} r_i = y_i - \sigma(z_i) \rightarrow \frac{\partial r_i}{\partial \theta_j} &= -\sigma(z_i)(1 - \sigma(z_i))x_{i,j} \\ \frac{\partial r_i}{\partial \mathcal{b}} &= -\sigma(z_i)(1 - \sigma(z_i)) \end{aligned} \quad (2.12)$$

Kolom terakhir dari J menyatakan turunan terhadap bias \mathcal{b} . Notasi ringkasnya berupa:

$$J_{i,j} = -\sigma(z_i)(1 - \sigma(z_i))x_{i,j}, \text{ untuk } j = 1, \dots, n$$

$$J_{i,n+1} = -\sigma(z_i)(1 - \sigma(z_i)), \text{ untuk kolom bias}$$

Jacobian ini akan digunakan dalam formulasi iteratif Gauss-Newton, yang membutuhkan perkalian $J^T J$ dan $J^T \mathbf{r}$ untuk menghitung perubahan parameter $\Delta\theta$.

2.1.3 Metode Gauss-Newton

Metode Gauss-Newton merupakan suatu algoritma yang digunakan untuk meminimalkan jumlah kuadrat dari galat atau untuk memaksimalkan fungsi *likelihood* dalam proses estimasi parameter model. Prinsip utama yang mendasari metode ini adalah pemanfaatan deret Taylor untuk mengaproksimasi persamaan nonlinier dalam bentuk linier. Dengan pendekatan tersebut, metode ini memungkinkan perolehan estimasi baru terhadap parameter yang bergerak menuju nilai yang memaksimalkan fungsi *likelihood* (Gallant, 1985).

Metode Gauss-Newton diterapkan untuk mengoptimalkan fungsi *loss non-linear* dengan melakukan pendekatan menggunakan model kuadrat. Dalam regresi logistik, penelitian yang dilakukan oleh (Jamhuri dkk., 2022) Fungsi *loss* untuk satu sampel i didefinisikan sebagaimana pada persamaan (2.3):

$$\mathcal{L}_i = -[y_i \log(\sigma(z_i)) + (1-y_i) \log(1-\sigma(z_i))].$$

Kemudian untuk mempermudah linearisasi, diandaikan dua fungsi berikut:

$$\begin{aligned} c_1(\theta) &= \log(\sigma(z_i)) \\ c_2(\theta) &= \log(1 - \sigma(z_i)) \end{aligned} \tag{2.13}$$

Sehingga:

$$\mathcal{L}_i(\theta) = -[y_i c_1(\theta) + (1-y_i) c_2(\theta)]$$

Karena kedua fungsi ini bersifat non-linear terhadap parameter θ , maka digunakan ekspansi deret Taylor orde pertama sebagai berikut:

$$\begin{aligned} c_1(\theta + \Delta\theta) &\approx c_1(\theta) + \sum_{j=0}^n \frac{\partial c_1(\theta)}{\partial w_j} \Delta\theta_j \\ c_2(\theta + \Delta\theta) &\approx c_2(\theta) + \sum_{j=0}^n \frac{\partial c_2(\theta)}{\partial w_j} \Delta\theta_j \end{aligned} \tag{2.14}$$

Ekspansi Taylor ini membantu mendekati fungsi yang kompleks dengan bentuk linear yang lebih mudah digunakan dalam metode Gauss-Newton. Untuk menyederhanakan notasi, ekspansi Taylor di atas dapat ditulis menggunakan notasi gradien sebagai berikut:

$$\begin{aligned} c_1(\theta + \Delta\theta) &\approx c_1(\theta) + \nabla c_1(\theta)^T \cdot (\Delta\theta) \\ c_2(\theta + \Delta\theta) &\approx c_2(\theta) + \nabla c_2(\theta)^T \cdot (\Delta\theta) \end{aligned} \quad (2.14)$$

Dengan bentuk ini, dapat menggambarkan perubahan fungsi secara lebih ringkas dalam bentuk vektor dan matrix. Di mana vektor gradien dari fungsi c_1 dan c_2 masing-masing terhadap parameter θ dapat didefinisikan sebagai $\nabla c_1(\theta) = \left(\frac{\partial c_1}{\partial \theta_0}, \frac{\partial c_1}{\partial \theta_1}, \dots, \frac{\partial c_1}{\partial \theta_n} \right)$, $\nabla c_2(\theta) = \left(\frac{\partial c_2}{\partial \theta_0}, \frac{\partial c_2}{\partial \theta_1}, \dots, \frac{\partial c_2}{\partial \theta_n} \right)$, dan $\Delta\theta = (\Delta\theta_0, \Delta\theta_1, \dots, \Delta\theta_n)^T$. Dan turunan parsial dari c_1 dan c_2 terhadap θ_j dapat di definisikan sebagai :

$$\frac{\partial c_1}{\partial \theta_j} = (1 - \sigma(z_i))x_{ij}, \quad \frac{\partial c_2}{\partial \theta_j} = -\sigma(z_i)x_{ij}, \quad \forall j = 0, 1, \dots, n \quad (2.16)$$

Untuk melakukan linierisasi pada persamaan (2.3) menggunakan fungsi aproksimasi dari gradien c_1 dan c_2 pada persamaan (2.15), serta turunannya pada persamaan (2.16) Dengan demikian fungsi *loss* dapat diaproksimasi sebagai:

$$\mathcal{L}_i(\theta + \Delta\theta) \approx -[y_i(c_1(\theta) + \nabla c_1(\theta)^T \cdot \Delta\theta) + (1 - y_i)(c_2(\theta) + \nabla c_2(\theta)^T \cdot \Delta\theta)]$$

Persamaan ini dapat ditulis ulang dengan mengelompokkan suku yang mengandung $\Delta\theta$ pada *leftside*:

$$[y_i \nabla c_1(\theta)^T + (1 - y_i) \nabla c_2(\theta)^T] \cdot \Delta\theta = -[y_i c_1(\theta) + (1 - y_i) c_2(\theta)] \quad (2.17)$$

Gradien dari fungsi logaritma sigmoid yang telah dihitung sebelumnya menunjukkan bahwa selisih antara $\nabla c_1(\theta)$ dan $\nabla c_2(\theta)$ selalu menghasilkan vektor input \mathbf{x}_i . Hal ini dapat direpresentasikan sebagai :

$$\nabla c_1(\theta) - \nabla c_2(\theta) = (1 - \sigma(z_i))\mathbf{x}_i - (-\sigma(z_i)\mathbf{x}_i) = (1 - \sigma(z_i) + \sigma(z_i))\mathbf{x}_i = \mathbf{x}_i$$

Maka, jika dikalikan menggunakan label y_i , hasilnya akan tetap berupa vektor fitur yang telah disesuaikan dengan label tersebut berupa :

$$y_i (\nabla c_1(\theta) - \nabla c_2(\theta)) = y_i \mathbf{x}_i$$

Dari persamaan diatas, kita dapat menyederhanakan as kanan yaitu sebagai berikut

$$y_i (\nabla c_1(\theta) - \nabla c_2(\theta)) + \nabla c_2(\theta) = y_i \mathbf{x}_i - \sigma(z_i) \mathbf{x}_i = (y_i - \sigma(z_i)) \mathbf{x}_i$$

Terapkan persamaan diatas kedalam persamaan yang telah dilinearisasi. diperoleh

$$(y_i - \sigma(z_i)) \mathbf{x}_i \cdot \Delta \theta = -[y_i \log(\sigma(z_i)) + (1 - y_i) \log(1 - \sigma(z_i))].$$

Di mana

$$g_i = -[y_i \log(\sigma(z_i)) + (1 - y_i) \log(1 - \sigma(z_i))] \quad (2.18)$$

Dalam konteks optimasi model, pendekatan ini terbukti efektif karena memungkinkan perhitungan gradien dilakukan secara lebih sederhana dan efisien, khususnya pada metode yang bergantung pada informasi turunan Gauss-Newton.

Jika hasil diatas disubstitusikan ke dalam persamaan (2.17) maka akan didapatkan:

$$(y_i - \sigma(z_i)) \mathbf{x}_i \cdot \Delta \theta = g_i \quad (2.19)$$

Persamaan diatas membentuk sistem persamaan sebanyak m , yaitu:

$$\begin{aligned} (y_1 - \sigma(z_1)) [1, x_{1,1}, x_{1,2}, \dots, x_{1,n}] \cdot \Delta \theta &= g_1 \\ (y_2 - \sigma(z_2)) [1, x_{2,1}, x_{2,2}, \dots, x_{2,n}] \cdot \Delta \theta &= g_2 \\ &\vdots \\ (y_m - \sigma(z_m)) [1, x_{m,1}, x_{m,2}, \dots, x_{m,n}] \cdot \Delta \theta &= g_m \end{aligned}$$

Sistem persamaan tersebut dapat ditulis dalam bentuk persamaan matrix sebagai berikut:

$$\begin{bmatrix} y_1 - \sigma(z_1) & (y_1 - \sigma(z_1)) x_{1,1} & \dots & (y_1 - \sigma(z_1)) x_{1,n} \\ y_2 - \sigma(z_2) & (y_2 - \sigma(z_2)) x_{2,1} & \dots & (y_2 - \sigma(z_2)) x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ y_m - \sigma(z_m) & (y_m - \sigma(z_m)) x_{m,1} & \dots & (y_m - \sigma(z_m)) x_{m,n} \end{bmatrix} \begin{bmatrix} \Delta\theta_0 \\ \Delta\theta_1 \\ \vdots \\ \Delta\theta_n \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_m \end{bmatrix}$$

dengan:

$$\begin{bmatrix} y_1 - \sigma(z_1) & (y_1 - \sigma(z_1)) x_{1,1} & \dots & (y_1 - \sigma(z_1)) x_{1,n} \\ y_2 - \sigma(z_2) & (y_2 - \sigma(z_2)) x_{2,1} & \dots & (y_2 - \sigma(z_2)) x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ y_m - \sigma(z_m) & (y_m - \sigma(z_m)) x_{m,1} & \dots & (y_m - \sigma(z_m)) x_{m,n} \end{bmatrix} = J, \quad \begin{bmatrix} \Delta\theta_0 \\ \Delta\theta_1 \\ \vdots \\ \Delta\theta_n \end{bmatrix} = \Delta\theta, \quad \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_m \end{bmatrix} = G$$

Dalam metode Gauss–Newton, setelah dilakukan pendekatan linier terhadap fungsi residual, diperoleh suatu sistem persamaan linier dalam bentuk:

$$J\Delta\theta = G \quad (2.20)$$

Dalam konteks regresi logistik, hubungan antara jumlah data (m) dan jumlah parameter (n) memengaruhi sifat sistem persamaan linier $J\Delta\theta = G$. Jika $m = n$, dan matrix J tidak singular, maka sistem memiliki satu solusi tunggal. Jika $m < n$, sistem menjadi *underdetermined* dan umumnya memiliki banyak solusi, artinya terdapat lebih dari satu himpunan parameter yang memenuhi persamaan tersebut. Sebaliknya, pada kasus yang digunakan dalam penelitian ini, yaitu $m < n$, sistem bersifat *overdetermined* sehingga umumnya tidak memiliki solusi eksak yang memenuhi $J\Delta\theta = G$ untuk semua baris. Oleh karena itu, dicari solusi pendekatan dalam arti *least squares* dengan meminimalkan $\|J\Delta\theta - G\|^2$.

2.1.4 Metode Dekomposisi QR

Dekomposisi QR telah dimanfaatkan dalam berbagai konteks optimasi untuk menangani masalah stabilitas numerik dan memecah tugas optimasi yang kompleks menjadi subtugas yang lebih mudah dikelola (Björck, 1996). Ketika

diterapkan dalam kerangka Gauss-Newton, Dekomposisi QR telah terbukti mengurangi intensitas komputasi dalam menangani matrix *Hessian* (Schraudolph, 1999), sehingga membuat metode ini lebih layak untuk masalah yang lebih besar.

Dekomposisi QR adalah metode yang kuat untuk memecahkan persamaan linier (Zhang dkk., 2020). Metode dekomposisi QR digunakan untuk menyelesaikan sistem persamaan linier dalam kerangka optimasi regresi logistik menggunakan pendekatan *least-squares*. Metode tersebut akan mengubah matrix J menjadi produk dari matrix ortogonal Q dan matrix segitiga atas R , yang memfasilitasi identifikasi solusi melalui substitusi balik. Direpresentasikan sebagai:

$$J = QR \quad (2.21)$$

di mana:

Q = matrix ortogonal, yaitu matrix yang memenuhi sifat $Q^T Q = I$, yang berarti kolom-kolomnya saling ortonormal.

R = matrix segitiga atas, yaitu matrix yang memiliki elemen nol di bawah diagonal utama.

Pada tahap selanjutnya, matrix Jacobian J akan direpresentasikan dalam bentuk dekomposisi ortogonal menggunakan faktorisasi $J = QR$ sebagaimana ditunjukkan pada Persamaan (2.21). Dalam perhitungan manual, proses dekomposisi ini akan dilakukan menggunakan metode Gram-Schmidt, yang bertujuan untuk menghasilkan matrix ortonormal Q dan matrix segitiga atas R . Metode ini bekerja dengan melakukan proses ortogonalitas bertahap terhadap setiap kolom vektor dari J . Misalkan:

$$J = [a_1, a_2, \dots, a_n]$$

maka proses Gram–Schmidt dapat dirumuskan sebagai berikut:

$$q_1 = \frac{a_1}{\|a_1\|}, \quad (2.22)$$

$$q_k = \frac{a_k - \sum_{j=1}^{k-1} (q_j^t a_k) q_j}{\|a_k - \sum_{j=1}^{k-1} (q_j^t a_k) q_j\|}, k = 2, 3, \dots, n. \quad (2.23)$$

Hasil dari proses ini menghasilkan matrix $Q = [q_1, q_2, \dots, q_n]$ yang ortonormal, dan $R = Q^T J$ matrix segitiga atas.

Dalam metode *least-squares*, sistem *overdetermined* tanpa solusi eksak yang diperoleh dari langkan Gauss-Newton dengan persamaan (2.20):

$$J\Delta\theta = G$$

di mana:

J = matrix jacobian (ukuran $m \times n$, dengan $m > n$).

$\Delta\theta$ = vektor parameter yang ingin di estimasi (ukuran $n \times 1$).

G = vektor residual (ukuran $m \times 1$).

Karena sistem tersebut umumnya tidak memiliki solusi eksak, dicari solusi $\Delta\theta$ yang mendekati dalam arti *least squares* dengan meminimalkan:

$$\min_{\Delta\theta} \|J\Delta\theta - G\|_2^2$$

dengan mendefinisikan:

$$E(\Delta\theta) = 2J^T(J\Delta\theta - G) = 0$$

kemudian $E(\Delta\theta)$ terhadap $\Delta\theta$ dan menyamakan gradienya dengan nol, lalu diperoleh:

$$\nabla E(\Delta\theta) = 2J^T(J\Delta\theta - G) = 0$$

Dari sini didapatkan persamaan normal:

$$[J^T J]\Delta\theta = [J^T G] \quad (2.25)$$

Persamaan diatas merupakan sistem persamaan linier berukuran $n \times n$ yang solusinya memberikan $\Delta\theta$ terbaik dalam arti least squares terhadap sistem semula $J\Delta\theta \approx G$.

Namun menyelesaikan persamaan (2.25) secara langsung akan menyebabkan ketidakstabilan numerik, karena matrix $J^T J$ dapat menjadi *ill-conditioned* dan memperbesar kesalahan pembulatan. Untuk meningkatkan stabilitas numerik, pada penelitian ini digunakan dekomposisi QR pada matrix J sebagaimana pada Persamaan (2.21).

Dengan menggunakan Dekomposisi QR, dapat mensubstitusikan persamaan (2.21) ke dalam persamaan (2.25) sebagai berikut:

$$[J^T J]\Delta\theta = [J^T G]$$

$$[(QR)^T (QR)]\Delta\theta = [(QR)^T G]$$

karena $(QR)^T = R^T Q^T$, diperoleh:

$$[R^T Q^T Q R]\Delta\theta = [R^T Q^T G]$$

kemudian Q adalah matrix ortogonal seperti yang telah didefinisikan pada (Golub & Van Loan, 2013) maka $Q^T Q = I$, sehingga persamaan diatas dapat disederhanakan menjadi:

$$[R^T I R]\Delta\theta = [R^T Q^T G]$$

karena R diasumsikan invertibel, kedua ruas dapat dikalikan dari kiri dengan $(R^T)^{-1}$, sehingga diperoleh sistem yang lebih sederhana:

$$R\Delta\theta = Q^T G$$

Selanjutnya didefinisikan $s = Q^T G$, sehingga persamaan dapat ditulis dalam bentuk yang lebih sederhana:

$$R\Delta\theta = s \quad (2.26)$$

karena R adalah matrix segitiga atas, sistem persamaan ini dapat diselesaikan dengan menggunakan metode substitusi mundur (*back-substitution*), yaitu dengan menyelesaikan satu per satu elemen dari $\Delta\theta$ dimulai dari elemen terakhir hingga elemen pertama. Misalkan R berbentuk sebagai berikut:

$$R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & r_{nn} \end{bmatrix}$$

maka sistem $R\Delta\theta = s$ dapat diselesaikan dengan:

$$\begin{aligned} \Delta\theta_n &= \frac{s_n}{r_{nn}} \\ \Delta\theta_{n-1} &= \frac{s_{n-1} - r_{(n-1)n}\Delta\theta_n}{r_{(n-1)(n-1)}} \\ &\vdots \\ \Delta\theta_1 &= \frac{s_1 - (r_{12}\Delta\theta_2 + \cdots + r_{1n}\Delta\theta_n)}{r_{11}} \end{aligned}$$

Dengan demikian mendapatkan solusi $\Delta\theta$ yang kemudian digunakan untuk memperbarui parameter regresi logistik. Dalam konteks ini, n mempresentasikan jumlah parameter dalam sistem, yaitu jumlah elemen dalam vektor θ . Pembaruan parameter dilakukan dengan rumus

$$\Delta\theta^{k+1} = \Delta\theta^k + \Delta\theta$$

di mana $\Delta\theta$ adalah solusi dari sistem *least-squares*. Proses ini diulangi hingga konvergensi tercapai, yaitu ketika

$$||\Delta\theta|| < \varepsilon$$

dengan ε merupakan toleransi terkecil yang menyatakan bahwa perubahan parameter sudah cukup kecil atau residual telah memenuhi kriteria yang ditetapkan.

2.2 Keutamaan Orang Berilmu

Dalam perspektif Islam, Al-Qur'an menjadi rujukan utama dalam memperoleh ilmu pengetahuan, karena kebenarannya bersifat mutlak dan diturunkan oleh Allah SWT Sang Pencipta dan Pemberi ilmu. Al-Qur'an menetapkan arah dan tujuan pendidikan yang integral, yang mencakup aspek-aspek kehidupan secara menyeluruh, termasuk proses pembelajaran (Lubis & Asry, 2020).

Bahkan, wahyu pertama yang diturunkan kepada Nabi Muhammad SAW menjadi titik awal perintah Allah kepada umat manusia agar senantiasa belajar, membaca, dan memahami ilmu sebagai bentuk ibadah. Wahyu tersebut disampaikan melalui malaikat Jibril dan termaktub dalam Surah Al-'Alaq ayat 1–5 (Nata, 2016). Ayat-ayat tersebut menegaskan bahwa Islam memberikan perhatian besar terhadap proses pembelajaran dan pencarian ilmu. Oleh karena itu, dalam ajaran Islam, menuntut ilmu diwajibkan bagi setiap individu, baik laki-laki maupun perempuan (Nurjan, 2016). Adapun surah Al-'Alaq ayat 1-5 berbunyi:

اقْرَأْ بِاسْمِ رَبِّكَ الَّذِي خَلَقَ (١) خَلَقَ الْإِنْسَانَ مِنْ عَلَقٍ (٢) اقْرَأْ وَرَبُّكَ الْأَكْرَمُ (٣) الَّذِي عَلَّمَ بِالْقَلَمِ
(٤) عَلَّمَ الْإِنْسَانَ مَا لَمْ يَعْلَمْ (٥)

(Kementerian Agama, 2017)

Artinya: "Bacalah dengan (menyebut) nama Tuhanmu yang menciptakan. Dia menciptakan manusia dari 'alaq. Bacalah, dan Tuhanmu-lah Yang Maha Pemurah. Yang mengajar manusia dengan pena, mengajar manusia apa yang tidak diketahuinya." (QS.Al-'Alaq:1-2)

Berdasarkan tafsir Ibnu Katsir, ayat-ayat ini menandai awal turunnya rahmat Allah kepada hamba-Nya, di mana Dia mengajarkan manusia melalui pena, memungkinkan mereka mempelajari hal-hal yang sebelumnya tidak diketahui. Sebagai bentuk peringatan (*tanbih*), surah ini mengisyaratkan bahwa Allah Ta'ala memulai penciptaan manusia dari 'alaq, yakni segumpal darah, melalui kekuasaan-Nya yang mutlak (Mukmin, 2016).

Oleh sebab itu, ayat ini mengajarkan bahwa Allah memiliki kuasa untuk menjadikan manusia mampu membaca, meskipun sebelumnya tidak memiliki pengetahuan. Manusia diciptakan oleh Allah dari segumpal darah dan dikaruniai akal untuk berpikir, sehingga ia dapat menguasai berbagai makhluk di bumi. Sebagaimana lisan digunakan untuk berbicara, Allah menjadikan pena sebagai media komunikasi antar manusia. Qalam yang dimaksud dalam ayat tersebut adalah benda tak bernyawa yang secara alami tidak bisa menyampaikan pengertian. Namun, dengan kehendak-Nya, Allah menjadikan benda mati sebagai alat komunikasi. Hal tersebut menunjukkan bahwa Allah menciptakan manusia dari 'alaq, segumpal darah yang tampak tak berarti, namun meninggikannya dengan ilmu melalui perantara pena, agar manusia menyadari asal-usulnya yang rendah hingga mencapai kesempurnaan sebagai makhluk yang berpengetahuan dan memahami hakikat segala sesuatu (Wahidi, 2016).

Ibnu Katsir rahimahullah juga berkata, "Seseorang itu akan semakin mulia dengan ilmu yang ia miliki. Ilmu itulah yang membedakan bapak manusia, yaitu Adam dengan para malaikat. Ilmu ini terkadang di pikiran. Ilmu juga kadang di lisan. Ilmu juga terkadang di dalam tulisan tangan untuk menyalurkan apa yang dalam pikiran, lisan, maupun yang tergambarkan di pikiran." (Mukmin, 2016).

Ibnu Katsir rahimahullah menjelaskan bahwa kemuliaan seseorang terletak pada ilmu, terutama ilmu agama, yang menjadi pembeda antara manusia dan makhluk lain, termasuk para malaikat. Nabi Adam diberikan keistimewaan oleh Allah dengan ilmu yang diajarkan langsung kepadanya, sehingga ia memiliki kedudukan lebih tinggi dibandingkan malaikat, meskipun malaikat merupakan makhluk yang selalu patuh dan tidak pernah melanggar perintah (Syakir, 2012).

Ilmu yang dimiliki manusia dapat muncul dalam tiga bentuk utama. Pertama, ilmu yang tersimpan dalam pikiran, yaitu pemahaman dan wawasan yang diperoleh melalui pembelajaran dan pengalaman. Kedua, ilmu yang diungkapkan melalui lisan, yaitu ilmu yang disampaikan kepada orang lain melalui ucapan, ceramah, diskusi, atau pengajaran. Ketiga, ilmu yang ditulis dalam bentuk catatan, buku, atau tulisan lainnya, yang bertujuan untuk mendokumentasikan serta menyebarluaskan pengetahuan agar dapat dimanfaatkan oleh banyak orang dan diwariskan kepada generasi mendatang (Octaviana & Ramadhani, 2021).

Tulisan memiliki peran penting dalam menjaga keberlangsungan ilmu, karena tanpa dokumentasi, ilmu yang hanya ada dalam pikiran atau lisan bisa hilang seiring berjalannya waktu. Oleh karena itu, Islam sangat menganjurkan pembelajaran dan pencatatan ilmu sebagai bentuk penjagaan terhadap pengetahuan, sebagaimana disebutkan dalam berbagai hadis dan atsar para ulama. Dengan adanya tiga bentuk ilmu ini, manusia dapat mengembangkan dirinya, berbagi manfaat dengan sesama, serta menjaga dan meneruskan ilmu agar tidak punah.

2.3 Kajian Topik dan Teori Pendukung

Pada penelitian ini, berbagai metode analisis diterapkan untuk mengidentifikasi variabel-variabel yang memengaruhi tingkat keselamatan penumpang kapal Titanic. Langkah awal dalam penelitian ini adalah pengumpulan data, di mana peneliti mengumpulkan dataset yang berisi informasi karakteristik penumpang Titanic, seperti umur, jenis kelamin, kelas kabin, serta status keselamatannya. Dataset tersebut tersedia secara publik dan dapat diakses melalui platform Kaggle. Keakuratan serta relevansi data menjadi aspek penting guna menghasilkan hasil analisis yang valid.

Langkah kedua mencakup identifikasi terhadap nilai hilang maupun nilai-nilai anomali yang tidak sesuai dengan distribusi wajar. Langkah pembersihan data mencakup melengkapi yang hilang, menghapus outlier, dan mengubah variabel kategori menjadi numerik untuk analisis regresi logistik. Eksplorasi dan pembersihan data berperan penting agar model yang dibangun terbebas dari pengaruh kesalahan atau ketidakkonsistenan data.

Langkah ketiga adalah pemilihan model, penelitian ini menetapkan regresi logistik sebagai pendekatan analisis yang dianggap paling sesuai untuk mengolah data yang tersedia. Pemilihan model ini didasarkan pada kemampuannya untuk menangani variabel dependen yang bersifat biner, serta kemampuannya dalam mengestimasi probabilitas keselamatan berdasarkan sejumlah variabel prediktor. Selain itu, metode Gauss-Newton dan Dekomposisi QR akan ditinjau guna memperoleh estimasi parameter model yang lebih akurat.

Langkah selanjutnya adalah pelatihan model, data pelatihan digunakan untuk membangun model regresi logistik. Dalam proses ini, dilakukan estimasi

parameter model menggunakan algoritma Gauss-Newton, dengan bantuan Dekomposisi QR untuk menjaga stabilitas numerik selama proses perhitungan. Setelah pelatihan selesai, data pengujian digunakan untuk mengevaluasi performa model yang bertujuan untuk mengukur tingkat akurasi model dalam memprediksi keselamatan penumpang.

Langkah terakhir adalah analisis hasil, di mana peneliti akan meninjau output dari model regresi logistik yang telah dilatih sebelumnya. Hasil analisis akan disajikan dalam bentuk tabel dan grafik, penyajian visual ini dimaksudkan untuk mempermudah pemahaman dan interpretasi terhadap pengaruh masing-masing faktor.

BAB III

METODE PENELITIAN

3.1 Jenis Penelitian

Penelitian ini menggunakan pendekatan kuantitatif berbasis komputasi yang bertujuan untuk mengoptimalkan model regresi logistik untuk klasifikasi biner melalui penerapan metode Gauss-Newton dan Dekomposisi QR. Melalui pendekatan ini, regresi logistik digunakan sebagai metode utama untuk mengevaluasi signifikan variabel melalui estimasi koefisien yang dihasilkan. Selain itu, penelitian ini mengimplementasikan metode optimasi numerik, yaitu algoritma Gauss-Newton dan Dekomposisi QR, dengan tujuan meningkatkan akurasi estimasi parameter serta menjaga stabilitas komputasi model.

3.2 Data dan Sumber Data

Data yang digunakan dalam penelitian ini merupakan data sekunder berupa dataset Titanic yang tersedia pada *platform Kaggle*. Dataset yang digunakan dalam penelitian ini terdiri dari 1309 sampel dan mencakup 8 fitur. Dalam penelitian ini, variabel yang digunakan terdiri dari satu variabel respon (Y) dan tujuh variabel prediktor (X) yaitu: *sex*, *fare*, *Pclass*, *Embarktion*, *age*, *sibps*, *parch*. Pemilihan variabel didasarkan pada relevansi informasi yang dimiliki masing-masing fitur terhadap status keselamatan penumpang (*survived*) disajikan sebagai berikut.

3.3 Tahapan Penelitian

Tahapan penelitian yang akan dilakukan, antara lain:

3.3.1 Persiapan Data

Tahapan persiapan data mencakup berbagai proses penting, berikut akan dijelaskan secara rinci:

1. Mengidentifikasi dan menangani nilai-nilai yang hilang pada data dengan imputasi median, menggunakan *SimpleImputer median*.
2. Bagi dataset ke dalam tiga subset utama: data pelatihan (70%), data validasi (15%), dan data pengujian (15%). Pembagian ini bertujuan untuk mengoptimalkan pelatihan model, menyetel parameter secara objektif, serta mengevaluasi performa akhir model dengan adil.
3. Sebelum analisis dilakukan, beberapa fitur (*sex*, *embarked*, *destination*, *cabin class*, *pclass*) dinormalisasi agar nilainya berada pada rentang 0–1 menggunakan metode *min-max normalization* dengan rumus:

$$x'_{ij} = \frac{x_{ij} - \min(\vec{x}_j)}{\max(\vec{x}_j) - \min(\vec{x}_j)}$$

4. *One-hot encoding* pada variabel kategorikal seperti *sex* dan *embarked* agar dapat digunakan dalam model *machine learning*.

3.3.2 Penerapan Model Regresi Logistik Biner

Dalam penerapannya, regresi logistik biner digunakan untuk menilai pengaruh variabel prediktor terhadap kemungkinan terjadinya suatu kejadian tertentu. Model ini mengelompokkan hasil ke dalam dua kategori, yaitu selamat (1) atau tidak selamat (0). Berikut adalah tahapan dalam proses pelatihan model regresi logistik:

1. Hitung nilai z_i menggunakan Persamaan (2.1), yaitu: $z_i = \mathbf{w}^T \cdot \mathbf{x}_i + b$.

2. Menghitung probabilitas $\sigma(z_i)$ menggunakan Persamaan (2.2), yaitu:

$$P(\mathbf{y}_i = 1|x_i) = \sigma(z_i) = \frac{1}{1+\exp(-z_i)}.$$

3. Hitung fungsi *loss (likelihood)* dengan mendefinisikan fungsi binary *cross-entropy* untuk setiap sampel i menggunakan Persamaan (2.3), yaitu: $\mathcal{L}_i = -[y_i \log(\sigma(z_i)) + (1-y_i)\log(1-\sigma(z_i))]$.
4. Setelah menghitung fungsi *loss*, minimalkan fungsi *loss* menggunakan metode Gauss-Newton dan Dekomposisi QR.

3.3.3 Penerapan Metode Gauss-Newton

Metode Gauss-Newton digunakan untuk menyetel parameter $\theta = [w, b]$ pada regresi logistik agar *binary cross-entropy* minimum, menggunakan pendekatan Gauss-Newton yang efisien tanpa menghitung Hessian penuh.. Adapun tahapan dalam penerapan metode ini adalah sebagai berikut:

1. Membentuk residual menggunakan Persamaan (2.10) dari probabilitas yang telah dihitung.
2. Menyusun matrix Jacobian J seperti pada Persamaan (2.18).
3. Gabungkan semua data menjadi sistem linear bentuk matrik pada Persamaan (2.20), yaitu : $J\Delta\theta = G$.
4. Selesaikann masalah least squares Persamaan (2.20) menggunakan Dekomposisi QR.

3.3.4 Penerapan Metode Dekomposisi QR

Penerapan Dekomposisi QR dalam regresi logistik bertujuan untuk memperbaiki kestabilan numerik dengan menata ulang matrix input X menjadi

bentuk yang lebih efisien untuk perhitungan. Proses ini dilakukan dengan memfaktorisasi matrix X menjadi dua komponen. Adapun tahapan implementasinya adalah sebagai berikut:

1. Faktorisasi matrix jacobian J ke Persamaan (2.21) yaitu: $J = QR$
2. Selesaikan langkah $\Delta\theta$ menggunakan *back-substitution* menggunakan Persamaan (2.26).
3. Perbarui parameter model secara iteratif menggunakan persamaan:

$$\theta^{(k+1)} = \theta^{(k)} + \Delta\theta$$
4. Ulangi Proses hingga konvergensi tercapai, yaitu ketika perubahan parameter cukup kecil atau nilai residual sudah memenuhi toleransi yang ditetapkan.

3.3.5 Evaluasi Performa Metode Optimasi

Evaluasi performa model klasifikasi berfokus pada seberapa baik model membuat prediksi untuk menilai kualitas hasil prediksi dari model regresi logistik terhadap data yang diberikan. Dalam penelitian ini, evaluasi dilakukan dengan menerapkan sejumlah metrik utama, yaitu akurasi, presisi, *recall*, *F1-score*.

Masing-masing metrik memiliki fungsi yang berbeda dalam menggambarkan kekuatan dan kelemahan model, khususnya pada kasus klasifikasi dengan distribusi kelas yang mungkin tidak seimbang. Berikut adalah penjelasan rinci dari tiap metrik:

1. Akurasi (*Accuration*)

Akurasi mengukur proporsi prediksi yang benar terhadap total keseluruhan prediksi, dan dihitung menggunakan rumus:

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

di mana:

TP (*True Positive*) = Kasus positif yang diprediksi benar.

TN (*True Negative*) = Kasus negatif yang diprediksi benar.

FP (*False Positive*) = Kasus negatif yang salah diprediksi sebagai positif.

FN (*False Negative*) = Kasus positif yang salah diprediksi sebagai negatif.

Akurasi kurang representatif bila terjadi ketidakseimbangan kelas, seperti jumlah penumpang selamat jauh lebih sedikit daripada yang tidak selamat.

2. Presisi (*precision*)

Presisi menunjukkan ketepatan model dalam memprediksi kelas positif dan dihitung dengan rumus:

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

Presisi tinggi berarti sedikit kesalahan positif (*false positive*). Metrik ini sangat penting ketika kesalahan dalam mengklasifikasikan data negatif sebagai positif harus dihindari, misalnya dalam deteksi kondisi kritis atau penumpang yang diprediksi selamat padahal tidak.

3. Recall (*Sensitivity* atau *True Positive Rate* - TPR)

Recall mengukur seberapa banyak kasus positif yang berhasil diidentifikasi model, dengan rumus:

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

Recall sangat penting dalam konteks prediksi keselamatan penumpang, karena kesalahan dalam tidak mendeteksi penumpang yang seharusnya selamat (FN) menurunkan keandalan model dalam mengidentifikasi kelompok berisiko.

4. *F1-score*

F1-score adalah rata-rata harmonik antara presisi dan *recall*, digunakan saat diperlukan keseimbangan antara keduanya, terutama pada data yang tidak seimbang.

$$F1score = 2 \times \frac{precision \times recall}{precision + recall}$$

F1-score memberikan gambaran kinerja model dalam menangani data minoritas secara keseluruhan.

Berdasarkan temuan akurasi, presisi, *recall*, dan *F1-score* yang telah dihitung, selanjutnya menyajikan perbandingan kinerja antara Gauss-Newton, Adam, dan SGD di bawah pengaturan yang setara, mencakup pra-proses, *split stratified*, kriteria berhenti, dan evaluasi pada *validation/test*. Perbandingan mencakup akurasi, presisi, *recall*, *F1-score*, serta indikator efisiensi seperti jumlah epoch/iterasi dan waktu latih. Kurva pembelajaran (*loss* dan akurasi) digunakan untuk melihat laju penurunan, stabilitas konvergensi. Hasilnya diinterpretasikan untuk menilai *trade-off* kecepatan kinerja, dengan fokus apakah Gauss-Newton benar-benar lebih cepat konvergen sekaligus kompetitif/unggul pada metrik klasifikasi dibanding Adam dan SGD.

Hasil penelitian ini kemudian dikontekstualisasikan dengan studi-studi Titanic terdahulu yang melaporkan metrik serupa (akurasi, presisi, *recall*, *F1-*

score). Pembahasan menyoroti perbedaan prosedur (skema *split*, imputasi/encoding, *feature engineering* seperti *Title/FamilySize*, dan ambang keputusan) agar perbandingan adil. Bila tersedia, aspek waktu komputasi turut dibahas untuk menilai *trade-off* kinerja efisiensi yang kerap luput dilaporkan. Dengan kerangka ini, posisi Gauss-Newton ditunjukkan sebagai pendekatan yang kompetitif pada metrik dan unggul dalam kecepatan/konvergensi, sehingga relevan baik secara metodologis maupun praktis.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Deskripsi Dataset

Deskripsi dataset memaparkan sumber dan karakteristik Titanic Dataset yang digunakan sebagai landasan eksperimen. Disajikan ringkas komposisi data, fitur yang dipilih beserta target (*Survived*), serta alasan seleksi fitur dengan menghindari *data leakage*. Statistik deskriptif awal dan tingkat ketidakseimbangan kelas diperlihatkan untuk memberi konteks terhadap strategi praproses (imputasi, encoding, penskalaan). Terakhir, dijelaskan protokol pembagian data (*train/validation/test*) berbasis *stratified split* berikut *random seed* agar seluruh analisis reproduisibel dan bebas kebocoran informasi.

4.1.1 Sumber Data dan Karakteristik Data

Penelitian ini menggunakan Titanic Dataset yang diperoleh dari laman *website data online* Kaggle dalam format excel. Data tersebut berisi sebanyak 14 fitur dan 1.309 data. Fitur yang digunakan pada data tersebut sebanyak 7 fitur dari 14 fitur, yaitu : *sex, fare, Passenger Class, Age, Siblings/Spouses, Parent/Children, Embarkation*. Seluruh data asli penumpang kapal Titanic dapat dilihat secara lengkap pada lampiran 1.

4.1.2 Fitur dan Target yang Digunakan

Dataset Titanic yang digunakan dalam penelitian ini awalnya memiliki 14 atribut (fitur) beserta 1 label target yaitu *survived*. Namun, tidak seluruh fitur dapat dimanfaatkan dalam proses pemodelan karena terdapat beberapa atribut yang bersifat identitas unik maupun atribut yang mengandung data *leakage*

(informasi yang hanya diketahui setelah kejadian). Oleh karena itu, dilakukan tahap seleksi fitur untuk menentukan variabel yang relevan serta berpengaruh terhadap prediksi keselamatan penumpang. Sebagai tindak lanjut seleksi fitur, beberapa atribut dikeluarkan karena tidak memenuhi kriteria kelayakan (misalnya kardinalitas sangat tinggi, potensi *data leakage*, atau tingkat *missing* yang besar).

Untuk memastikan kualitas model regresi logistik, dilakukan evaluasi terhadap setiap fitur pada dataset Titanic. Proses ini bertujuan menentukan fitur mana yang layak digunakan berdasarkan relevansi statistik, kestabilan representasi data, dan risiko data leakage. Tabel 4.1 merangkum status penggunaan fitur, nilai korelasi dengan variabel target (r), serta alasan pemilihan atau pengecualian fitur.

Tabel 4.1 Hasil Korelasi Pearson untuk Seluruh Data terhadap Survived

Status	Fitur	r	Alasan
Digunakan	<i>sex</i>	0.5287	Prediktor terpenting, tidak ada masalah
Digunakan	<i>pclass</i>	0.3125	Indikator status sosial valid
Digunakan	<i>fare</i>	0.2444	Signifikan, tidak ada masalah
Digunakan	<i>embarked</i>	0.1753	Signifikan, missing minimal
Digunakan	<i>parch</i>	0.0827	Valid untuk interaksi fitur
Digunakan	<i>age</i>	0.0438	Relevan domain <i>knowledge</i>
Digunakan	<i>sibsp</i>	0.0278	Valid untuk model non-linear
Tidak	<i>boat</i>	0.9482	Data leakage
Tidak	<i>name</i>	0.3525	Kardinalitas tinggi, <i>overfitting</i>
Tidak	<i>cabin</i>	0.3023	Missing 77.5%
Tidak	<i>home.dest</i>	0.1982	Kardinalitas tinggi, <i>missing</i> 43.1%
Tidak	<i>ticket</i>	-0.1273	Representasi tidak stabil
Tidak	<i>body</i>	-0.0143	Data leakage, missing 90.8%

Berdasarkan hasil evaluasi, fitur *sex* memiliki korelasi tertinggi terhadap *survived* ($r = 0.5287$) dan menjadi prediktor yang paling menentukan sehingga tetap digunakan. Fitur *pclass*, *fare*, dan *embarked* juga menunjukkan hubungan yang cukup kuat dan stabil, sehingga layak dipertahankan dalam model. Fitur

parch, *age*, dan *sibsp* meskipun nilai korelasinya lebih rendah, tetap digunakan karena memiliki relevansi berdasarkan pengetahuan domain dan berpotensi memberikan kontribusi pada model terutama saat interaksi fitur atau struktur non-linear dimasukkan.

Sebaliknya, beberapa fitur dikeluarkan karena berisiko menyebabkan data leakage, yakni *boat* dan *body*, yang diketahui secara langsung terkait informasi setelah kejadian. Fitur seperti *name*, *home.dest*, dan *ticket* juga tidak digunakan karena tingkat kardinalitas yang tinggi atau representasinya tidak stabil sehingga dapat memicu *overfitting*. Adapun *cabin* dikeluarkan karena proporsi missing value yang sangat besar sehingga tidak layak dipertahankan.

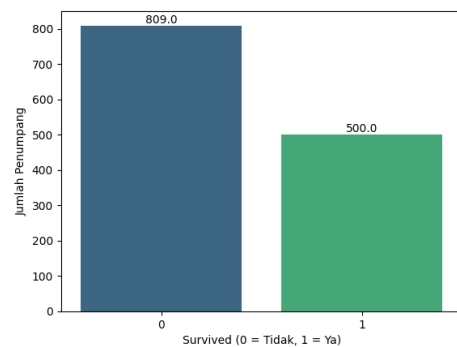
Tabel 4.2 menyajikan daftar fitur keputusan yang digunakan dalam pemodelan, meliputi satu variabel respon (target) dan tujuh variabel prediktor beserta keterangannya. Penyajian ini dimaksudkan untuk memperjelas ruang lingkup fitur yang dianalisis pada tahap eksperimen.

Tabel 4.2 Variabel Respon dan Prediktor

No	Variabel	Keterangan	Tipe Data
A	Variabel Respon		
1	<i>Survived</i> (y)	Status Keselamatan	Numerik
B	Variabel Prediktor		
1	<i>Sex</i> (x_1)	Jenis kelamin	Kategorik
2	<i>Fare</i> (x_2)	Harga tiket	Numerik
3	<i>Parch</i> (x_3)	Jumlah orang tua/anak di kapal	Numerik
4	<i>Pclass</i> (x_4)	Kelas tiket (1, 2, 3)	Numerik
5	<i>Embarked</i> (x_5)	Pelabuhan naik	Kategorik
6	<i>Age</i> (x_6)	Usia Penumpang	Numerik
7	<i>Sibsp</i> (x_7)	Jumlah saudara/istri di kapal	Numerik

4.1.3 Statistik Deskriptif dan Ketidakseimbangan Kelas

Gambar 4.1 menyajikan ringkasan deskriptif dari variabel target *Survived* (1 = selamat, 0 = tidak selamat) untuk menilai keseimbangan kelas sebagai dasar penetapan strategi evaluasi dan praproses.



Gambar 4.1 Distribusi Kelas Target

Visualisasi menunjukkan perbedaan yang jelas antara jumlah penumpang yang tidak selamat dan yang selamat. Dari total 1.309 penumpang, terdapat 809 penumpang (61,8%) tidak selamat dan 500 penumpang (38,2%) selamat. Rasio ketidakseimbangan (Imbalance Ratio, IR) diperoleh sebesar:

$$IR = \frac{\max(n_0, n_1)}{\min(n_0, n_1)} = \frac{809}{500} = 1,618$$

Nilai $IR \approx 1,618$ mengindikasikan ketidakseimbangan moderat, kelas mayoritas (tidak selamat) sekitar 1,6 kali lebih banyak daripada kelas minoritas (selamat). Kondisi ini penting dicatat karena akurasi sederhana yang selalu menebak kelas mayoritas saja sudah mencapai kira-kira 61,8%, sehingga metrik akurasi tidak cukup representatif.

Untuk itu, evaluasi kinerja pada bab hasil akan menekankan Presisi, *Recall*, *F1-score*, serta *confusion matrix*. Selain itu, pembagian data dilakukan secara stratified agar proporsi kelas terjaga pada setiap split (*train/validation/test*). Penanganan ketidakseimbangan seperti penyesuaian ambang keputusan dan/atau

class weighting pada fungsi *loss* juga dipertimbangkan agar model tidak bias ke kelas mayoritas.

4.1.4 Pembagian Data

Pada studi ini, dataset Titanic terlebih dahulu dibagi (*data splitting*) menjadi tiga subset yang saling lepas, yaitu: *train* 70%, *validation* 15%, dan *test* 15% menggunakan *stratified split*, yaitu metode pemisahan data yang mempertahankan proporsi kelas target secara konsisten pada setiap subset. Teknik ini diterapkan karena data Titanic memiliki distribusi kelas yang tidak seimbang, di mana jumlah penumpang yang tidak selamat lebih banyak dibanding penumpang yang selamat. Jika pembagian dilakukan secara acak biasa tanpa stratifikasi, terdapat risiko bahwa salah satu subset (misalnya *validation* atau *test*) tidak mewakili distribusi kelas sebenarnya, sehingga model dapat menjadi bias terhadap kelas mayoritas dan menghasilkan evaluasi performa yang menyesatkan. Variabel target *survived* terlebih dahulu diubah ke tipe bilangan bulat dan dipisahkan dari fitur sebelum pemisahan dilakukan. Secara fungsional, Statistik praproses (imputasi, penskalaan) dipelajari dari *train* lalu diaplikasikan ke *validation* dan *test* untuk mencegah data *leakage*.

Dengan demikian, *stratified splitting* menjamin bahwa seluruh subset memiliki distribusi kelas yang konsisten dan realistis, sehingga pelatihan dan evaluasi model dapat dilakukan secara objektif dan stabil. Hasil data *splitting* ditampilkan pada Tabel 4.3 di bawah ini.

Tabel 4.3 Hasil Data Splitting

Subset	Jumlah Baris	Proporsi dari Total
<i>Train</i>	916	$\approx 70,03\%$
<i>Validation</i>	196	$\approx 14,98\%$
<i>Test</i>	197	$\approx 15,05\%$
Total	1.309	100%

Pada Tabel 4.3 tersebut menunjukkan ringkasan pembagian data Titanic ke tiga subset yang saling lepas dengan skema fitur yang identik (7 kolom) di seluruh subset. Sebanyak 916 baris ($\approx 70,03\%$) dialokasikan untuk *train* sebagai bahan utama pelatihan model, 196 baris ($\approx 14,98\%$) untuk *validation* sebagai dasar penyetelan hiperparameter dan pemilihan model, serta 197 baris ($\approx 15,05\%$) untuk *test* yang disimpan khusus guna evaluasi akhir kemampuan generalisasi. Jumlah keseluruhan mencapai 1.309 baris, dengan proporsi yang mendekati target 70% : 15% : 15%; selisih satu baris antara *validation* dan *test* merupakan konsekuensi wajar dari proses pembulatan ketika membagi sisa 30% data menjadi dua bagian hampir sama besar.

4.2 Hasil

Hasil akan menyajikan hasil eksperimen dari dua skala, yaitu studi perhitungan manual pada contoh kecil untuk menelusuri langkah Gauss–Newton dan Dekomposisi QR secara transparan, dan eksperimen penuh pada seluruh data. Disertakan rincian pengaturan eksperimen (*pra-proses*, *encoding*, kriteria henti), lalu ditunjukkan konvergensi dan waktu pelatihan guna menilai efisiensi. Selanjutnya dilaporkan kinerja klasifikasi (akurasi, presisi, *recall*, *F1-Score*), diikuti perbandingan dengan *baseline* (SGD dan Adam).

4.2.1 Studi Perhitungan Manual (Contoh Kecil)

Pada penelitian ini penulis akan menerapkan Metode Gauss–Newton dan dekomposisi QR dalam proses optimasi pada model regresi logistik. Untuk memperjelas alur perhitungan, akan disajikan Tabel 4.4 yang berisi data sampel dengan 6 data dengan 3 fitur yang memiliki korelasi tinggi dari dataset titanic, agar proses perhitungan manual dapat dilakukan.

Tabel 4.4 Data Sampel dengan 6 Data dengan 3 Fitur yang Memiliki Korelasi Tinggi dari Dataset Titanic

i	x			y
	x_1 (sex)	x_2 (fare)	x_3 (parch)	
1	1	211,3375	0	1
2	0	151,5500	2	1
3	1	151,5500	2	0
4	0	151,5500	2	0
5	1	151,5500	2	0
6	0	26,5500	0	1

1. Normalisasi Fitur

Sebelum melakukan analisis lebih lanjut, data Titanic perlu melalui tahap normalisasi agar setiap fitur memiliki skala yang sebanding dan tidak menimbulkan bias pada model. Normalisasi dilakukan dengan cara mengubah nilai setiap data ke dalam rentang tertentu, biasanya $[0,1]$. Dengan demikian, setiap nilai fitur akan dicari nilai minimum dan maksimum masing-masing fitur, kemudian setiap nilai dihitung ulang dicari nilai minimum dan maksimum masing-masing fitur, kemudian setiap nilai dihitung ulang menggunakan rumus *Min-Max Normalization* sebagai berikut:

$$x'_{ij} = \frac{x_{ij} - \min(\vec{x}_j)}{\max(\vec{x}_j) - \min(\vec{x}_j)}$$

Contoh perhitungan normalisasi fitur normalisasi fitur x_{ij} diambil berdasarkan Tabel 4.4 untuk $i = 1, 2, \dots, 6$ dan $j = 1, 2, 3$ yang akan didefinisikan sebagai berikut

$$\vec{x}_1 = (1; 0; 1; 0; 1; 0)$$

$$\vec{x}_2 = (211,3375; 151,5500; 151,5500; \\ 151,5500; 151,5500; 26.5500)$$

$$\vec{x}_3 = (0, 2, 2, 2, 2, 0)$$

Dengan keterangan $\min \vec{x}_1 = 0$, $\max \vec{x}_1 = 1$, $\min \vec{x}_2 = 26.5500$, $\max \vec{x}_2 = 211,3375$, $\min \vec{x}_3 = 0$, dan $\max \vec{x}_3 = 2$.

a. Normalisasi untuk x_{i1}

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

$$\begin{aligned} x'_{11} &= \frac{x_{11} - \min(\vec{x}_1)}{\max(\vec{x}_1) - \min(\vec{x}_1)} \\ &= \frac{1 - 0}{1 - 0} \\ &= 1 \end{aligned}$$

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

$$\begin{aligned} x'_{21} &= \frac{x_{21} - \min(\vec{x}_1)}{\max(\vec{x}_1) - \min(\vec{x}_1)} \\ &= \frac{0 - 0}{1 - 0} \\ &= 0 \end{aligned}$$

b. Normalisasi untuk x_{i2}

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

$$x'_{12} = \frac{x_{12} - \min(\vec{x}_2)}{\max(\vec{x}_2) - \min(\vec{x}_2)}$$

$$\begin{aligned}
&= \frac{211,3375 - 26.500}{211,3375 - 26.500} \\
&= 1
\end{aligned}$$

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

$$\begin{aligned}
x'_{22} &= \frac{x_{22} - \min(\vec{x}_2)}{\max(\vec{x}_2) - \min(\vec{x}_2)} \\
&= \frac{151,5500 - 26.500}{211,3375 - 26.500} \\
&= 0,6765
\end{aligned}$$

c. Normalisasi untuk x_{i3}

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

$$\begin{aligned}
x'_{13} &= \frac{x_{13} - \min(\vec{x}_3)}{\max(\vec{x}_3) - \min(\vec{x}_3)} \\
&= \frac{0 - 0}{2 - 0} \\
&= 0
\end{aligned}$$

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

$$\begin{aligned}
x'_{23} &= \frac{x_{23} - \min(\vec{x}_3)}{\max(\vec{x}_3) - \min(\vec{x}_3)} \\
&= \frac{2 - 0}{2 - 0} \\
&= 1
\end{aligned}$$

d. Hasil Normalisasi

Setelah proses perhitungan normalisasi dilakukan, seluruh nilai pada fitur telah berhasil ditransformasikan ke dalam rentang $[0,1]$. Hasil normalisasi ini menunjukkan bagaimana setiap data pada fitur. Dengan demikian, data yang awalnya memiliki skala berbeda kini berada pada

skala yang seragam, sehingga dapat digunakan secara lebih adil dalam proses analisis maupun pemodelan. Tabel 4.5 berikut menyajikan hasil akhir normalisasi dari data yang digunakan sebagai sampel:

Tabel 4.5 Hasil Data Setelah Normalisasi

i	x			y
	x_1 (sex)	x_2 (fare)	x_3 (parch)	
1	1	1	0	1
2	0	0,6765	1	1
3	1	0,6765	1	0
4	0	0,6765	1	0
5	1	0,6765	1	0
6	0	0	0	1

2. Inisialisasi Parameter

Setelah seluruh fitur dinormalisasi sehingga berada pada skala yang seragam, tahap selanjutnya adalah melakukan inisialisasi bobot. Pemilihan bobot awal sangat berpengaruh terhadap konvergensi dan kinerja model, karena bobot yang terlalu besar atau terlalu kecil dapat menyebabkan proses pelatihan menjadi lambat atau bahkan tidak stabil. Oleh karena itu, pada penelitian ini bobot diinisialisasi untuk contoh manual dengan tiga fitur, digunakan parameter awal yaitu:

$$\theta^{(0)} = \begin{bmatrix} 0 \\ 0,1 \\ 0,2 \\ 0,3 \end{bmatrix}$$

Setelah bobot awal diinisialisasi, langkah berikutnya adalah menghitung nilai z_i untuk setiap data. Nilai z_i diperoleh dari kombinasi linier antara variabel input dengan bobot yang bersesuaian serta ditambahkan bias. Namun, dalam Tabel 4.5 nilai bias b tidak ditampilkan secara eksplisit. Dalam praktiknya, bias umumnya direpresentasikan sebagai $x_0 = 1$, sehingga tetap

terlibat dalam proses komputasi meskipun tidak dituliskan dalam tabel data fitur.

Dengan demikian pada saat perhitungan nilai z_i , bias tetap diperhitungkan melalui perkalian x_0 dengan bobot bias (θ_0) dan hasilnya dijumlahkan bersama dengan perkalian antara fitur lain dan bobotnya masing-masing. Oleh karena itu, perhitungan tetap lengkap, meskipun nilai x_0 tidak dicantumkan secara langsung pada tabel data fitur.

a. Menghitung nilai z_i

Untuk data dengan variabel numerik, nilai fitur dapat langsung dikalikan dengan koefisien parameter θ dalam perhitungan nilai kombinasi linear z_i . Namun, apabila terdapat variabel kategorik, variabel tersebut tidak memiliki skala numerik sehingga tidak dapat langsung digunakan dalam perhitungan regresi logistik. Oleh karena itu, variabel kategorik harus dikonversi terlebih dahulu ke dalam bentuk numerik melalui proses encoding sebelum dapat dimasukkan ke dalam model.

Pada penelitian ini terdapat dua variabel kategorik, yaitu *Sex* dan *Embarked*, yang secara alami tidak memiliki nilai numerik. Untuk mengatasi hal tersebut, digunakan metode *One-Hot Encoding*, di mana setiap kategori direpresentasikan dalam bentuk variabel biner bernilai 0 atau 1. Sebagai contoh, variabel *Sex* dikonversi menjadi variabel baru *sex*, yang bernilai 1 apabila penumpang berjenis kelamin perempuan dan bernilai 0 apabila laki-laki. Demikian pula, variabel *Embarked* yang memiliki tiga kategori (C, Q, dan S) direpresentasikan menjadi dua

variabel dummy: Emb_C dan Emb_Q , sementara kategori S dijadikan kategori referensi yang bernilai 0 pada kedua variabel dummy tersebut.

Setelah proses konversi ini selesai, seluruh variabel kategorik telah menjadi numerik sehingga diperlakukan sama seperti variabel numerik lainnya dalam perhitungan nilai linear kombinasi $z_i = \theta_0 + \theta_1 x_i + \dots + \theta_n x_n$. Tidak ada formula tambahan maupun perlakuan khusus untuk variabel kategorik dalam regresi logistik. Setiap dummy biner yang dihasilkan akan dikalikan langsung dengan parameter model θ , sama seperti variabel numerik yang lain. Dengan demikian, perhitungan kontribusi variabel kategorik di dalam model dilakukan secara identik dengan variabel numerik setelah proses encoding selesai.

Pernyataan ini sejalan dengan penjelasan dalam jurnal *Regression Analysis with Categorical Variables* oleh (Venkataramana dkk., 2016). Pada halaman 136, jurnal tersebut menyatakan bahwa variabel kualitatif dengan k kategori harus direpresentasikan menggunakan $(k-1)$ variabel dummy yang memiliki nilai numerik, sehingga variabel kualitatif dapat diubah menjadi kuantitatif “*to run the regression*”. Selanjutnya, jurnal tersebut memperlihatkan bahwa variabel dummy tersebut dimasukkan langsung dalam persamaan regresi linear, yaitu:

$$Y = B_0 + \sum B_j D_{ij} + \varepsilon,$$

yang menunjukkan bahwa setelah dikodekan menjadi variabel dummy bernilai 0/1, variabel kategorik berfungsi sama seperti variabel numerik dalam model regresi. Dengan demikian, literatur ini mendukung secara teoritis bahwa regresi baik linear maupun logistik memperlakukan variabel

kategorik yang telah di*encoding* sebagai variabel numerik penuh dalam pembentukan kombinasi linear model.

Dengan demikian, perhitungan nilai z_i tetap mengikuti persamaan umum regresi logistik. Hitung nilai z_i untuk $i = 1, 2, \dots, 6$ menggunakan persamaan (2.1):

1. Untuk $i = 1$,

$$\begin{aligned} z_1 &= \theta_0 + x_{11} \times \theta_1 + x_{12} \times \theta_2 + x_{13} \times \theta_3 \\ &= 0 + 1 \times 0,1 + 1 \times 0,2 + 0 \times 0,3 \\ &= 0,3 \end{aligned}$$

2. Untuk $i = 2$,

$$\begin{aligned} z_2 &= \theta_0 + x_{21} \times \theta_1 + x_{22} \times \theta_2 + x_{23} \times \theta_3 \\ &= 0 + 0 \times 0,1 + 0,6765 \times 0,2 + 1 \times 0,3 \\ &= 0,4353 \end{aligned}$$

3. Untuk $i = 3$,

$$\begin{aligned} z_3 &= \theta_0 + x_{31} \times \theta_1 + x_{32} \times \theta_2 + x_{33} \times \theta_3 \\ &= 0 + 1 \times 0,1 + 0,6765 \times 0,2 + 1 \times 0,3 \\ &= 0,5353 \end{aligned}$$

Dengan cara yang sama dilakukan untuk $i = 4, 5$, dan 6 . Hasil lengkap dari perhitungan nilai z_i untuk seluruh data ditunjukkan pada Tabel 4.6 di bawah ini.

Tabel 4.6 Hasil Perhitungan Nilai z_i

i	z_i
$i = 1$	0,3
$i = 2$	0,4353
$i = 3$	0,5353
$i = 4$	0,4353
$i = 5$	0,5353
$i = 6$	0

b. Menghitung $\sigma(z_i)$

Setelah diperoleh nilai linier z_i , tahap berikutnya adalah menghitung nilai $\sigma(z_i)$ dengan menerapkan fungsi aktivasi sigmoid pada nilai z_i menggunakan persamaan (2.2) :

1. Untuk $i = 1$

$$\begin{aligned}
 \sigma(z_1) &= \frac{1}{1 + \exp(-z_1)} \\
 &= \frac{1}{1 + \exp(-0,3)} \\
 &= 0,5744
 \end{aligned}$$

2. Untuk $i = 2$

$$\begin{aligned}
 \sigma(z_2) &= \frac{1}{1 + \exp(-z_1)} \\
 &= \frac{1}{1 + \exp(-0,4353)} \\
 &= 0,6071
 \end{aligned}$$

3. Untuk $i = 3$

$$\begin{aligned}
 \sigma(z_3) &= \frac{1}{1 + \exp(-z_3)} \\
 &= \frac{1}{1 + \exp(-0,5353)}
 \end{aligned}$$

$$= 0,6307$$

Dengan cara yang sama dilakukan untuk $i = 4, 5$, dan 6 . Hasil lengkap dari perhitungan nilai untuk seluruh data ditunjukkan pada Tabel 4.7 di bawah ini.

Tabel 4.7 Hasil Perhitungan Nilai σz_i

i	Nilai σz_i
$i = 1$	0,5744
$i = 2$	0,6071
$i = 3$	0,6307
$i = 4$	0,6071
$i = 5$	0,6307
$i = 6$	0,5000

3. Perhitungan *Loss*

Setelah diperoleh nilai probabilitas melalui fungsi aktivasi sigmoid, tahap berikutnya adalah menghitung fungsi *loss*. Pada sampel data ini, fungsi yang digunakan adalah *Binary Cross-Entropy Loss*, yang dinyatakan dengan persamaan:

$$\mathcal{L}_i = -[y_i \log(\sigma(z_i)) + (1-y_i)\log(1-\sigma(z_i))]$$

Langkah selanjutnya yaitu menghitung \mathcal{L}_i dengan $i = 1, 2, \dots, 6$ menggunakan persamaan diatas sebagai berikut:

a. Untuk $i = 1$

$$\begin{aligned}\mathcal{L}_1 &= -[y_1 \cdot \log(\sigma(z_1)) + (1-y_1)\log(1-\sigma(z_1))] \\ &= -[1 \cdot \log(0,5744) + (1-1)\log(1-(0,5744))] \\ &= 0,5543\end{aligned}$$

b. Untuk $i = 2$

$$\mathcal{L}_2 = -[y_2 \cdot \log(\sigma(z_2)) + (1-y_2)\log(1-\sigma(z_2))]$$

$$\begin{aligned}
&= -[1 \cdot \log(0,6071) + (1 - 1) \log(1 - (0,6071))] \\
&= 0,4990
\end{aligned}$$

c. Untuk $i = 3$

$$\begin{aligned}
\mathcal{L}_3 &= -[y_3 \cdot \log(\sigma(z_3)) + (1 - y_3) \log(1 - \sigma(z_3))] \\
&= -[0 \cdot \log(0,6307) + (1 - 0) \log(1 - (0,6307))] \\
&= 0,9962
\end{aligned}$$

Dengan cara yang sama dilakukan untuk $i = 4, 5$, dan 6 . Hasil lengkap dari perhitungan nilai \mathcal{L}_i untuk seluruh data ditunjukkan pada Tabel 4.8 di bawah ini.

Tabel 4.8 Hasil Perhitungan Nilai \mathcal{L}_i

i	Nilai \mathcal{L}_i
$i = 1$	0,5543
$i = 2$	0,4989
$i = 3$	0,9962
$i = 4$	0,9343
$i = 5$	0,9962
$i = 6$	0,6931

Setelah menghitung *loss* untuk masing masing sampel, langkah berikutnya yaitu menghitung nilai gradien untuk setiap parameter bobot yang akan dibahas pada tahap berikutnya.

4. Langkah Gauss-Newton

Setelah diperoleh nilai fungsi *loss*, langkah penting berikutnya adalah menghitung $\nabla \mathcal{L}_i$ untuk setiap parameter bobot. Gradien ini memberikan informasi mengenai arah perubahan fungsi *loss* terhadap bobot, sehingga dapat digunakan sebagai dasar dalam proses optimasi. Dengan kata lain, gradien

menunjukkan seberapa besar kontribusi setiap fitur dalam memengaruhi nilai *error* yang dihasilkan model, dengan persamaan :

$$\frac{\partial \mathcal{L}_i}{\partial \theta_j} = (1 - \sigma(z_1))x_{ij}$$

dengan $x_i^{aug} = [1, x_{1j}, x_{2j}, x_{3j}]$.

Nilai yang dipakai :

$$\sigma(z_1) = 0,5744, \sigma(z_2) = 0,6071, \sigma(z_3) = 0,6307,$$

$$\sigma(z_4) = 0,6071, \sigma(z_5) = 0,6307, \sigma(z_6) = 0,5000$$

a. Menghitung $\nabla \mathcal{L}_1$

$$\nabla \mathcal{L}_1 = \left[\frac{\partial \mathcal{L}_1}{\partial \theta_0}, \frac{\partial \mathcal{L}_1}{\partial \theta_1}, \frac{\partial \mathcal{L}_1}{\partial \theta_2}, \frac{\partial \mathcal{L}_1}{\partial \theta_3} \right]$$

1. Untuk $\frac{\partial \mathcal{L}_1}{\partial \theta_0}$

$$\begin{aligned} \frac{\partial \mathcal{L}_1}{\partial \theta_0} &= (1 - \sigma(z_1))x_{1,0} \\ &= (1 - 0,5744) \cdot 1 \\ &= 0,4256 \end{aligned}$$

2. Untuk $\frac{\partial \mathcal{L}_1}{\partial \theta_1}$

$$\begin{aligned} \frac{\partial \mathcal{L}_1}{\partial \theta_1} &= (1 - \sigma(z_1))x_{1,1} \\ &= (1 - 0,5744) \cdot 1 \\ &= 0,4256 \end{aligned}$$

3. Untuk $\frac{\partial \mathcal{L}_1}{\partial \theta_2}$

$$\begin{aligned} \frac{\partial \mathcal{L}_1}{\partial \theta_2} &= (1 - \sigma(z_1))x_{1,2} \\ &= (1 - 0,5744) \cdot 1 \end{aligned}$$

$$= 0,4256$$

4. Untuk $\frac{\partial \mathcal{L}_1}{\partial \theta_3}$

$$\frac{\partial \mathcal{L}_1}{\partial \theta_2} = (1 - \sigma(z_1))x_{1,2}$$

$$= (1 - 0,5744) \cdot 0$$

$$= 0$$

b. Menghitung $\nabla \mathcal{L}_2$

$$\nabla \mathcal{L}_2 = \left[\frac{\partial \mathcal{L}_2}{\partial \theta_0}, \frac{\partial \mathcal{L}_2}{\partial \theta_1}, \frac{\partial \mathcal{L}_2}{\partial \theta_2}, \frac{\partial \mathcal{L}_2}{\partial \theta_3} \right]$$

1. Untuk $\frac{\partial \mathcal{L}_2}{\partial \theta_0}$

$$\frac{\partial \mathcal{L}_2}{\partial \theta_0} = (1 - \sigma(z_2))x_{2,0}$$

$$= (1 - 0,6071) \cdot 1$$

$$= 0,3929$$

2. Untuk $\frac{\partial \mathcal{L}_2}{\partial \theta_1}$

$$\frac{\partial \mathcal{L}_2}{\partial \theta_1} = (1 - \sigma(z_2))x_{2,1}$$

$$= (1 - 0,6071) \cdot 0$$

$$= 0$$

3. Untuk $\frac{\partial \mathcal{L}_2}{\partial \theta_2}$

$$\frac{\partial \mathcal{L}_2}{\partial \theta_2} = (1 - \sigma(z_2))x_{2,2}$$

$$= (1 - 0,6071) \cdot 0,6765$$

$$= 0,2658$$

4. Untuk $\frac{\partial \mathcal{L}_2}{\partial \theta_3}$

$$\begin{aligned}
\frac{\partial \mathcal{L}_2}{\partial \theta_2} &= (1 - \sigma(z_2))x_{2,2} \\
&= (1 - 0,6071) \cdot 1 \\
&= 0,3929
\end{aligned}$$

Dengan cara yang sama dilakukan untuk $i = 3, 4, 5$, dan 6. Hasil lengkap dari perhitungan nilai gradien untuk seluruh data ditunjukkan pada matrix di bawah ini.

$$\begin{aligned}
\nabla \mathcal{L}_1 &\approx [0,4256, 0,4256, 0,4256, 0] \\
\nabla \mathcal{L}_2 &\approx [0,3929, 0, 0,2658, 0,3929] \\
\nabla \mathcal{L}_3 &\approx [-0,6307, -0,6307, -0,4267, -0,6307] \\
\nabla \mathcal{L}_4 &\approx [-0,6071, 0, -0,4107, -0,6071] \\
\nabla \mathcal{L}_5 &\approx [-0,6307, -0,6307, -0,4267, -0,6307] \\
\nabla \mathcal{L}_6 &= [0,5, 0, 0, 0]
\end{aligned}$$

Setelah diperoleh nilai gradien untuk setiap parameter bobot, tahap selanjutnya dalam proses optimasi model regresi logistik adalah membangun matrix Jacobian dan melakukan dekomposisi QR. Kedua matrix tersebut merupakan komponen penting dalam metode optimasi Gauss–Newton yang digunakan untuk memperbarui nilai bobot hingga mencapai solusi yang optimal. Matrix Jacobian merepresentasikan besarnya perubahan fungsi *error* terhadap masing-masing parameter model.

Dengan demikian, setiap elemen pada matrix Jacobian menunjukkan nilai turunan parsial dari fungsi *loss* terhadap bobot yang bersesuaian, sehingga menggambarkan tingkat sensitivitas atau pengaruh setiap fitur terhadap perubahan nilai *error* model. Pembentukan matrix Jacobian menjadi tahapan

esensial untuk memahami kontribusi setiap fitur terhadap hasil prediksi serta menentukan arah koreksi yang diperlukan dalam proses pembaruan bobot model. Kemudian ketika dilakukan perhitungan menggunakan data yang tersedia, diperoleh matrix jacobian seperti berikut:

$$J = \begin{bmatrix} 0,4256 & 0,4256 & 0,4256 & 0 \\ 0,3929 & 0 & 0,2658 & 0,3929 \\ -0,6307 & -0,6307 & -0,4266 & -0,6307 \\ -0,6071 & 0 & -0,4107 & -0,6071 \\ -0,6307 & -0,6307 & -0,4266 & -0,6307 \\ 0,5 & 0 & 0 & 0 \end{bmatrix}$$

Setelah matrix Jacobian J terbentuk, langkah selanjutnya adalah membentuk matrix sisi kanan (*Right-Hand Side*) yang dilambangkan dengan G . Matrix G ini diperoleh dari hasil perhitungan fungsi *loss* pada setiap observasi data ditunjukkan pada persamaan berikut:

$$G = \begin{bmatrix} 0,5543 \\ 0,4989 \\ 0,9962 \\ 0,9343 \\ 0,9962 \\ 0,6931 \end{bmatrix}$$

Setelah mendapatkan matrix jacobian dan matrix G , langkah selanjutnya yaitu membentuk sistem persamaan linear yang digunakan dalam proses pembaruan bobot dengan metode Gauss–Newton yang terdapat pada persamaan (2.25):

$$(J^T J) \Delta \theta = J^T G$$

Dengan hasil perhitungan J^T adalah sebagai berikut:

$$J^T = \begin{bmatrix} -0,4256 & -0,3929 & 0,6307 & 0,6071 & 0,6307 & -0,5 \\ -0,4256 & 0 & 0,6307 & 0 & 0,6307 & 0 \\ -0,4256 & -0,2658 & 0,4266 & 0,4107 & 0,4266 & 0 \\ 0 & -0,3929 & 0,6307 & 0,6071 & 0,6307 & 0 \end{bmatrix}$$

Lalu menghitung $J^T J$:

$$J^T J = \begin{bmatrix} 1,7496 & 0,9767 & 1,0730 & 1,0505 \\ 0,9767 & 0,7956 & 0,7192 & 0,7956 \\ 1,0730 & 0,7192 & 0,7844 & 0,7844 \\ 1,0505 & 0,7956 & 0,7844 & 0,8919 \end{bmatrix}$$

Setelah mendapatkan nilai $J^T J$, maka langkah selanjutnya yaitu menghitung nilai $J^T G$:

$$\begin{aligned} J^T G &= \begin{bmatrix} -0,4256 & -0,3929 & 0,6307 & 0,6071 & 0,6307 & -0,5 \\ -0,4256 & 0 & 0,6307 & 0 & 0,6307 & 0 \\ -0,4256 & -0,2658 & 0,4266 & 0,4107 & 0,4266 & 0 \\ 0 & -0,3929 & 0,6307 & 0,6071 & 0,6307 & 0 \end{bmatrix} \begin{bmatrix} 0,5543 \\ 0,4989 \\ 0,9962 \\ 0,9343 \\ 0,9962 \\ 0,6931 \end{bmatrix} \\ &= \begin{bmatrix} -0,4539 \\ -0,4432 \\ -0,3757 \\ -0,4246 \end{bmatrix} \end{aligned}$$

Selanjutnya untuk memperoleh vektor pembaruan $\Delta\theta$, dari perhitungan diatas didapatkan persamaan sebagai berikut:

$$\begin{bmatrix} 1,7496 & 0,9767 & 1,0730 & 1,0505 \\ 0,9767 & 0,7956 & 0,7192 & 0,7956 \\ 1,0730 & 0,7192 & 0,7844 & 0,7844 \\ 1,0505 & 0,7956 & 0,7844 & 0,8919 \end{bmatrix} \begin{bmatrix} \Delta\theta_0 \\ \Delta\theta_1 \\ \Delta\theta_2 \\ \Delta\theta_3 \end{bmatrix} = \begin{bmatrix} -0,4539 \\ -0,4432 \\ -0,3757 \\ -0,4246 \end{bmatrix}$$

5. Penyelesaian SPL dengan Dekomposisi QR

Pada tahap ini proses optimasi regresi logistik dilanjutkan dengan penyelesaian sistem persamaan linear menggunakan metode dekomposisi QR. Metode ini merupakan alternatif yang lebih stabil secara numerik dibandingkan penyelesaian langsung persamaan normal $(J^T J)\Delta\theta = J^T G$ menggunakan invers matrix. Dengan melakukan dekomposisi QR pada matrik J , persamaan normal

tersebut dapat ditulis sebagai sistem segitiga atas $R\Delta\theta = Q^T G$ yang kemudian diselesaikan dengan substitusi mundur.

Pada proses dekomposisi QR, matrix J akan diuraikan menjadi dua matrix Q dan R dengan Q merupakan matrix ortonormal dan R adalah matrix segitiga atas. Proses ini dilakukan dengan metode Gram-Schmidt pada Persamaan (2.22) dan (2.23), yang membentuk kolom-kolom ortogonal dari J dengan langkah sebagai berikut:

- a. Membentuk vektor pertama q_1

Ambil kolom pertama dari matrix J :

$$J_1 = \begin{bmatrix} 0,4256 \\ 0,3929 \\ -0,6307 \\ -0,6071 \\ -0,6307 \\ 0,5 \end{bmatrix}$$

Hitung norma r_{11} :

$$\begin{aligned} r_{11} = \|J_1\| &= \sqrt{(0,4256)^2 + (0,3929)^2 + (-0,6307)^2 + (-0,6071)^2 \\ &\quad + (-0,6307)^2 + (0,5)^2} \\ &= \sqrt{1,7497} = 1,3228 \end{aligned}$$

Selanjutnya normalisasi J_1 untuk mendapatkan q_1 :

$$q_1 = \frac{J_1}{r_{11}} = \begin{bmatrix} 0,4256/1,3228 \\ 0,3929/1,3228 \\ -0,6307/1,3228 \\ -0,6071/1,3228 \\ -0,6307/1,3228 \\ 0,5/1,3228 \end{bmatrix} = \begin{bmatrix} 0,3217 \\ 0,2970 \\ -0,4768 \\ -0,4590 \\ -0,4768 \\ 0,3780 \end{bmatrix}$$

- b. Membentuk vektor pertama q_2

Ambil kolom kedua dari J :

$$J_2 = \begin{bmatrix} 0,4256 \\ 0 \\ -0,6307 \\ 0 \\ -0,6307 \\ 0 \end{bmatrix}$$

Hitung proyeksi j_2 terhadap q_1 :

$$\begin{aligned} r_{12} &= q_1^T j_2 = (0,3217)(0,4256) + (0,2970)(0) \\ &\quad + (-0,4768)(-0,6307) + (-0,4590)(0) \\ &\quad + (-0,4768)(-0,6307) + (0,3780)(0) \\ &= 0,1369 + 0 + 0,3007 + 0 + 0,3007 + 0 \\ &= 0,7384 \end{aligned}$$

Kemudian hilangkan komponen yang searah dengan q_1 :

$$\begin{aligned} u_2 &= J_2 - r_{12} q_1 \\ u_2 &= \begin{bmatrix} 0,4256 \\ 0 \\ -0,6307 \\ 0 \\ -0,6307 \\ 0 \end{bmatrix} - 0,7384 \begin{bmatrix} 0,3217 \\ 0,2970 \\ -0,4768 \\ -0,4590 \\ -0,4768 \\ 0,3780 \end{bmatrix} \\ u_2 &= \begin{bmatrix} 0,4256 - 0,2376 \\ 0 - 0,2193 \\ -0,6307 + 0,3521 \\ 0 + 0,3389 \\ -0,6307 + 0,3521 \\ 0 - 0,2791 \end{bmatrix} = \begin{bmatrix} 0,1880 \\ -0,2193 \\ -0,2786 \\ 0,3389 \\ -0,2786 \\ -0,2791 \end{bmatrix} \end{aligned}$$

Hitung norma r_{22} :

$$\begin{aligned} r_{22} &= \|u_2\| \\ &= \sqrt{0,1880^2 + (-0,2193)^2 + (-0,2786)^2 + 0,3389^2 \\ &\quad + (-0,2786)^2 + (-0,2791)^2} \end{aligned}$$

$$\begin{aligned}
&= \sqrt{\frac{0,0353 + 0,0481 + 0,0776 + 0,1149 + 0,0776 + 0,0779}{2}} \\
&= \sqrt{0,4315} = 0,656
\end{aligned}$$

Normalisasi :

$$q_2 = \frac{u_2}{r_{22}} = \begin{bmatrix} 0,2862 \\ -0,3339 \\ -0,4242 \\ 0,5160 \\ -0,4242 \\ -0,4249 \end{bmatrix}$$

c. Membentuk vektor q_3

Ambil kolom ketiga dari J :

$$J_3 = \begin{bmatrix} 0,4256 \\ 0,2658 \\ -0,4266 \\ -0,4107 \\ -0,4266 \\ 0 \end{bmatrix}$$

Hitung proyeksi J_3 terhadap q_1 dan q_2 :

$$\begin{aligned}
1. \quad r_{13} &= q_1^T J_3 = (0,3217)(0,4256) + (0,2970)(0,2658) + \\
&\quad (-0,4768)(-0,4266) + (-0,4590)(-0,4107) + \\
&\quad (-0,4768)(-0,4266) + (0,3780)(0) \\
&= 0,1369 + 0,0789 + 0,2034 + 0,1885 + 0,2034 \\
&\quad + 0 \\
&= 0,8112
\end{aligned}$$

$$\begin{aligned}
2. \quad r_{23} &= q_2^T J_3 = (0,2862)(0,4256) + (-0,3339)(0,2658) + \\
&\quad (-0,4242)(-0,4266) + (0,5160)(-0,4107) + \\
&\quad (-0,4242)(-0,4266) + (-0,4249)(0)
\end{aligned}$$

$$\begin{aligned}
&= 0,1218 - 0,0887 + 0,1810 - 0,2119 + 0,1810 \\
&\quad + 0 \\
&= 0,1831
\end{aligned}$$

Selanjutnya, masing-masing proyeksi dihitung komponennya sebagai berikut:

$$u_3 = J_3 - r_{13}q_1 - r_{23}q_2$$

1. Untuk komponen pertama:

$$r_{13}q_1 = 0,8112 \times 0,3217 = 0,2610$$

$$r_{23}q_2 = 0,1831 \times 0,2862 = 0,0524$$

$$\text{Sehingga } u_{3,1} = 0,4256 - 0,2610 - 0,0524 = 0,1122$$

2. Untuk komponen kedua:

$$r_{13}q_1 = 0,8112 \times 0,2970 = 0,2409$$

$$r_{23}q_2 = 0,1831 \times (-0,3339) = -0,0611$$

$$\text{Sehingga } u_{3,2} = 0,2658 - 0,2409 + 0,0611 = 0,0860$$

3. Untuk komponen ketiga:

$$r_{13}q_1 = 0,8112 \times (-0,4768) = -0,3868$$

$$r_{23}q_2 = 0,1831 \times (-0,4242) = -0,0777$$

$$\text{Sehingga } u_{3,3} = -0,4266 + 0,3868 + 0,0777 = 0,0378$$

4. Untuk komponen keempat:

$$r_{13}q_1 = 0,8112 \times (-0,4590) = -0,3723$$

$$r_{23}q_2 = 0,1831 \times 0,5160 = 0,0945$$

$$\text{Sehingga } u_{3,4} = -0,4107 + 0,3727 - 0,0945 = -0,1328$$

5. Untuk komponen kelima:

$$r_{13}q_1 = 0,8112 \times (-0,4768) = -0,3868$$

$$r_{23}q_2 = 0,1831 \times (-0,4242) = -0,0777$$

$$\text{Sehingga } u_{3,5} = -0,4266 + 0,3868 + 0,0777 = 0,0378$$

6. Untuk komponen keenam:

$$r_{13}q_1 = 0,8112 \times 0,3780 = 0,3066$$

$$r_{23}q_2 = 0,1831 \times (-0,4249) = -0,0778$$

$$\text{Sehingga } u_{3,6} = 0 - 0,3066 + 0,0778 = -0,2288$$

Dengan demikian vektor u_3 yang dihasilkan adalah

$$u_3 = \begin{bmatrix} 0,1122 \\ 0,0860 \\ 0,0378 \\ -0,1328 \\ 0,0378 \\ -0,2288 \end{bmatrix}$$

Hitung norma:

$$\begin{aligned} r_{33} = \|u_3\| &= \sqrt{0,1122^2 + 0,0864^2 + 0,0378^2 + (-0,1328)^2 \\ &\quad + 0,0378^2 + (-0,2288)^2} \\ &= \sqrt{0,0126 + 0,0074 + 0,0014 + 0,0176 + 0,0014 + 0,0524} \\ &= \sqrt{0,0928} = 0,3047 \end{aligned}$$

Normalisasi:

$$q_3 = \frac{u_3}{r_{33}} = \begin{bmatrix} 0,3681 \\ 0,2821 \\ 0,1241 \\ -0,4359 \\ 0,1241 \\ -0,7510 \end{bmatrix}$$

d. Membentuk vektor pertama q_4

Ambil kolom keempat dari J :

$$J_4 = \begin{bmatrix} 0 \\ 0,3929 \\ -0,6307 \\ -0,6071 \\ -0,6307 \\ 0 \end{bmatrix}$$

Hitung proyeksi:

1. $r_{14} = q_1^T J_4 = (0,3217)(0) + (0,2970)(0,3929) +$
 $(-0,4768)(-0,6307) + (-0,4590)(-0,6071) +$
 $(-0,4768)(-0,6307) + (0,3780)(0)$
 $= 0 + 0,1167 + 0,3007 + 0,2787 + 0,3007 + 0$
 $= 0,9968$
2. $r_{24} = q_2^T J_4 = (0,2862)(0) + (-0,3339)(0,3929) +$
 $(-0,4242)(-0,6307) + (0,5160)(-0,6071) +$
 $(-0,4242)(-0,6307) + (-0,4249)(0)$
 $= 0 - 0,1312 + 0,2675 - 0,3133 + 0,2675 + 0$
 $= 0,0907$
3. $r_{34} = q_3^T J_4 = (0,3681)(0) + (0,2821)(0,3929) +$
 $(0,1241)(-0,6307) + (-0,4359)(-0,6071) +$
 $(0,1241)(-0,6307) + (-0,7510)(0)$
 $= 0 + 0,1108 - 0,0783 + 0,2647 - 0,0783 + 0$
 $= 0,2189$

Selanjutnya, masing-masing proyeksi dihitung komponennya, serta nilai

$u_4 = J_4 - (r_{14}q_1 + r_{24}q_2 + r_{34}q_3)$ untuk setiap elemen vektor sebagai berikut:

1. Untuk komponen pertama:

$$r_{14}q_1 = 0,9968 \times 0,3217 = 0,3207$$

$$r_{24}q_2 = 0,0907 \times 0,2862 = 0,0260$$

$$r_{34}q_3 = 0,2189 \times 0,3681 = 0,0806$$

$$\text{Sehingga } u_{4,1} = 0 - 0,3207 - 0,0260 - 0,0806 = -0,4272$$

2. Untuk komponen kedua:

$$r_{14}q_1 = 0,9968 \times 0,2970 = 0,2961$$

$$r_{24}q_2 = 0,0907 \times (-0,3339) = -0,0303$$

$$r_{34}q_3 = 0,2189 \times 0,2821 = 0,0617$$

$$\text{Sehingga } u_{4,2} = 0,3929 - 0,2961 + 0,0303 - 0,0617 = 0,0653$$

3. Untuk komponen ketiga:

$$r_{14}q_1 = 0,9968 \times (-0,4768) = -0,4753$$

$$r_{24}q_2 = 0,0907 \times (-0,4242) = -0,0385$$

$$r_{34}q_3 = 0,2189 \times 0,1241 = 0,0272$$

$$\text{Sehingga } u_{4,3} = -0,6307 + 0,4753 + 0,0385 - 0,0272 = -0,1441$$

4. Untuk komponen keempat:

$$r_{14}q_1 = 0,9968 \times (-0,4590) = -0,4575$$

$$r_{24}q_2 = 0,0907 \times 0,5160 = 0,0468$$

$$r_{34}q_3 = 0,2189 \times (-0,4359) = -0,0954$$

$$\text{Sehingga } u_{4,4} = -0,6071 + 0,4575 - 0,0468 + 0,0954 = -0,1010$$

5. Untuk komponen kelima:

$$r_{14}q_1 = 0,9968 \times (-0,4768) = -0,4753$$

$$r_{24}q_2 = 0,0907 \times (-0,4242) = -0,0385$$

$$r_{34}q_3 = 0,2189 \times 0,1241 = 0,0272$$

$$\text{Sehingga } u_{4,5} = -0,6307 + 0,4753 + 0,0385 - 0,0272 = -0,1441$$

6. Untuk komponen keenam:

$$r_{14}q_1 = 0,9968 \times 0,3780 = 0,3768$$

$$r_{24}q_2 = 0,0907 \times (-0,4249) = -0,0385$$

$$r_{34}q_3 = 0,2189 \times 0,7510 = -0,1644$$

$$\text{Sehingga } u_{4,6} = 0 - 0,3768 + 0,0385 + 0,1644 = -0,1739$$

Dengan demikian vektor hasilnya adalah:

$$u_4 = \begin{bmatrix} -0,4272 \\ 0,0653 \\ -0,1441 \\ -0,1010 \\ -0,1441 \\ -0,1739 \end{bmatrix}$$

Langkah berikutnya adalah menghitung normanya:

$$\begin{aligned} r_{44} &= \sqrt{(-0,4272)^2 + 0,0653^2 + (-0,1441)^2 + (-0,1010)^2 + (-0,1441)^2 + (-0,1739)^2} \\ &= \sqrt{0,1825 + 0,0043 + 0,0208 + 0,0102 + 0,0208 + 0,0302} \\ &= \sqrt{0,2866} = 0,5184 \end{aligned}$$

Normalisasi:

$$q_4 = \frac{u_4}{r_{44}} = \begin{bmatrix} -0,8241 \\ 0,1260 \\ -0,2780 \\ -0,1948 \\ -0,2780 \\ -0,3354 \end{bmatrix}$$

e. Pembentukan matrix Q dan R

Gabungkan semua vektor ortonormal:

$$Q = \begin{bmatrix} 0,3217 & 0,2862 & 0,3681 & -0,8241 \\ 0,2970 & -0,3339 & 0,2821 & 0,1260 \\ -0,4768 & -0,4242 & 0,1241 & -0,2780 \\ -0,4590 & -0,5160 & -0,4359 & -0,1948 \\ -0,4768 & -0,4242 & 0,1241 & -0,2780 \\ 0,3780 & -0,4249 & -0,7510 & -0,3354 \end{bmatrix}$$

Dan bentuk matrix R dari semua r_{ij} :

$$R = \begin{bmatrix} 1,3228 & 0,7384 & 0,8112 & 0,9968 \\ 0 & 0,6569 & 0,1831 & 0,0907 \\ 0 & 0 & 0,3047 & 0,2189 \\ 0 & 0 & 0 & 0,5184 \end{bmatrix}$$

Setelah matrix Q dan R diperoleh kemudian akan mensubstitusikan ke sistem *least-squares* sebagai berikut:

a. Hitung $b = Q^T G$

Ini merupakan proyeksi vektor *error* G ke ruang kolom matrix Q^{**}

1. Komponen pertama

$$\begin{aligned} b_1 &= 0,3217(0,5547) + 0,2970(0,4991) + (-0,4768)(0,9961) \\ &\quad + (-0,4590)(0,9356) + (-0,4768)(0,9961) \\ &\quad + (-0,4296)(0,6931) \\ &= 0,0776 + 0,0644 - 0,2060 - 0,1864 - 0,2060 + 0,1138 \\ &= -0,3426 \end{aligned}$$

2. Komponen kedua

$$\begin{aligned} b_2 &= 0,2862(0,5547) + (-0,3339)(0,4991) \\ &\quad + (-0,4242)(0,9961) + 0,5160(0,9356) \\ &\quad + (-0,4242)(0,9961) + (-0,4249)(0,6931) \\ &= 0,0689 - 0,0723 - 0,1835 + 0,2094 - 0,1835 - 0,1279 \\ &= -0,2889 \end{aligned}$$

3. Komponen ketiga

$$\begin{aligned}
 b_3 &= 0,3681(0,5547) + 0,2821(0,4991) + 0,1241(0,9961) \\
 &\quad + (-0,4359)(0,9356) + 0,1241(0,9961) \\
 &\quad + (-0,7510)(0,6931) \\
 &= 0,0887 + 0,0612 + 0,0537 - 0,1769 + 0,0537 - 0,2260 \\
 &= -0,1456
 \end{aligned}$$

4. Komponen keempat

$$\begin{aligned}
 b_4 &= (-0,8241)(0,5547) + 0,1260(0,4991) \\
 &\quad + (-0,2780)(0,9961) + (-0,1948)(0,4058) \\
 &\quad + (-0,2780)(0,4326) + (-0,3354)(0,6931) \\
 &= -0,1986 + 0,0273 - 0,1203 - 0,0790 - 0,1203 - 0,1010 \\
 &= -0,5919
 \end{aligned}$$

Maka hasilnya:

$$b = Q^T G = \begin{bmatrix} -0,3426 \\ -0,2889 \\ -0,1456 \\ -0,5919 \end{bmatrix}$$

b. Selesaikan $R\Delta\theta = s$

Hitung sistem segitiga atas pada persamaan (2.26) :

$$\begin{bmatrix} 1,3228 & 0,7384 & 0,8112 & 0,9968 \\ 0 & 0,6569 & 0,1831 & 0,0907 \\ 0 & 0 & 0,3047 & 0,2189 \\ 0 & 0 & 0 & 0,5184 \end{bmatrix} \begin{bmatrix} \Delta\theta_1 \\ \Delta\theta_2 \\ \Delta\theta_3 \\ \Delta\theta_4 \end{bmatrix} = \begin{bmatrix} -0,3426 \\ -0,2889 \\ -0,1456 \\ -0,5919 \end{bmatrix}$$

Hitung mulai:

1. Persamaan ke-4:

$$0,5184 \Delta\theta_4 = -0,5919 \Rightarrow \Delta\theta_4 = \frac{-0,5919}{0,5184} = -1,142$$

2. Persamaan ke-3

$$0,3047 \Delta\theta_3 + 0,2189(-1,142) = -0,1456$$

$$0,3047 \Delta\theta_3 + 0,2501 = -0,1456$$

$$\Rightarrow 0,3047\Delta\theta_3 = 0,1045$$

$$\Rightarrow \Delta\theta_3 = 0,3430$$

3. Persamaan ke-2

$$0,6569\Delta\theta_2 + 0,1831(0,3430) + 0,0907(-1,142) = -0,3426$$

$$0,6569\Delta\theta_2 + 0,0629 - 0,1037 = -0,2889$$

$$\Rightarrow 0,6569\Delta\theta_2 = -0,2481$$

$$\Rightarrow \Delta\theta_2 = -0,3780$$

4. Persamaan ke-1

$$1,3228 \Delta\theta_1 + 0,7384(-0,378) + 0,8112(0,343) + 0,9968(-1,142)$$

$$= -0,3426$$

$$1,3228 \Delta\theta_1 - 0,2788 + 0,2781 - 1,138 = -0,3426$$

$$\Rightarrow 1,3228 \Delta\theta_1 - 1,1387 = -0,3426$$

$$\Rightarrow 1,3228 \Delta\theta_1 = 0,7961$$

$$\Rightarrow \Delta\theta_1 = 0,6020$$

Lalu didapatkan hasil akhir sebagai berikut:

$$\Delta\theta = \begin{bmatrix} 0,6020 \\ -0,3780 \\ 0,3440 \\ -1,142 \end{bmatrix}$$

Nilai $\Delta\theta$ digunakan untuk memperbarui parameter regresi logistik dengan rumus $\theta^{k+1} = \theta^k + \Delta\theta$. Proses pembaruan dilakukan secara iteratif hingga memenuhi kriteria konvergensi, yaitu ketika norma perubahan parameter lebih kecil dari batas toleransi yang telah ditetapkan $||\Delta\theta|| < \varepsilon$. Jika kondisi ini

belum terpenuhi, maka iterasi dilanjutkan ke epoch berikutnya hingga mencapai konvergensi.

Setelah diperoleh parameter optimal hasil optimasi menggunakan metode Gauss–Newton dengan dekomposisi QR, tahap berikutnya adalah evaluasi performa model regresi logistik. Evaluasi dilakukan melalui kurva pembelajaran (*learning curve*) yang menggambarkan perubahan nilai *loss* dan akurasi terhadap jumlah epoch atau iterasi pembelajaran.

Kurva ini berfungsi untuk mengamati apakah model mengalami *overfitting*, *underfitting*, atau telah mencapai konvergensi yang stabil selama proses pelatihan. Perlu ditegaskan bahwa pada tahap implementasi penuh, proses evaluasi tidak terbatas pada enam data dengan dua fitur seperti contoh perhitungan manual di atas. Evaluasi selanjutnya dilakukan menggunakan seluruh data pada dataset Titanic, sehingga performa model dapat diukur secara menyeluruh terhadap berbagai variasi data.

Dengan demikian, hasil evaluasi mampu merepresentasikan kemampuan generalisasi model regresi logistik yang dioptimasi menggunakan metode Gauss–Newton dengan dekomposisi QR, serta memastikan model mampu menyelesaikan permasalahan klasifikasi biner secara akurat, stabil, dan efisien secara numerik.

6. Pembaruan Parameter dan Ilustrasi Iterasi Awal

Pada pembaruan parameter akan memperlihatkan satu langkah Gauss–Newton dengan penyelesaian Dekomposisi QR memperbarui parameter regresi logistik pada contoh mini tabel 4.4 dengan parameter awal

$$\theta^{(0)} = \begin{bmatrix} 0 \\ 0,1 \\ 0,2 \\ 0,3 \end{bmatrix}$$

Setiap baris i memiliki vektor fitur terstandar $x_i = [1, x_{i1}, x_{i2}, x_{i3}]^T$ dengan elemen pertama = 1 adalah bias. Dengan data contoh 6 baris ini, proses tersebut menghasilkan vektor pembaruan:

$$\Delta\theta = \begin{bmatrix} 0,6020 \\ -0,3780 \\ 0,3440 \\ -1,142 \end{bmatrix}$$

Pembaruan parameter (iterasi 1):

$$\theta^1 = \theta^0 + \Delta\theta = \begin{bmatrix} 0 + 0,6020 \\ 0,1 + (-0,3780) \\ 0,2 + 0,3440 \\ 0,3 + (-1,142) \end{bmatrix} = \begin{bmatrix} 0,6020 \\ -0,2780 \\ 0,5430 \\ -0,8420 \end{bmatrix}$$

Vektor parameter baru di atas merupakan hasil satu langkah Gauss–Newton dengan penyelesaian QR. Setelah pembaruan ini, yang dilakukan adalah mengevaluasi ulang model pada keenam data contoh: hitung kembali skor model, ubah menjadi probabilitas, lalu lihat selisihnya terhadap label sebenarnya untuk memastikan galat keseluruhan mengecil. Selanjutnya, nilai kerugian (*loss*) rata-rata dihitung ulang dan dibandingkan dengan sebelum pembaruan untuk memeriksa apakah terjadi perbaikan.

Kemudian, cek kriteria berhenti yang telah ditetapkan misalnya, perubahan parameter sudah sangat kecil atau penurunan *loss* sudah tidak berarti. Jika kriteria terpenuhi, iterasi pada contoh mini ini dianggap konvergen. Jika belum, proses diulangi: bentuk kembali turunan/kepekaan model (Jacobian), lakukan dekomposisi QR, peroleh langkah perbaikan berikutnya, dan perbarui parameter lagi.

Akhir dari bagian ini menutup alur pembaruan, evaluasi ulang, cek konvergensi pada contoh kecil. Prosedur yang sama akan dipakai pada eksperimen seluruh data di Subbab 4.2.2 namun dengan *early stopping*, serta pelaporan kurva konvergensi, waktu pelatihan, dan metrik klasifikasi pada *validation* dan *test* agar gambaran kinerja model menjadi lengkap dan representatif.

4.2.2 Eksperimen pada Seluruh Data

Setelah sebelumnya disajikan perhitungan manual pada enam data awal guna mengilustrasikan langkah-langkah kombinasi Metode Gauss–Newton dan dekomposisi QR, tahap selanjutnya adalah penanganan komputasional pada keseluruhan data. Pada tahap ini, seluruh observasi dan fitur yang tersedia diproses dalam satu alur eksperimen terstandar untuk memperoleh parameter model final, kemudian dievaluasi menggunakan metrik performa (akurasi, presisi, *recall*, *F1-Score*, serta waktu pelatihan. Hasilnya disajikan dalam bentuk ringkasan numerik dan visual kurva konvergensi/*loss*, *confusion matrix*, guna menegaskan konsistensi antara ilustrasi manual skala kecil dan capaian model pada skala penuh.

1. Pengaturan Eksperimen: Pra-Proses, *Encoding*

Setelah dilakukan pembagian data, tahapan Pra-pemrosesan merupakan langkah penting sebelum dilakukan analisis dan pemodelan regresi logistik, sehingga data siap diolah lebih untuk diterapkan ke dalam model pembelajaran mesin. Dataset Titanic yang digunakan memiliki 7 fitur utama, baik kategorik maupun numerik, namun tidak seluruhnya dalam kondisi siap pakai. Oleh

karena itu, dilakukan beberapa proses pembersihan dan transformasi data agar lebih akurat, konsisten, serta sesuai dengan kebutuhan analisis.

a. Penanganan Nilai Hilang

Tabel 4.8 berikut menyajikan kondisi kelengkapan data pada masing-masing subset.

Tabel 4.9 Hasil Pengecekan *Missing Value*

Fitur	<i>Train</i>	<i>Validation</i>	<i>Test</i>
<i>age</i>	190	31	42
<i>fare</i>	1	0	0
<i>pclass</i>	0	0	0
<i>sex</i>	0	0	0
<i>sibsp</i>	0	0	0
<i>parch</i>	0	0	0
<i>embarked</i>	0	1	1

Kekosongan pada fitur numerik *age* dan *fare* diisi dengan median yang dihitung hanya dari data latih karena median tahan terhadap pencilan (*outliers*) dan tidak mudah “tertarik” oleh beberapa nilai ekstrem. Ini relevan untuk *fare* yang distribusinya sering miring ke kanan (banyak nilai kecil, sedikit nilai sangat besar, ekor panjang di kanan grafik) dan *age* yang kadang memiliki nilai ekstrem, pemakaian median menjaga pusat sebaran tanpa mendistorsi struktur data, membuat pelatihan lebih stabil dan mengurangi risiko bias.

Untuk fitur kategorik *embarked* dipilih modus dari data latih agar distribusi kategori dominan tetap terpelihara, tidak perlu memperkenalkan label fiktif, dan mudah direplikasi. Menghitung median/modus hanya dari train lalu menerapkan ke *validation* dan *test* mencegah data *leakage*

sehingga evaluasi sah. Setelah imputasi, nilai kategorik dinormalisasi lalu di-*encode*.

b. Pengkodean Variabel Kategorik

Proses pengkodean variabel kategorik merupakan tahap penting dalam prapemrosesan data sebelum model pembelajaran mesin dilatih. Sebagian besar algoritma hanya dapat menerima masukan berupa data numerik, sehingga variabel bertipe kategorik harus diubah terlebih dahulu ke bentuk numerik tanpa mengubah makna kategorinya. Dalam penelitian ini, proses pengkodean dilakukan setelah tahap pembagian data (*data splitting*) dan penanganan nilai hilang (*missing value imputation*). Urutan ini diterapkan untuk menghindari terjadinya *data leakage*, yaitu kondisi ketika informasi dari data validasi atau data uji secara tidak langsung memengaruhi proses pelatihan model. Langkah ini sesuai dengan praktik praproses yang direkomendasikan oleh Zhu, dkk., 2024., yang menegaskan bahwa seluruh transformasi berbasis data harus dipelajari hanya dari data latih dan diterapkan secara konsisten pada subset lainnya.

Variabel kategorik yang digunakan dalam penelitian ini terdiri atas “*sex*” dan “*embarked*”. Sebelum dilakukan proses pengkodean, kedua variabel tersebut terlebih dahulu melalui tahap imputasi nilai hilang menggunakan *SimpleImputer* dengan strategi *most_frequent*, untuk menggantikan nilai kosong dengan kategori yang paling sering muncul. Selanjutnya, dilakukan proses normalisasi teks menggunakan perintah Python untuk memastikan bahwa seluruh nilai kategori memiliki format

yang seragam dan bebas dari kesalahan ejaan atau spasi tambahan. Proses normalisasi ini penting agar kategori yang seharusnya sama tidak dibaca berbeda oleh sistem, sebagaimana disarankan oleh Cerda dkk., 2018 dalam penelitiannya mengenai *similarity encoding* untuk data kategorik “kotor”.

Untuk variabel “sex”, pengkodean dilakukan menggunakan *binary encoding*, di mana kategori *male* dikonversi menjadi nilai 0 dan *female* menjadi 1, sebagaimana ditunjukkan pada Tabel 4.10.

Tabel 4.10 Hasil Normalisasi Fitur Sex

<i>Sex</i>	Kode numerik
<i>Male</i>	0
<i>female</i>	1

Pemilihan metode ini didasarkan pada karakteristik variabel *sex* yang hanya memiliki dua kategori (*low cardinality*). Pada kasus seperti ini, penggunaan *binary encoding* jauh lebih efisien dibandingkan dengan *one-hot encoding*, karena tidak menambah jumlah kolom baru dan tetap mampu merepresentasikan perbedaan antar kategori secara jelas.

Menurut Bolikulov dkk., 2024, label atau *binary encoding* merupakan pilihan yang paling efisien untuk variabel kategorik sederhana karena tidak menimbulkan curse of dimensionality seperti halnya one-hot encoding. Selain itu, Dahouda & Joe, 2021 juga menekankan bahwa metode pengkodean kompleks seperti *deep embedding* hanya diperlukan untuk variabel kategorik berdimensi tinggi, sedangkan variabel biner seperti *sex* sudah cukup diwakili oleh pengkodean dua nilai. Poslavskaya & Korolev, 2023 turut memperkuat hal ini dengan menyatakan bahwa pada variabel biner, *binary encoding* memberikan kinerja yang setara

dengan *one-hot encoding* namun dengan kebutuhan memori dan waktu komputasi yang lebih kecil.

Sementara itu, variabel “*embarked*” memiliki tiga kategori, yaitu C, Q, dan S, yang merepresentasikan pelabuhan keberangkatan penumpang. Untuk variabel ini digunakan metode *one-hot encoding* dari pustaka *pandas*. Proses ini menghasilkan tiga kolom baru yang masing-masing mewakili ketiga kategori tersebut. Dengan demikian, data yang semula berbentuk teks atau simbol diubah menjadi bentuk numerik yang dapat dipahami oleh algoritma pembelajaran mesin, tanpa memberikan makna urutan atau hubungan antar kategori.

Fungsi ini juga memiliki kemampuan untuk menghindari duplikasi informasi antar kolom dengan menghapus satu kategori dasar (melalui parameter pengaturan tertentu), sehingga mencegah terjadinya multikolinearitas dalam model linier. Dalam konteks penelitian ini, hasil transformasi dari fungsi tersebut membuat setiap kategori *embarked* memiliki representasi yang terpisah dan eksplisit, sehingga model dapat menginterpretasikan masing-masing kategori secara independen, sebagaimana ditunjukkan pada Tabel 4.11.

Tabel 4.11 Hasil Normalisasi Fitur Embarked

Kategori Embarked	Emb C	Emb Q
C	1	0
Q	0	1
S	0	0

Konsep ini sejalan dengan prinsip yang dijelaskan oleh Zhu, dkk., 2024, yang menyatakan bahwa one-hot encoding merupakan teknik paling stabil dan efektif untuk variabel kategorik dengan lebih dari dua nilai unik. Metode ini memastikan bahwa algoritma pembelajaran mesin tidak salah menginterpretasikan nilai kategori sebagai besaran numerik yang memiliki urutan tertentu. Hal ini juga sejalan dengan temuan Poslavskaya & Korolev, 2023 yang menyebutkan bahwa yang menemukan *bahwa one-hot encoding* merupakan metode paling stabil secara lintas domain untuk data kategorik dengan beberapa kategori.

Secara teoretis, kedua variabel tersebut sebenarnya dapat dikodekan dengan *one-hot encoding*, karena metode ini bersifat umum untuk semua variabel kategorik. Namun, penerapan *one-hot* pada variabel biner seperti *sex* tidak efisien, sebab akan menghasilkan dua kolom dummy yang saling berlawanan (*redundant*). Misalnya, jika *male* dikonversi menjadi kolom sex_{male} dan *female* menjadi sex_{female} , maka informasi yang terkandung pada salah satu kolom sebenarnya sudah tercermin dari kolom lainnya. Oleh karena itu, dalam praktik pembelajaran mesin yang efisien, variabel biner seperti *sex* lebih baik direpresentasikan dengan satu kolom numerik menggunakan *binary encoding*. Sebaliknya, variabel *embarked* yang memiliki tiga nilai unik lebih tepat dikodekan dengan *one-hot encoding* karena dapat membedakan setiap kategori tanpa mengasumsikan hubungan ordinal di antara nilai-nilainya.

Seluruh proses pengkodean dilakukan setelah data dibagi menjadi subset pelatihan, validasi, dan pengujian. Pengkodean dilakukan dengan

fit hanya pada data latih dan hasil transformasinya diterapkan secara konsisten ke data validasi dan uji. Pendekatan ini memastikan bahwa informasi dari subset validasi dan uji tidak bocor ke dalam proses pelatihan, sesuai dengan prinsip pencegahan data *leakage* yang dijelaskan oleh Zhu, dkk., 2024. Dengan penerapan ini, variabel kategorik dalam dataset dapat direpresentasikan secara numerik dengan efisien, terhindar dari bias informasi, serta tetap mempertahankan makna semantik antar kategori.

Berdasarkan landasan teoretis dan pertimbangan implementatif tersebut, penelitian ini menetapkan bahwa variabel “*sex*” dikodekan menggunakan *Binary Encoding*, sedangkan variabel “*embarked*” dikodekan menggunakan *One-Hot Encoding*. Kedua metode ini dianggap paling sesuai dengan karakteristik datanya masing-masing, efisien secara komputasi, serta sejalan dengan pedoman ilmiah dalam pengolahan data tabular modern.

2. Penanganan Masalah Numerik dengan *Clipping*

Pada implementasi ini digunakan klip ± 500 pada input sigmoid untuk menjaga stabilitas numerik, yakni mencegah *overflow/underflow* pada perhitungan eksponensial sehingga probabilitas dan *log-loss* tidak menghasilkan nilai tak hingga atau tidak terdefinisi. Dampaknya, proses optimasi menjadi lebih stabil dan konsisten, gradien tidak pecah, struktur Jacobian tetap terjaga, dan penyelesaian sistem linier melalui dekomposisi QR

lebih jarang gagal, hasilnya konvergensi berjalan mulus dengan waktu komputasi yang lebih dapat diprediksi.

Adapun komprominya, nilai logit yang sangat ekstrem akan dipangkas sehingga informasi gradien pada area yang sangat jenuh sedikit diredam. Untuk memastikan manfaatnya tetap optimal, fitur tetap distandardisasi dan frekuensi terjadinya klip dipantau. Bila terlalu sering, perbaiki skala fitur atau tambahkan regularisasi alih-alih memperlonggar batas. Untuk menangani masalah tersebut, diterapkan fungsi *clipping* dengan memanfaatkan `np.clip()` untuk membatasi nilai input sigmoid ke rentang yang aman (misalnya ± 500). Sebelum masuk ke sigmoid, nilai logit z “dipagari” ke rentang aman.

- a. Jika $z > 500$, maka nilai z menjadi 500,
- b. Jika $z < -500$, maka nilai z menjadi -500,
- c. Jika $-500 \leq z \leq 500$, maka nilai z dibiarkan apa adanya.

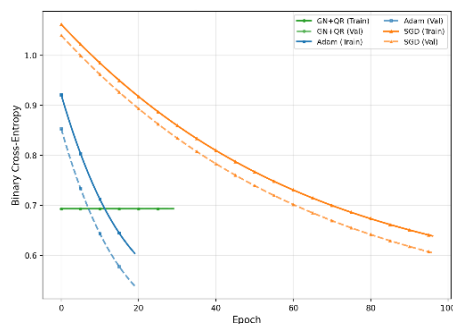
Dengan pembatasan tersebut, nilai eksponensial yang digunakan dalam perhitungan sigmoid tetap berada pada rentang yang aman untuk komputasi sehingga tidak jatuh ke nol persis maupun meledak menjadi tak hingga. Akibatnya, probabilitas keluaran dan *log-loss* selalu terdefinisi dengan baik, bebas dari nilai tidak sah seperti \inf atau NaN, dan proses optimasi berjalan stabil gradien tidak pecah, struktur Jacobian tetap terjaga, dan penyelesaian sistem linier melalui dekomposisi QR lebih andal.

3. Konvergensi dan Waktu Pelatihan

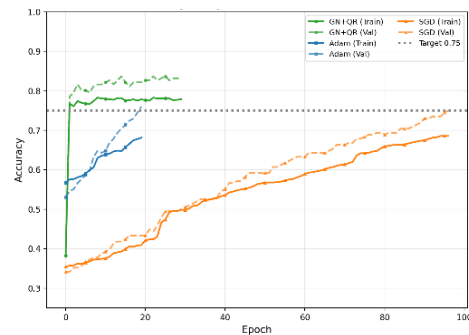
Seluruh eksperimen menggunakan praproses identik (standardisasi fitur), *clipping* logit ± 500 untuk menjaga stabilitas numerik pada fungsi sigmoid, dan

kriteria berhenti yang sama. Evaluasi validasi dilakukan pada akhir setiap epoch (1 epoch = 29 *mini-batch* berukuran 32), dengan *early stopping* ketika Val Akurasi $\geq 0,75$ (dengan *patience* berbasis epoch). Tiga skema optimasi yang dibandingkan adalah Gauss–Newton dengan penyelesaian sistem linier via dekomposisi QR, Stochastic Gradient Descent (SGD), dan Adam.

Agar pembaca dapat langsung melihat dampak dari pengaturan eksperimen tersebut terhadap proses optimasi, pada bagian berikut disajikan kurva pembelajaran yang menggambarkan dinamika pelatihan pada data train dan validation. Penyajian ini dimaksudkan untuk menilai kecepatan penurunan *loss*, kestabilan konvergensi, serta kesenjangan *train-validation* sehingga potensi *overfitting/underfitting* dapat diidentifikasi sebelum berlanjut ke ringkasan numerik pada tabel dan evaluasi akhir di subbab berikutnya.



Gambar 4.2 Kurva Pembelajaran *Loss* untuk Tiga Optimasi



Gambar 4.3 Kurva Pembelajaran Akurasi untuk Tiga Optimasi

Pada Gambar 4.2, Pada Adam (biru) memang turun paling cepat di awal lalu melandai, tanda konvergensi cepat dan stabil. SGD (oranye) menurun lebih lambat dan baru mendekati nilai rendah di akhir epoch. Sementara itu, Gauss–Newton (hijau) terlihat hampir datar bukan karena macet, melainkan karena sudah sangat cepat mencapai daerah optimum, yaitu penurunan *loss*

numeriknya kecil setelah beberapa langkah. Jadi, perilaku *loss* yang datar pada Gauss-Newton mencerminkan konvergensi sangat cepat dengan generalisasi yang baik, sehingga penambahan epoch memberi manfaat yang semakin kecil (*diminishing returns*).

Pada Gambar 4.3 Gauss-Newton mencapai akurasi tertinggi dan stabil dalam beberapa epoch saja dengan gap *train-validation* kecil (generalisasi baik), Adam naik cepat dan mencapai target akurasi lebih awal daripada SGD lalu stabil, sementara SGD meningkat paling lambat dan baru mendekati target menjelang akhir pelatihan. Secara keseluruhan, kecepatan dan stabilitas performa terbaik terlihat pada Gauss-Newton, diikuti Adam, kemudian SGD. Sebagai penguat temuan visual tersebut, Tabel 4.12 merangkum *checkpoint* terbaik masing-masing metode yakni epoch saat validasi mencapai performa tertinggi sesuai kriteria *early stopping*.

Tabel 4.12 Hasil *Loss* dan *Akurasi* untuk Data Train dan Val untuk 3 Optimasi

<i>Optimizer</i>	<i>Epoch</i>	<i>Train Akurasi</i>	<i>Train Loss</i>	<i>Val Akurasi</i>	<i>Val Loss</i>	Waktu Total (s)
GN	1	0,7675	0,6931	0,7857	0,6931	0,02
SGD	97	0,6856	0,6388	0,7500	0,6051	12,09
Adam	20	0,6812	0,6041	0,7602	0,5389	5,42

Pada tabel 4.12 membandingkan metode Gauss-Newton, Adam, dan SGD menggunakan metrik *Train/Val Loss* serta *Train/Val Akurasi*. Semua optimasi dilakukan evaluasi validasi di akhir setiap epoch (1 epoch = 29 batch dengan *mini-batch* 32), dan pelatihan dibatasi oleh *early stopping* saat *Val Akurasi* $\geq 0,75$ dengan *patience* berbasis epoch. Sebagai pembanding efisiensi varian

Gauss-Newton yang mengevaluasi validasi per-batch dan memungkinkan *early stop* di level batch.

Pada *checkpoint* terbaiknya, Gauss-Newton unggul sekaligus dari sisi akurasi dan efisiensi. Hasilnya, hanya dalam epoch 1 sudah mencapai $Val\ Acc = 0,7857$ dengan $Train\ Akurasi = 0,7675$ dan $Train/Val\ Loss = 0,6931$, Secara kuantitatif, Gauss-Newton unggul akurasi 0,0255 dibanding Adam dan 0,0357 dibanding SGD, namun dengan biaya komputasi yang jauh lebih rendah dan jauh lebih sedikit update parameter. Fakta bahwa $Val\ Akurasi > Train\ Akurasi$ merupakan sinyal generalisasi yang baik pada split validasi.

Sebagai pembandingan, Adam mencapai *checkpoint* terbaik pada epoch 20 dengan $Val\ Akurasi = 0,7602$, $Val\ Loss = 0,5389$, $Train\ Akurasi = 0,6812$, dan $Train\ Loss = 0,6388$. Profil ini menandakan kalibrasi probabilitas yang lebih baik meskipun akurasi validasinya masih di bawah Gauss-Newton ($-0,0255$). Dalam konteks penelitian yang mengutamakan akurasi dan kecepatan konvergensi, Gauss-Newton mempertahankan akurasi tertinggi

Pada epoch 97, SGD mencapai $Val\ Akurasi = 0,7500$, $Val\ Loss = 0,6051$, $Train\ Akurasi = 0,6856$, dan $Train\ Loss = 0,6041$, Dinamikanya stabil namun paling lambat, perlu banyak epoch untuk menembus ambang akurasi, dan pada titik terbaiknya masih di bawah Gauss-Newton dari sisi akurasi ($-0,0357$) serta di bawah Adam dari sisi *loss*. Metode SGD menunjukkan bahwa tanpa penyetelan agresif, konvergensinya kurang efisien. Ini makin menekankan nilai praktis Gauss-Newton yang mencapai akurasi tertinggi dengan cepat, sementara SGD berfungsi sebagai *baseline* yang dapat ditingkatkan melalui tuning bila diperlukan.

Gauss-Newton adalah metode paling efisien, ia mencapai ambang akurasi hanya dalam 1 epoch dengan waktu total $\sim 0,02$ s menandakan konvergensi yang sangat cepat tanpa ada kegagalan iterasi. Adam berada di posisi tengah, konvergen pada epoch 20 dengan waktu $\sim 5,42$, yang mencerminkan laju belajar yang cepat dan stabil. Sementara itu, SGD menjadi yang paling lambat, baru melampaui ambang pada epoch 97 dengan waktu $\sim 12,09$ s, meski tetap tanpa kegagalan iterasi. Secara praktis, hasil ini menegaskan keunggulan Gauss-Newton dalam kecepatan konvergensi (hemat waktu/komputasi) untuk mencapai target akurasi dengan sangat cepat.

4. Kinerja Klasifikasi (Akurasi, Presisi, *Recall*, *F1-Score*)

Evaluasi kinerja pada data uji dilakukan untuk menilai kemampuan generalisasi model di luar data pelatihan. Penilaian difokuskan pada *confusion matrix* dan akurasi, presisi, *recall*, dan *F1-score*. Karena dataset Titanic bersifat tidak seimbang, presisi dan recall dilaporkan bersama akurasi agar pembacaan kinerja lebih adil. Pada evaluasi ini, kelas positif didefinisikan sebagai “selamat”.

Berikut merupakan hasil confusion Matrix yang diperoleh dari model klasifikasi dengan data testing menggunakan algoritma Gauss-Newton. Ditunjukkan pada tabel berikut:

Tabel 4.13 Hasil *Confusion Matrix* untuk Data Uji

Aktual	Prediksi	
	0 (Tidak selamat)	1 (Selamat)
0 (Tidak selamat)	112	10
1 (Selamat)	23	52

Confusion matrix pada Tabel 4.13 memberikan gambaran komprehensif mengenai kemampuan model regresi logistik dengan metode Gauss–Newton dalam membedakan dua kelas, yaitu kelas negatif (0: tidak selamat) dan kelas positif (1: selamat). Kelas negatif menggambarkan kondisi penumpang yang tidak selamat, sedangkan kelas positif menggambarkan penumpang yang selamat. Pemahaman terhadap klasifikasi kedua kelas ini memiliki peran penting karena setiap bentuk kesalahan prediksi membawa implikasi berbeda ketika dianalisis pada konteks keselamatan penumpang kapal.

Pada kelas negatif, model menghasilkan 112 prediksi benar (*True Negative* / *TN*). Nilai ini menunjukkan bahwa model memiliki kemampuan yang baik dalam mengenali penumpang yang secara aktual tidak selamat. Keberhasilan model dalam mengidentifikasi kelas negatif memberikan indikasi bahwa pola-pola karakteristik yang berhubungan dengan risiko tidak selamat dapat dipelajari dengan cukup baik oleh model. Meskipun demikian, terdapat 10 kasus *False Positive* (*FP*), yaitu kondisi ketika model memprediksi penumpang sebagai “selamat” padahal pada kenyataannya penumpang tersebut tidak selamat. Jika dianalisis dalam konteks aplikatif, kesalahan jenis *FP* dapat memberikan dampak yang cukup signifikan. Misalnya, apabila model digunakan dalam sistem pendukung keputusan berbasis prediksi probabilitas keselamatan, kesalahan ini dapat menyebabkan situasi di mana individu yang seharusnya teridentifikasi sebagai berisiko tinggi justru tidak mendapatkan perhatian khusus. Dengan kata lain, *FP* dapat menciptakan persepsi keliru mengenai tingkat keselamatan seseorang, yang berpotensi memengaruhi evaluasi kebijakan keselamatan atau interpretasi hasil analisis risiko.

Sementara itu, pada kelas positif, model menghasilkan 52 prediksi benar (*True Positive / TP*). Hal ini menunjukkan bahwa model mampu mengenali sebagian besar penumpang yang secara aktual selamat. Namun demikian, terdapat 23 kasus False Negative (*FN*), yaitu kondisi ketika model memprediksi penumpang sebagai “tidak selamat” padahal penumpang tersebut sebenarnya selamat. Dalam konteks penilaian performa model, keberadaan *FN* memiliki implikasi analitis yang penting. *FN* menunjukkan bahwa model masih memiliki keterbatasan dalam menangkap seluruh pola yang berkaitan dengan peluang keselamatan penumpang. Jika model digunakan untuk tujuan analisis faktor keselamatan atau pembuatan simulasi skenario, keberadaan *FN* dapat menyebabkan gambaran yang terlalu pesimis terhadap tingkat keselamatan sehingga mengurangi ketepatan interpretasi mengenai kelompok penumpang yang berpotensi selamat. *Confusion matrix* ini menjadi dasar untuk menghitung berbagai metrik performa seperti akurasi, presisi, *recall*, dan *F1-score*.

a. Nilai Akurasi

$$\begin{aligned}
 \text{Accuracy} &= \frac{TP + TN}{TP + TN + FP + FN} \\
 &= \frac{52 + 112}{52 + 112 + 10 + 23} \\
 &= \frac{164}{197} \\
 &= 0,8325 \approx 83.25\%
 \end{aligned}$$

b. Nilai Presisi (kelas positif)

$$\text{Precision} = \frac{TP_{(1)}}{TP_{(1)} + FP_{(1)}}$$

$$\begin{aligned}
 &= \frac{52}{52 + 10} \\
 &= \frac{52}{62} \\
 &= 0,8387 \approx 83.87\%
 \end{aligned}$$

c. Nilai Presisi (kelas negatif)

$$\begin{aligned}
 Precision &= \frac{TP_{(0)}}{TP_{(0)} + FP_{(0)}} \\
 &= \frac{112}{112 + 23} \\
 &= \frac{112}{135} \\
 &= 0,8296 \approx 82,96\%
 \end{aligned}$$

d. Nilai *Recall* (kelas positif)

$$\begin{aligned}
 recall &= \frac{TP_{(1)}}{TP_{(1)} + FN_{(1)}} \\
 &= \frac{52}{52 + 23} \\
 &= \frac{52}{75} \\
 &= 0,6933 \approx 69,33\%
 \end{aligned}$$

e. Nilai *Recall* (kelas negatif)

$$\begin{aligned}
 recall &= \frac{TP_{(0)}}{TP_{(0)} + FN_{(0)}} \\
 &= \frac{112}{112 + 10} \\
 &= \frac{112}{122} \\
 &= 0,9180 \approx 91,80\%
 \end{aligned}$$

f. Nilai *F1-Score* (kelas positif)

$$\begin{aligned}
 F1Score &= 2 \times \frac{precision_{(1)} \times recall_{(1)}}{precision_{(1)} + recall_{(1)}} \\
 &= 2 \times \frac{0,8387 \times 0,6933}{0,8387 + 0,6933} \\
 &= 2 \times \frac{0,5813}{1,532} \\
 &= 0,7591 \approx 75.91\%
 \end{aligned}$$

g. Nilai *F1-Score* (kelas negatif)

$$\begin{aligned}
 F1Score &= 2 \times \frac{precision_{(0)} \times recall_{(0)}}{precision_{(0)} + recall_{(0)}} \\
 &= 2 \times \frac{0,8296 \times 0,9180}{0,8296 + 0,9180} \\
 &= 2 \times \frac{0,7617}{1,7476} \\
 &= 0,8720 \approx 87,20\%
 \end{aligned}$$

Dari perhitungan diatas didapatkan hasil keseluruhan pada Tabel 4.14

Tabel 4.14 Hasil Perhitungan Confusion Matriks

Metrik	Kelas selamat	Kelas tidak selamat
Presisi	83,87%	82,96%
<i>Recall</i>	69,33%	91,80%
<i>F1 – Score</i>	75,91%	87,20%

Berdasarkan hasil perhitungan metrik evaluasi yang diperoleh dari *confusion matrix*, model regresi logistik yang dioptimasi menggunakan metode Gauss–Newton dan dekomposisi QR menunjukkan kinerja yang cukup baik pada data uji. Model menghasilkan akurasi sebesar 83,25%, dengan presisi kelas 83,87%, *recall* 69,33%, dan *F1-score* 75,91% untuk kelas positif (kelas 1: selamat). Pada data ini, terdapat ketidakseimbangan distribusi kelas, di mana

jumlah penumpang tidak selamat (kelas 0) berjumlah 122 observasi, sedangkan penumpang selamat (kelas 1) hanya 75 observasi. Ketimpangan jumlah ini membuat evaluasi menggunakan presisi, *recall*, dan *F1-score* menjadi penting karena akurasi saja tidak mampu menggambarkan performa model secara menyeluruh.

Akurasi sebesar 83,25% memberikan gambaran umum bahwa sebagian besar prediksi model sesuai dengan kondisi aktual. Namun, pada konteks dataset Titanic yang tidak seimbang, akurasi dapat menyesatkan apabila berdiri sendiri. Jumlah korban jiwa yang lebih tinggi dibanding jumlah penumpang yang selamat menyebabkan model lebih mudah mempelajari pola ketidaksalamatan, sehingga performa pada kelas positif dapat tertutupi oleh performa pada kelas negatif. Oleh karena itu, diperlukan metrik tambahan seperti presisi dan *recall* untuk mengukur kualitas prediksi pada masing-masing kelas secara lebih komprehensif.

Presisi kelas positif sebesar 83,87% menunjukkan bahwa dari seluruh penumpang yang diprediksi selamat, sebagian besar benar-benar selamat. Hal ini menunjukkan jumlah *false positive* yang rendah, menandakan bahwa model cukup berhati-hati dalam memberi label “selamat” dan tidak menebak secara sembarangan. Dalam konteks kejadian nyata Titanic, nilai presisi ini mengindikasikan bahwa karakteristik penumpang selamat misalnya perempuan dan anak-anak yang mendapat prioritas evakuasi cukup berhasil ditangkap oleh model, meskipun tidak sepenuhnya optimal.

Sebaliknya, *recall* kelas positif berada pada nilai 69,33%, yang berarti model belum mampu menangkap seluruh penumpang yang sebenarnya

selamat. Adanya *FN* yang cukup besar menunjukkan bahwa sejumlah penumpang yang selamat justru diprediksi sebagai tidak selamat. Dari perspektif kondisi lapangan, hal ini mencerminkan tantangan dalam memprediksi kelompok selamat yang jumlahnya lebih sedikit dan sifatnya lebih beragam. Banyak penumpang selamat berasal dari kelas atas atau lokasi kabin tertentu yang dekat dengan akses sekoci, sehingga pola keselamatan mereka tidak seteratur pola ketidaksalamatan.

Nilai *recall* yang relatif lebih rendah dibandingkan metrik lainnya disebabkan oleh karakteristik distribusi kelas pada dataset Titanic yang tidak seimbang, di mana jumlah penumpang yang tidak selamat lebih dominan dibandingkan penumpang yang selamat. Dalam kondisi ini, model regresi logistik cenderung memprioritaskan klasifikasi kelas mayoritas sehingga sebagian observasi pada kelas minoritas (*survived*) masih terklasifikasikan sebagai tidak selamat (*false negative*). Hal ini sesuai dengan penelitian Ihfa & Harsanti, 2020 yang mengatakan “ketidakseimbangan data memberikan dampak yang buruk pada hasil klasifikasi dimana kelas minoritas sering disalah klasifikasikan sebagai kelas mayoritas”.

Selain itu, penggunaan ambang keputusan standar sebesar 0,5 pada probabilitas keluaran regresi logistik turut berkontribusi terhadap rendahnya nilai *recall*, karena ambang ini secara implisit lebih menguntungkan presisi dibandingkan sensitivitas terhadap kelas positif (Siringoringo dkk., 2024).

Masalah fundamental dalam *recall* yang rendah bukanlah terletak pada konvergensi algoritma, melainkan pada bias inheren yang terbentuk akibat ketidakseimbangan data. Perangin-angin, 2020 menjelaskan bahwa “alasan

buruknya kinerja metode klasifikasi biasa yang digunakan pada data tidak seimbang adalah bahwa tujuan metode klasifikasi dalam meminimumkan galat secara keseluruhan tidak dapat tercapai karena kelas minoritas hanya sedikit memberikan kontribusi.” Model regresi logistik secara natural memprioritaskan kelas mayoritas untuk meminimalkan error keseluruhan (Istiana & Mustafiril, 2023), dan menambah iterasi justru akan memperkuat bias ini karena model semakin “yakin” dengan pola yang didominasi kelas mayoritas.

Mengenai pengaruh penambahan iterasi, berdasarkan literatur yang tersedia, penambahan iterasi tidak secara signifikan mempengaruhi masalah fundamental ketidakseimbangan kelas ini. Istiana & Mustafiril, 2023 menekankan bahwa solusi yang lebih efektif adalah penerapan teknik resampling seperti SMOTE untuk menangani ketidakseimbangan distribusi kelas, bukan peningkatan jumlah iterasi training (Siringoringo dkk., 2024).

Nilai F1-score positif sebesar 75,91% menggambarkan keseimbangan antara presisi dan recall pada kelas selamat. F1-score ini menunjukkan bahwa meskipun model cukup baik dalam menandai penumpang yang benar-benar selamat, masih terdapat kekurangan dalam mendeteksi seluruh penumpang selamat secara lengkap. Dengan demikian, meskipun ketepatan prediksi sudah cukup tinggi, kemampuan model untuk menangkap kasus positif sepenuhnya perlu ditingkatkan.

Evaluasi juga dilakukan untuk kelas negatif (kelas 0: tidak selamat). Presisi kelas negatif sebesar 82,96% menunjukkan bahwa sebagian besar prediksi “tidak selamat” sesuai dengan kondisi aktual, dengan FP yang relatif

kecil. Sementara itu, recall kelas negatif mencapai 91,80%, yang merupakan nilai tertinggi di antara seluruh metrik. Nilai recall ini menandakan bahwa model mampu menangkap hampir seluruh penumpang yang benar-benar tidak selamat. Dalam konteks lapangan, hal ini menunjukkan bahwa model sangat sensitif terhadap ciri-ciri kelompok yang memiliki risiko tinggi untuk tidak selamat, seperti laki-laki dewasa dari kelas ekonomi yang aksesnya terbatas menuju dek sekoci. F1-score kelas negatif sebesar 87,20% mempertegas bahwa performa model pada kelas ini sangat stabil dan kuat.

Perbandingan performa kedua kelas menunjukkan pola yang jelas, model jauh lebih unggul dalam memprediksi penumpang tidak selamat dibandingkan penumpang selamat. Hal ini dapat dipahami karena distribusi kelas tidak seimbang, sehingga pola ketidaksalamatan lebih mudah dipelajari oleh model. Perbedaan recall antara kelas positif (69,33%) dan kelas negatif (91,80%) menjadi bukti bahwa model masih kesulitan mengidentifikasi kelompok penumpang yang selamat, yang dalam peristiwa nyata memiliki variasi kondisi yang lebih kompleks.

Secara keseluruhan, kombinasi *TN* yang tinggi, *FP* yang rendah, serta F1-score kelas negatif yang unggul menunjukkan bahwa model memiliki kemampuan kuat dalam memprediksi penumpang yang tidak selamat. Meski demikian, keberadaan *FN* yang cukup besar pada kelas selamat menunjukkan bahwa kemampuan model dalam mendeteksi penumpang selamat belum maksimal. Oleh karena itu, meskipun performa model dapat dikategorikan baik secara umum, terdapat ruang perbaikan terutama dalam meningkatkan recall kelas positif, misalnya melalui penyeimbangan data, penyesuaian threshold

klasifikasi, atau penambahan fitur yang lebih relevan terhadap pola keselamatan penumpang.

5. Perbandingan dengan *Baseline* (Adam dan SGD)

Pada Komparasi optimizer ini akan membahas evaluasi performa metode Gauss–Newton dengan Dekomposisi QR dan dibandingkan dengan dua metode baseline, yaitu Stochastic Gradient Descent (SGD) dan Adam Optimizer. Seluruh pengujian dilakukan menggunakan data test untuk memastikan bahwa perbandingan performa mencerminkan kemampuan generalisasi masing-masing model.

Proses *training* dirancang untuk terus berjalan selama algoritma optimasi masih dapat memperbaiki parameter, tanpa menetapkan batas maksimum epoch maupun target akurasi tertentu. Tidak adanya batas epoch berarti loop pelatihan hanya bergantung pada kondisi internal, misalnya nilai loss yang semakin kecil, gradien yang mendekati nol, atau perubahan parameter yang sudah sangat kecil. Dengan demikian, model tidak dihentikan berdasarkan jumlah iterasi, tetapi berhenti ketika proses optimasi dianggap konvergen. Demikian pula, akurasi tidak dijadikan acuan penghentian. Meskipun akurasi dihitung di setiap epoch untuk kepentingan monitoring, hal tersebut tidak menetapkan bahwa pelatihan harus berhenti setelah akurasi mencapai tingkat tertentu. Alurnya sepenuhnya mengikuti karakteristik metode optimasi yang digunakan, baik itu Gauss-Newton, batch Gradient Descent, SGD, maupun Adam, sehingga pembaruan parameter berlangsung hingga syarat konvergensi

masing-masing metode terpenuhi. Berikut disajikan tabel hasil komparasi data uji untuk tiga metode.

Tabel 4.15 Hasil Komparasi Data Uji untuk Tiga Metode

<i>Metode</i>	<i>Akurasi</i>	<i>Presisi</i>	<i>Recall</i>	<i>F1Score</i>
<i>GN</i>	0,8325	0,8387	0,6933	0,7591
<i>SGD</i>	0,7919	0,7500	0,6800	0,7133
<i>ADAM</i>	0,8223	0,8226	0,6800	0,7445

Dari Tabel 4.14 terlihat bahwa Gauss–Newton memberikan kinerja terbaik pada seluruh metrik evaluasi. Gauss-Newton mencapai akurasi sebesar 0,8325, presisi 0,8387, *recall* 0,6933, dan *F1-score* 0,7591, Hal ini menunjukkan bahwa model dengan optimasi Gauss–Newton paling sering memberikan prediksi yang benar secara keseluruhan, sekaligus memiliki ketepatan yang tinggi saat memprediksi kelas positif (“selamat”). Nilai presisi yang tinggi menandakan bahwa proporsi false positive relatif kecil, sedangkan *recall* yang tertinggi di antara ketiga metode menunjukkan bahwa Gauss-Newton sedikit lebih baik dalam menangkap kasus penumpang yang benar-benar selamat. *F1-score* yang juga paling tinggi menegaskan bahwa Gauss-Newton mampu menjaga keseimbangan antara presisi dan recall secara konsisten.

Metode Adam menempati posisi kedua dengan akurasi 0,8223, presisi 0,8226, *recall* 0,6800, dan *F1-score* 0,7445. Kinerjanya mendekati Gauss–Newton, dengan selisih akurasi sekitar satu poin persentase dan selisih *F1-score* sekitar 1,5 poin. Pola presisi dan *recall* pada Adam menunjukkan bahwa model ini juga cukup selektif dalam memberikan label selamat (presisi tinggi), namun sedikit kurang mampu dibanding Gauss-Newton dalam menangkap seluruh kasus selamat (*recall* sedikit lebih rendah). Dengan demikian, Adam masih menjadi optimizer yang kompetitif, terutama jika dipertimbangkan dari

sisi kemudahan implementasi dan efisiensi komputasi dibandingkan Gauss–Newton.

Metode SGD menunjukkan kinerja terendah di antara ketiga metode, dengan akurasi 0,7919, presisi 0,7500, *recall* 0,6800, dan *F1-score* 0,7133. Nilai-nilai ini mengindikasikan bahwa pada konfigurasi saat ini (tanpa penalaan hyperparameter yang agresif seperti penyesuaian *learning rate*, momentum, atau skema *mini-batch* yang lebih kompleks), SGD kurang efektif baik dari sisi ketepatan maupun keseimbangan antara presisi dan *recall*. Untuk membuat SGD lebih kompetitif, umumnya diperlukan tuning hyperparameter yang lebih ekstensif, misalnya penggunaan *learning rate schedule*, penambahan momentum, atau pengaturan ukuran *mini-batch* yang berbeda.

Secara keseluruhan hasil pada Tabel 4.14 menunjukkan bahwa Gauss–Newton dengan dekomposisi QR layak dijadikan metode utama dalam implementasi dan pembahasan hasil pada penelitian ini. Metode ini tidak hanya unggul pada satu metrik tertentu, tetapi secara konsisten melampaui dua optimizer lain pada akurasi, presisi, *recall*, dan *F1-score*. Hal ini memberikan dasar yang kuat bagi rekomendasi teknis bahwa pendekatan berbasis Gauss–Newton mampu menghasilkan model regresi logistik yang lebih andal untuk kasus klasifikasi keselamatan penumpang Titanic, meskipun dengan konsekuensi biaya komputasi yang umumnya lebih tinggi dibanding metode *first-order* seperti SGD dan Adam.

4.3 Pembahasan

Inti pembahasan mencakup bukti teknis bahwa dekomposisi QR meredam *ill-conditioning* sehingga langkah optimasi menjadi lebih andal dan ringan komputasi, keandalan ini tercermin pada hasil komparatif di mana Gauss-Newton dan Dekomposisi QR tampil kompetitif dengan profil latensi terbaik dibanding pendekatan lain dan karena prosesnya stabil sekaligus efisien, pemodelan kemudian dibingkai oleh prinsip etika keagamaan adil pada semua kelas, transparan dalam pengambilan keputusan, serta diarahkan untuk memberi manfaat nyata bagi manusia. Analisis stabilitas numerik dengan dekomposisi QR.

4.3.1 Analisis Stabilitas Numerik dengan Dekomposisi QR

Seluruh eksperimen pada penelitian ini dijalankan dengan konfigurasi praproses dan pengaturan pelatihan yang konsisten agar perbandingan antar-metode optimasi bersifat adil. Sebelum masuk ke proses training, seluruh fitur numerik distandardisasi terlebih dahulu sehingga memiliki skala yang sebanding, dan diterapkan *clipping* pada nilai logit di rentang ± 500 untuk menjaga stabilitas perhitungan fungsi sigmoid dan *log-loss*. Pada tahap pelatihan, data latih dibagi ke dalam mini-batch berukuran 32 sampel. Dengan ukuran dataset train yang digunakan, satu epoch setara dengan 29 *mini-batch*. Setelah seluruh batch dalam satu epoch diproses, model selalu dievaluasi pada *validation set* sehingga dinamika konvergensi dapat dipantau dari waktu ke waktu, baik dari sisi *loss* maupun akurasi.

Untuk menjaga efisiensi komputasi sekaligus menghindari *overfitting*, penelitian ini menggunakan mekanisme early stopping berbasis akurasi validasi. Secara garis besar, pelatihan akan dihentikan apabila nilai val akurasi sudah

mencapai atau melampaui ambang 0,75 dan tidak menunjukkan peningkatan berarti lagi dalam beberapa epoch berturut-turut (patience berbasis epoch). Dengan strategi ini, jumlah epoch yang ditempuh setiap metode tidak ditentukan oleh batas maksimum epoch yang kaku, melainkan mengikuti dinamika konvergensi masing-masing optimizer. Metode yang lebih cepat menemukan parameter yang baik akan berhenti lebih dini, sedangkan metode yang konvergensinya lambat akan berjalan lebih lama sampai memenuhi kriteria berhenti.

Dalam penelitian ini dibandingkan tiga skema optimasi, yaitu Gauss–Newton, Stochastic Gradient Descent (SGD), dan Adam. Metode Gauss–Newton memperbarui parameter dengan memanfaatkan informasi orde dua, yakni sensitivitas kesalahan terhadap parameter yang dirangkum dalam matrix Jacobian, lalu menyelesaikan sistem linier pada setiap langkah menggunakan dekomposisi QR. SGD merupakan metode berbasis gradien orde satu dengan laju belajar tetap yang memperbarui bobot secara bertahap menggunakan informasi gradien pada setiap *mini-batch*. Sementara itu, Adam menggunakan skema adaptif yang menggabungkan ide momentum dan penyesuaian laju belajar per-parameter, sehingga secara praktik sering kali konvergen lebih cepat dan stabil dibanding SGD standar.

Ringkasan titik konvergensi terbaik untuk masing-masing metode disajikan pada Tabel 4.11 dan Tabel 4.12. Dari hasil tersebut terlihat bahwa Gauss–Newton mencapai val akurasi sebesar 0,7857 hanya dalam 1 epoch, atau dengan kata lain sudah memenuhi kriteria *early stopping* pada epoch pertama. Sebaliknya, Adam memerlukan 20 epoch untuk mencapai performa validasi

terbaiknya, sedangkan SGD baru menembus ambang yang sama setelah 97 epoch. Perbedaan ini juga tercermin pada waktu eksekusi Gauss–Newton hanya memerlukan sekitar 0,02 detik, sementara Adam membutuhkan kurang lebih 5,42 detik, dan SGD sekitar 12,09 detik hingga mencapai titik konvergensi masing-masing.

Hasil ini secara jelas menunjukkan bahwa, dalam konfigurasi eksperimen yang sama, Gauss–Newton dengan dekomposisi QR merupakan metode yang paling cepat mencapai konvergensi, baik dari sisi jumlah epoch maupun waktu pelatihan aktual. Adam menempati posisi kedua dengan kecepatan dan stabilitas yang cukup baik, sedangkan SGD menjadi metode paling lambat karena membutuhkan epoch dan waktu pelatihan paling besar untuk mencapai performa yang memadai. Dengan demikian, dari perspektif konvergensi dan efisiensi komputasi, Gauss–Newton dan Deomposisi QR terbukti memberikan keuntungan yang signifikan dibanding dua optimizer berbasis gradien yang dijadikan pembandingan.

4.3.2 Perbandingan dengan Penelitian Sebelumnya

Tabel 4.16 menyajikan perbandingan hasil penelitian dengan karya-karya terdahulu yang menggunakan dataset Titanic dan melaporkan metrik evaluasi seperti akurasi, presisi, *recall*, *F1-score*, serta bila tersedia waktu komputasi. Tujuannya adalah memberi konteks kuantitatif bagi temuan penelitian ini, khususnya pendekatan regresi logistik yang dioptimasi Gauss-Newton, terhadap beragam metode yang pernah diuji pada skenario yang sama. Perlu dicatat bahwa variasi angka antarstudi wajar terjadi akibat perbedaan pembagian data

(*train/valid/test* atau *cross-validation*), teknik prapemrosesan dan imputasi, *feature engineering* (mis. *Title*, *FamilySize*), penyeimbangan kelas, definisi metrik (mis. *weighted average*), serta penataan hiperparameter.

Tabel 4.16 Perbandingan Hasil Penelitian dengan Penelitian Sebelumnya

Author (Tahun)	Metode	Akurasi	Prec	Recall	F1-score
Penelitian ini	(Gauss–Newton + QR)	0,8325	0,8387	0,6933	0,7591
(Gandhi, dkk., 2025)	Decision Tree	0,7754	0,78	0,78	0,78
	Random Forest	0,7836	0,78	0,78	0,78
	XGBoost	0,7914	0,79	0,79	0,78
	CatBoost	0,8182	0,82	0,82	0,82
(Hasan, 2025)	Logistic Regression	0,7940	-	-	-
	Decision Tree	0,7720	0,76	0,72	0,74
	Random Forest	0,8150	0,80	0,78	0,79
	SVM	0,8210	0,81	0,79	0,80
	Neural Network	0,8430	0,83	0,81	0,82

Tabel 4.16 menyajikan perbandingan hasil penelitian ini dengan dua karya terdahulu yang menggunakan dataset Titanic, yaitu (Gandhi, dkk., 2025) dan (Hasan, 2025). Metrik yang dibandingkan meliputi akurasi, presisi, *recall*, *F1-score*, serta waktu komputasi jika dilaporkan. Tujuan perbandingan ini adalah memberikan konteks kuantitatif bagi kinerja regresi logistik yang dioptimasi dengan Gauss–Newton dan dekomposisi QR pada penelitian ini terhadap berbagai metode lain yang pernah diuji pada kasus Titanic.

Perbedaan nilai antarstudi merupakan hal yang wajar karena dipengaruhi oleh banyak faktor, antara lain pembagian data (misalnya *hold-out* 70:30 versus skema *train/val/test*), teknik prapemrosesan dan imputasi, rekayasa fitur (seperti *Title*, *FamilySize*, *Fare per Person*), strategi penyeimbangan kelas, definisi metrik (misalnya *weighted average*), serta pengaturan hiperparameter yang digunakan.

Berdasarkan Tabel 4.16, regresi logistik pada penelitian ini yang dioptimasi dengan Gauss–Newton dan Dekomposisi QR mencapai akurasi 0,8325, presisi 0,8387, recall 0,6933, F1-score 0,7591, dengan waktu komputasi sekitar 0,01 detik. Nilai-nilai ini menempatkan model penulis pada kisaran performa yang kompetitif bahkan bila dibandingkan dengan model-model yang secara arsitektur lebih kompleks.

Pada studi (Gandhi, dkk., 2025), berbagai model pohon dan boosting diujikan pada dataset Titanic. Decision Tree, Random Forest, dan XGBoost menghasilkan akurasi di kisaran 0,77–0,79 dengan presisi, *recall*, dan *F1-score* yang relatif seimbang (sekitar 0,78–0,79). Model CatBoost menjadi metode terbaik di studi tersebut dengan akurasi sekitar 0,8182 dan presisi, recall, serta F1-score yang masing-masing berada di kisaran 0,82. Jika dibandingkan, regresi logistik Gauss–Newton dan dekomposisi QR pada penelitian ini memperoleh akurasi dan *F1-score* yang sedikit lebih tinggi daripada CatBoost, meskipun secara model jauh lebih sederhana (linear) dan menggunakan lebih sedikit fitur. Hal ini menunjukkan bahwa pemilihan metode optimasi dan pengelolaan aspek numerik dapat memberikan kontribusi yang signifikan terhadap kualitas model, bahkan ketika dibandingkan dengan algoritma boosting yang umumnya sangat kuat pada dataset tabular.

Pada studi (Hasan, 2025), diuji beberapa algoritma populer, antara lain *Logistic Regression*, Decision Tree, Random Forest, SVM, dan Neural Network. Hasilnya menunjukkan bahwa *Logistic Regression* standar mencapai akurasi sekitar 0,7940, sementara model-model non-linear seperti Random Forest, SVM, dan khususnya Neural Network mampu mendorong akurasi hingga sekitar 0,8430

dengan F1-score sekitar 0,82. Dengan demikian, hasil penelitian ini (akurasi 0,8325, F1 0,7591) sedikit berada di bawah Neural Network (Hasan, 2025) yang lebih kompleks, tetapi jelas melampaui *Logistic Regression baseline* (0,7940) dan bersaing dengan Random Forest dan SVM yang berada di rentang 0,8150–0,8210, Artinya, regresi logistik yang dioptimasi Gauss–Newton dan dekomposisi QR mampu “mengejar” performa model non-linear yang lebih canggih pada studi Hasan, sekaligus mempertahankan sifat interpretabilitas yang menjadi keunggulan regresi logistik.

Perlu ditekankan bahwa setup eksperimen antara penelitian ini dan dua rujukan tidak sepenuhnya sebanding (*apple-to-apple*). (Gandhi, dkk., 2025) dan (Hasan, 2025) umumnya menggunakan dataset Titanic versi train dengan sekitar 891 baris dan skema hold-out sederhana (misalnya 70:30), serta memanfaatkan rekayasa fitur yang cukup agresif (seperti *Title*, *FamilySize*, dan turunan lainnya) untuk mendorong akurasi. Sementara itu, penelitian ini menggunakan 1,309 baris data dengan skema stratified 70:15:15 (*train/validation/test*) sehingga terdapat validation set eksplisit untuk pemilihan model dan pengecekan konvergensi. Selain itu, penelitian ini secara sengaja membatasi fitur pada tujuh fitur inti (*Sex*, *Fare*, *Pclass*, *Embarked*, *Age*, *SibSp*, *Parch*) agar fokus analisis berada pada kontribusi metode optimasi Gauss–Newton dekomposisi QR, bukan pada kekayaan rekayasa fitur. Karena itu, perbandingan angka akurasi dan *F1-score* sebaiknya dibaca dengan konteks bahwa literatur terdahulu banyak menekankan maksimum akurasi melalui rekayasa fitur dan split sederhana, sementara penelitian ini menekankan protokol evaluasi yang lebih ketat dan desain fitur yang lebih minimalis.

Dari sisi proses komputasi, penelitian ini menerapkan skema pelatihan yang dirancang untuk berjalan selama algoritma optimasi masih dapat memperbaiki parameter, tanpa menetapkan batas maksimum epoch maupun target akurasi tertentu. Tidak adanya batas epoch berarti *loop training* dikendalikan sepenuhnya oleh kondisi internal optimasi, seperti penurunan nilai loss, gradien yang semakin kecil, atau perubahan parameter yang mendekati nol. Dengan demikian, model tidak dihentikan berdasarkan jumlah iterasi, tetapi berhenti ketika proses optimasi dianggap konvergen sesuai kriteria metode yang digunakan. Akurasi tetap dihitung di setiap epoch untuk keperluan pemantauan, namun tidak menjadi dasar penghentian pelatihan. Pendekatan ini konsisten dengan karakter Gauss–Newton yang beroperasi sebagai metode orde dua mirip line-search, ketika solusi *least-squares* tercapai, pembaruan parameter akan secara alami mengecil dan akhirnya berhenti.

Secara keseluruhan, tabel perbandingan ini menunjukkan pola bahwa metode boosting dan Neural Network memang cenderung mendominasi akurasi puncak pada Titanic, terutama jika didukung rekayasa fitur yang kaya. Namun, regresi logistik yang dioptimasi dengan Gauss–Newton dan dekomposisi QR menawarkan profil yang berbeda, performa yang kompetitif (melampaui *Logistic Regression* standar dan bersaing dengan Random Forest, SVM, serta CatBoost), waktu komputasi yang sangat rendah, stabilitas numerik yang baik, dan interpretabilitas model yang tinggi. Dengan demikian, pendekatan ini layak diposisikan sebagai baseline yang kuat dan praktis, terutama pada konteks di mana kecepatan pelatihan, ketahanan numerik, dan transparansi proses lebih krusial daripada sekadar mengejar akurasi absolut tertinggi.

4.3.3 Penerapan Metode Gauss-Newton dan Dekomposisi QR dalam Pandangan Islam

Islam memandang ilmu pengetahuan sebagai sarana untuk mengenal kebesaran Allah dan mewujudkan kemaslahatan bagi seluruh umat manusia. Dalam Al-Qur'an, banyak ayat yang mendorong manusia untuk berpikir, meneliti, dan menggali rahasia alam semesta.

Pada Q.S Al-Mujadilah : 11, ayat ini menunjukkan bahwa ilmu memiliki posisi yang sangat mulia dalam Islam. Setiap penemuan dan metode ilmiah yang dikembangkan oleh manusia sejatinya adalah bagian dari proses mengenal ciptaan Allah melalui rasionalitas dan penelitian ilmiah. Dalam konteks modern, pengembangan metode komputasi dan analisis data seperti metode Gauss-Newton dan Dekomposisi QR dalam regresi logistic merupakan wujud nyata dari perintah untuk berpikir dan mengoptimalkan akal (Rofina dkk., 2024).

Penelitian tentang penerapan metode Gauss-Newton dan dekomposisi QR dalam optimasi model regresi logistik berhubungan erat dengan upaya manusia untuk mencari solusi terbaik melalui pendekatan matematis dan teknologi. Penelitian ini berfokus pada bagaimana metode tersebut digunakan untuk mengoptimalkan model regresi logistik dalam klasifikasi biner, yakni mengelompokkan data menjadi dua kategori, misalnya selamat atau tidak selamat pada kasus penumpang Titanic. Pendekatan ini menunjukkan bagaimana ilmu matematika terapan dapat berkontribusi pada pengambilan keputusan berbasis data. Metode Gauss-Newton digunakan untuk menyelesaikan permasalahan nonlinear melalui pendekatan iteratif terhadap solusi optimal, sedangkan dekomposisi QR dimanfaatkan untuk memperbaiki stabilitas numerik dan

efisiensi dalam proses komputasi. Dengan menggabungkan keduanya, proses optimasi menjadi lebih akurat dan efisien, mencerminkan prinsip Islam tentang ihsan (melakukan yang terbaik) dalam setiap pekerjaan.

Penelitian ini juga menyoroti analisis performa kedua metode tersebut dalam pengklasifikasian keselamatan penumpang pada dataset Titanic. Hal ini menggambarkan penerapan nyata ilmu sains dan teknologi untuk memahami pola dan faktor-faktor keselamatan manusia. Dalam perspektif Islam, penelitian semacam ini termasuk dalam bentuk ikhtiar ilmiah untuk memperoleh pengetahuan yang dapat digunakan bagi peningkatan keselamatan dan kesejahteraan hidup manusia. Sebagaimana firman Allah dalam Surah Al-Ankabut ayat 20.

Ayat ini menegaskan pentingnya observasi dan penelitian terhadap fenomena dunia, agar manusia mampu memahami hukum-hukum alam yang telah Allah tetapkan. Dengan memahami pola-pola tersebut melalui pendekatan ilmiah, manusia dapat membuat keputusan yang lebih bijak dan bermanfaat. Penelitian ini juga merefleksikan prinsip tauhid dalam ilmu pengetahuan, yakni bahwa segala bentuk penelitian dan teknologi sejatinya bersumber dari Allah, dan harus dikembalikan untuk kemaslahatan umat. Ketika seorang peneliti menggunakan metode matematika untuk memecahkan permasalahan nyata, ia sedang menjalankan amanah sebagai khalifah di bumi untuk memakmurkan kehidupan dengan ilmu (Hisan, 2016).

Dengan demikian, penelitian ini menjadi salah satu bentuk implementasi ilmu yang bermanfaat bagi umat manusia, sejalan dengan ajaran Islam yang mendorong umatnya untuk selalu belajar dan mencari solusi terbaik dalam

kehidupan. Penerapan metode Gauss-Newton dan Dekomposisi QR dalam analisis keselamatan penumpang Titanic mencerminkan bagaimana ilmu pengetahuan dan teknologi dapat menjadi alat untuk meningkatkan kesejahteraan manusia, sebagaimana Islam mengajarkan bahwa ilmu harus digunakan untuk kebaikan dan kemajuan peradaban.

BAB V

KESIMPULAN

5.1 Kesimpulan

Berdasarkan rumusan masalah dan pembahasan pada bab sebelumnya, maka diperoleh kesimpulan sebagai berikut:

1. Penerapan metode Gauss-Newton dalam optimasi regresi logistik dilakukan dengan membangun fungsi log-likelihood dan menghitung matriks Jacobian sebagai representasi sensitivitas kesalahan terhadap parameter model. Pada setiap iterasi, langkah pembaruan parameter diperoleh dari penyelesaian sistem persamaan Gauss-Newton, yang secara numerik diselesaikan menggunakan dekomposisi QR. Penggunaan dekomposisi QR menggantikan invers matriks langsung sehingga sistem persamaan dapat diselesaikan secara stabil, menghindari masalah singularitas, dan memastikan proses optimasi berjalan konsisten hingga konvergen.
2. Regresi logistik yang dioptimasi menggunakan Gauss-Newton dan dekomposisi QR menunjukkan performa klasifikasi terbaik dibandingkan metode pembanding. Pada data uji, model mencapai akurasi sekitar 83%, presisi 84%, *recall* 69%, dan *F1-score* 76%, yang mencerminkan kemampuan klasifikasi yang baik dan seimbang. Selain unggul dari sisi performa, metode ini juga sangat efisien secara komputasi, karena mencapai konvergensi hanya dalam satu epoch dengan waktu pelatihan yang jauh lebih singkat dibandingkan SGD dan Adam. Dengan demikian, Gauss–Newton dengan dekomposisi QR terbukti efektif dan efisien untuk optimasi regresi logistik pada klasifikasi keselamatan penumpang Titanic.

5.2 Saran

Berdasarkan hasil penelitian yang diperoleh, terdapat beberapa saran yang dapat diberikan untuk mengembangkan penelitian selanjutnya, maka penulis memberikan saran berikut:

1. Nilai *recall* yang masih relatif rendah menunjukkan bahwa sebagian penumpang selamat belum berhasil terdeteksi oleh model. Untuk mengatasi keterbatasan ini, penelitian selanjutnya disarankan melakukan penyesuaian ambang keputusan pada probabilitas keluaran regresi logistik atau menerapkan pembobotan kelas agar model lebih sensitif terhadap kelas *survived*. Pendekatan tersebut dinilai lebih tepat dibandingkan menambah iterasi pelatihan, karena model telah mencapai konvergensi dan optimasi Gauss–Newton dengan dekomposisi QR sudah berjalan stabil.
2. Disarankan menguji metode Gauss–Newton dan Dekomposisi QR pada dataset berbeda yang bervariasi dari sisi banyaknya fitur, pola distribusi, serta skala data, guna memastikan konsistensi hasil dan reliabilitasnya pada berbagai skenario klasifikasi biner.
3. Untuk memperluas pemahaman, penelitian selanjutnya dapat mengevaluasi Gauss–Newton terhadap Levenberg–Marquardt dan BFGS saat melatih regresi logistik, dengan membandingkan metrik seperti Akurasi, presisi, *recall*, *F1-score*, dan waktu pemrosesan.

DAFTAR PUSTAKA

- Abdurrahman, M. R., Ispriyanti, D., & Prahutama, A. (2014). Pembentukan Pohon Klasifikasi Biner Dengan Algoritman Quest (Quick, Unbiased, And Efficient Statistical Tree) Pada Data Pasien Liver. *Jurnal Gaussian*, 3(4), 731–739. <http://ejournal-s1.undip.ac.id/index.php/gaussian>
- Agresti, A. (2012). *Categorical data analysis* (Vol. 792). John Wiley & Sons.
- Björck, Å. (1996). *Numerical methods for least squares problems*. SIAM.
- Bolikulov, F., Nasimov, R., Rashidov, A., Akhmedov, F., & Cho, Y.-I. (2024). Effective methods of categorical data encoding for artificial intelligence algorithms. *Mathematics*, 12(16), 2553.
- Cerda, P., Varoquaux, G., & Kégl, B. (2018). Similarity encoding for learning with dirty categorical variables. *Machine Learning*, 107(8), 1477–1494.
- Dahouda, M. K., & Joe, I. (2021). A deep-learned embedding technique for categorical features encoding. *IEEE Access*, 9, 114381–114391.
- Dora, L., Agrawal, S., Panda, R., & Abraham, A. (2017). PT US CR. *Expert Systems With Applications*. <https://doi.org/10.1016/j.eswa.2017.05.035>
- Estuningtyas, R. D. (2018). Ilmu dalam Perspektif al-Qur'an. *Qof*, 2(2), 203–216.
- Gallant, A. R. (1985). *Nonlinear Statistical Models*. 1682.
- Gandhi, H., Attwal, K. P. S., & Lal, M. (2025). A Comparison of Machine Learning Techniques for Prediction of Survival using Titanic Dataset. *A Comparison of Machine Learning Techniques for Prediction of Survival Using Titanic Dataset*.
- Golub, G. H., & Van Loan, C. F. (2013). *Matrix computations*. JHU press.
- Ha, J., Kambe, M., & Pe, J. (2011). Data Mining: Concepts and Techniques. In *Data Mining: Concepts and Techniques*. <https://doi.org/10.1016/C2009-0-61819-5>
- Hasan, A. T. M. (2025). *Using Titanic Dataset for Comprehensive Machine Learning Model Training*. 9(10), 3063–3065.
- Hisan, A. (2016). *Nilai-nilai Pendidikan Akidah Akhlak yang Terkandung dalam QS. Al-Ankabut ayat 8-11*. jakarta: FITK UIN Syarif Hidayatullah Jakarta.
- Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied logistic regression*. John Wiley & Sons.

- Hutagalung, M. A. I., & Sya'ban, M. Y. P. El. (2024). Stochastic Gradient Descents Optimizer and Its Variants: Performance of the Optimizers for Multinomial Logistic Models on Large Data Sets by Simulation. *Mathematical Modelling of Engineering Problems*, 11(10).
- Ihfa, R., & Harsanti, T. (2020). Komparasi Teknik Resampling Pada Pemodelan Regresi Logistik Biner. *Seminar Nasional Official Statistics*, 2020(1), 863–870.
- Istiana, N., & Mustafiril, A. (2023). Perbandingan metode klasifikasi pada data dengan imbalance class dan missing value. *Jurnal Informatika*, 10(2), 101–108.
- Jamhuri, M. (n.d.). *Performance Improvement of Logistic Regression for Binary Classification by Gauss-Newton Method*.
- Jamhuri, M., Irawan, M. I., Mukhlash, I., Iqbal, M., & Puspaningsih, N. N. T. (2025). Neural Networks Optimization Via Gauss-Newton Based Qr Factorization on Sars-Cov-2 Variant Classification. *Available at SSRN* 4672737.
- Jamhuri, M., Mukhlash, I., & Irawan, M. I. (2022). Performance Improvement of Logistic Regression for Binary Classification by Gauss-Newton Method. *Proceedings of the 2022 5th International Conference on Mathematics and Statistics*, 12–16.
- Kementerian Agama. (2017). Al-Qur'an dan Terjemahan dilengkapi Panduan Waqaf & Ibtida'. In *PT. Suara Agung*.
- Kulle, H. (2016). Nilai-nilai Tarbawi dalam Surah Al-Mujadalah Ayat 11.pdf. In *Jurnal Ilmiah Ilmu Dasar Keislaman: Vol. IV* (pp. 1–66).
- Lubis, L., & Asry, W. (2020). *Ilmu Pendidikan Islam*.
- Mahesh, B. (2020). Machine Learning Algorithms - A Review. *International Journal of Science and Research (IJSR)*, 9(1), 381–386. <https://doi.org/10.21275/art20203995>
- Mukmin, T. (2016). Urgensi Belajar dalam Perspektif Al-Qur'an Surat Al-Alaq Ayat 1-5 Menurut Tafsir Ibnu Katsir. *El-Ghiroh: Jurnal Studi Keislaman*, 11(2), 1–21.
- Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.
- Nata, H. A. (2016). *Pendidikan dalam perspektif Al-Qur'an*. Prenada Media.
- Nurjan, S. (2016). *Psikologi belajar*. Wade Group.

- Octaviana, D. R., & Ramadhani, R. A. (2021). HAKIKAT MANUSIA: Pengetahuan (Knowledge), Ilmu Pengetahuan (Sains), Filsafat Dan Agama. *Jurnal Tawadhu*, 5(2), 143–159.
- Perangin-angin, R. (2020). Penerapan Algoritma Safe-Level-SMOTE Untuk Peningkatan Nilai G-Mean Dalam Klasifikasi Data Tidak Seimbang. *Journal Information System Development (ISD)*, 5(2).
- Poslavskaya, E., & Korolev, A. (2023). Encoding categorical data: Is there yet anything 'hotter' than one-hot encoding? *ArXiv Preprint ArXiv:2312.16930*.
- Raschka, S., & Mirjalili, V. (2017). Python machine learning second edition. *Birmingham, England: Packt Publishing*.
- Reza, S., Scholar, M. S., Sciences, M., & Kiet, P. A. F. (2021). *Application of Logistic Regression on Passenger Survival Data of the Titanic Liner*. 7(4), 861–867.
- Rofina, A., Ilmi, M. N. A. H., Nursyamsiyah, S., & Huda, H. (2024). Konsep Ilmu dan Pendidikan dalam Perspektif Surat Al-Mujadalah Ayat 11. *TARLIM: Jurnal Pendidikan Agama Islam*, 7(1), 107–119.
- Schraudolph, N. N. (1999). A fast, compact approximation of the exponential function. *Neural Computation*, 11(4), 853–862.
- Schreiber-Gregory, D., & Bader, K. (2018). Logistic and Linear Regression Assumptions: Violation Recognition and Control. *SESUG Paper 247, May*, 1–22. <https://www.researchgate.net/publication/341354759>
- Semendawai, J. N., Stiawan, D., & Pahendra, I. (2024). *Klasifikasi Shellcode dengan Machine Learning Berbasis Klasifikasi Biner*. 5(9), 1514–1526.
- Siringoringo, R., Arisandi, D., Kurniawan, E., & Nababan, E. B. (2024). Model klasifikasi dengan logistic regression dan recursive feature elimination pada data tidak seimbang. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 11(4), 735–742.
- Syakir, S. A. (2012). *Tafsir Ibnu Katsir*. Dar al Sunnah Press.
- Venkataramana, M., Subbarayudu, M., Rajani, M., & Sreenivasulu, K. N. (2016). Regression analysis with categorical variables. *International Journal of Statistics and Systems*, 11(2), 135–143.
- Wahidi, R. (2016). *Tafsir Ayat-Ayat Tarbawi Tafsir dan Kontekstualisasi Ayat-Ayat Pendidikan*. Trussmedia Grafika.
- Wali, R. (2022). *Xtreme Margin: A Tunable Loss Function for Binary Classification Problems*. 1–10. <http://arxiv.org/abs/2211.00176>

- Zhang, S., Baharlouei, E., & Wu, P. (2020). High accuracy matrix computations on neural engines: A study of QR factorization and its applications. *Proceedings of the 29th International Symposium on High-Performance Parallel and Distributed Computing*, 17–28.
- Zhu, W., Qiu, R., & Fu, Y. (2024). Comparative study on the performance of categorical variable encoders in classification and regression tasks. *ArXiv Preprint ArXiv:2401.09682*.

LAMPIRAN

Lampiran 1. Tabel Asli Penumpang Kapal Titanic

pclass	survived	name	sex	age	siblings	parth	ticket	fare	cabin	embarked	boat	body	home.dest
1	1	Allen, Miss. Elisabeth Walton	female	29	0	0	24160	211,337.5	B5	S	2		St Louis, MO
1	1	Allison, Master. Hudson Trevor	male	0,9167	1	2	113781	151,5500	C22 C26	S	11		Montreal, PQ / Chesterville, ON
1	0	Allison, Miss. Helen Loraine	female	2	1	2	113781	151,5500	C22 C26	S			Montreal, PQ / Chesterville, ON
1	0	Allison, Mr. Hudson Joshua Creighton	male	30	1	2	113781	151,5500	C22 C26	S		135	Montreal, PQ / Chesterville, ON
1	0	Allison, Mrs. Hudson J C (Bessie Waldo Daniels)	female	25	1	2	113781	151,5500	C22 C26	S			Montreal, PQ / Chesterville, ON
1	1	Anderson, Mr. Harry	male	48	0	0	19952	26,5500	E12	S	3		New York, NY
1	1	Andrews, Miss. Kornelia Theodosia	female	63	1	0	13502	77,9583	D7	S	10		Hudson, NY
...
3	0	Zakarian, Mr. Mapriededer	male	26,5	0	0	2656	7,2250		C		304	
3	0	Zakarian, Mr. Ortin	male	27	0	0	2670	7,2250		C			
3	0	Zimmerman, Mr. Leo	male	29	0	0	315082	7,8750		S			


```

# Logistic Regression (Titanic) – Adam (Keras) vs SGD
# (Keras) vs GAUSS-NEWTON+QR Mini-Batch (NumPy)
# - Dataset: /kaggle/input/titanic3/titanic3.xls
# - Anti-leakage: impute/encoding/scale fit di TRAIN
# - Early stop TUNGGAL: val_akurasiuracy >= 0,75

import numpy as np
import pandas as pd
import google.protobuf

from sklearn.model_selection import train_test_split
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import (
    akurasiuracy_score, precision_score,
    recall_score,
    f1_score, confusion_matrix, classification_report
)

# KERAS: impor SETELAH tensorflow utama (sel 2)
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.optimizers import Adam, SGD
from tensorflow.keras.callbacks import Callback

print("protobuf:", google.protobuf.__version__)
print("tensorflow:", tf.__version__)

# Reproducibility
SEED = 42
random.seed(SEED)
np.random.seed(SEED)
tf.random.set_seed(SEED)

# Hyperparams umum
TARGET_AKURASI = 0,75          # kriteria berhenti
                                tunggal
EPOCHS_TF = 200                # batas maksimum epoch untuk
                                Adam/SGD (akan berhenti sendiri)
BATCH_SIZE = 32
LR_ADAM = 1e-3
LR_SGD = 1e-3                  # turunkan bila agresif
                                (mis. 2e-2 / 1e-2)
MOMENTUM = 0,0                 # 0,0 = pure GD; bisa 0,9
                                untuk momentum

# GAUSS-NEWTON (NumPy) hyperparams (mini-batch)
EPOCHS_GAUSS-NEWTON_MB = 200

```

```

LR_GAUSS-NEWTON_EPOCH = 1e-3      # "base per-epoch"
step size; distabilkan per-batch
DAMPING                = 0          # regularisasi ( $JTJ + \lambda I$ ) saat
fallback
GAUSS-NEWTON_BATCH     = 32
GAUSS-NEWTON_VERBOSE   = True

# (Optional) tampilkan file di /kaggle/input
for dirname, _, filenames in
os.walk('/kaggle/input'):
    for filename in filenames:
        if 'titanic3' in dirname:
            print(os.path.join(dirname, filename))

pd.read_excel("/kaggle/input/titanic3/titanic3.xls")
df = df.drop(columns=['name', 'ticket', 'body',
'boat', 'home.dest', 'cabin'], errors='iGauss-
Newtonore')
assert 'survived' in df.columns, "Kolom target
'survived' tidak ditemukan."

y = df['survived'].astype(int).values
X = df.drop(columns=['survived'])

X_train, X_temp, y_train, y_temp = train_test_split(
    X, y, test_size=0,30, random_state=SEED,
stratify=y
)
X_val, X_test, y_val, y_test = train_test_split(
    X_temp, y_temp, test_size=0,50,
random_state=SEED, stratify=y_temp
)
print(f"Train: {X_train.shape}, Val: {X_val.shape},
Test: {X_test.shape}")

def missing_nonzero(df, label):
    t = df.isna().sum()
    t = t[t > 0].sort_values(ascending=False)
    return pd.DataFrame({label: t})

nz = pd.concat([
    missing_nonzero(X_train, 'Train'),
    missing_nonzero(X_val, 'Validation'),
    missing_nonzero(X_test, 'Test'),
], axis=1).fillna(0).astype(int)

display(nz)

```

```

num_cols = [c for c in ['age', 'fare'] if c in
X_train.columns]
cat_cols = [c for c in ['sex', 'embarked'] if c in
X_train.columns]

if num_cols:
    imp_num =
SimpleImputer(strategy="median").fit(X_train[num_cols
])
    X_train[num_cols] =
imp_num.transform(X_train[num_cols])
    X_val[num_cols] =
imp_num.transform(X_val[num_cols])
    X_test[num_cols] =
imp_num.transform(X_test[num_cols])

if 'sex' in cat_cols:
    imp_sex =
SimpleImputer(strategy="most_frequent").fit(X_train[[
'sex']])
    for S in (X_train, X_val, X_test):
        S[['sex']] = imp_sex.transform(S[['sex']])
        S['sex'] =
S['sex'].astype(str).str.lower().str.strip().map({'ma
le':0, 'female':1}).fillna(0).astype(int)

if 'embarked' in cat_cols:
    imp_emb =
SimpleImputer(strategy="most_frequent").fit(X_train[[
'embarked']])
    for S in (X_train, X_val, X_test):
        S[['embarked']] =
imp_emb.transform(S[['embarked']])
        S['embarked'] =
S['embarked'].astype(str).str.upper().str.strip()

    # one-hot fit dari train; val/test reindex agar
kolom konsisten
    X_train = pd.get_dummies(X_train,
columns=['embarked'], prefix='emb', drop_first=True)
    X_val =
pd.get_dummies(X_val, columns=['embarked'],
prefix='emb', drop_first=True)
    X_test =
pd.get_dummies(X_test, columns=['embarked'],
prefix='emb', drop_first=True)
    X_val = X_val.reindex(columns=X_train.columns,
fill_value=0)

```

```

X_test = X_test.reindex(columns=X_train.columns,
fill_value=0)

scaler =
StandardScaler().fit(X_train.values.astype('float32')
)
X_train_s =
scaler.transform(X_train.values.astype('float32'))
X_val_s =
scaler.transform(X_val.values.astype('float32'))
X_test_s =
scaler.transform(X_test.values.astype('float32'))

input_dim = X_train_s.shape[1]

def sigmoid_np(z):
    z = np.clip(z, -500, 500)
    return 1,0/(1,0 + np.exp(-z))

def eval_metrics_np(y_true_bin, y_prob, thr=0,5):
    y_pred = (y_prob >= thr).astype(int).ravel()
    y_true = y_true_bin.astype(int).ravel()
    akurasi = akurasiuracy_score(y_true, y_pred)
    prec = precision_score(y_true, y_pred,
zero_division=0)
    rec = recall_score(y_true, y_pred,
zero_division=0)
    f1 = f1_score(y_true, y_pred, zero_division=0)
    return akurasi, prec, rec, f1

# ===== KERAS: Callback early-stop target akurasi
=====
class StopOnValAkurasi(Callback):
    def __init__(self, target=0,75):
        super().__init__()
        self.target = target
    def on_epoch_end(self, epoch, logs=None):
        #v = (logs or {}).get('val_akurasiuracy')
        v = (logs or {}).get('val_akurasiuracy')
        if v is not None and v >= self.target:
            print(f"\n✔ Stop: val_akurasiuracy
{v:.4f} ≥ {self.target} @ epoch {epoch+1}")
            self.model.stop_training = True

# ===== KERAS: build / train / evaluate =====
def build_logreg(input_dim: int) -> Sequential:
    return Sequential([ Dense(1,
activation='sigmoid', input_shape=(input_dim,)) ])

```

```

def train_keras_optimizer(opt_name: str, Xtr, ytr,
                           Xva, yva,
                           lr_adam=LR_ADAM,
                           lr_sgd=LR_SGD, momentum=MOMENTUM,
                           epochs=EPOCHS_TF,
                           batch_size=BATCH_SIZE,
                           target_akurasi=TARGET_AKURASI):
    tf.keras.backend.clear_session()
    model = build_logreg(input_dim)
    if opt_name.lower() == 'adam':
        optimizer = Adam(learning_rate=lr_adam)
    elif opt_name.lower() == 'sgd':
        optimizer = SGD(learning_rate=lr_sgd,
                        momentum=momentum, nesterov=False)
    else:
        raise ValueError("opt_name harus 'adam' atau 'sgd'.")
    model.compile(optimizer=optimizer,
                  loss='binary_crossentropy', metrics=['akurasiuracy'])
    hist = model.fit(Xtr, ytr, validation_data=(Xva,
                                                yva),
                    epochs=epochs,
                    batch_size=batch_size, verbose=1,
                    callbacks=[StopOnValAkurasi(target_akurasi)])
    return model, hist

def evaluate_on_test(name, model, Xte, yte):
    y_prob = model.predict(Xte).ravel()
    y_pred = (y_prob >= 0.5).astype(int)
    akurasi = akurasiuracy_score(yte, y_pred)
    prec = precision_score(yte, y_pred,
                           zero_division=0)
    rec = recall_score(yte, y_pred, zero_division=0)
    f1 = f1_score(yte, y_pred, zero_division=0)
    cm = confusion_matrix(yte, y_pred)
    print(f"\n=== {name.upper()} - TEST METRICS ===")
    print(f"Akurasi : {akurasi:.4f} ({akurasi*100:.2f}%)")
    print(f"Presisi : {prec:.4f}")
    print(f"Recall : {rec:.4f}")
    print(f"F1 : {f1:.4f}")
    print("Confusion Matrix:\n", cm)
    print("\nClassification Report:\n",
          classification_report(yte, y_pred, digits=4))
    return akurasi, prec, rec, f1, cm, y_prob

def _sigmoid_np(z):
    z = np.clip(z, -500, 500)

```

```

        return 1, 0 / (1, 0 + np.exp(-z))

def _iterate_minibatches(n, batch_size, rng,
    shuffle=True):
    idx = np.arange(n)
    if shuffle:
        rng.shuffle(idx)
    for s in range(0, n, batch_size):
        e = min(s + batch_size, n)
        yield idx[s:e]

def run_Gauss-Newton_gr_minibatch(
    Xtr_s, ytr, Xva_s, yva, Xte_s, yte,
    epochs=200,
    batch_size=32,
    lr_epoch=5e-3,                # "base per-epoch" step
    size
    target_akurasi=0,75,          # satu-satunya
    kriteria berhenti
    damping=1e-6,                 # lambda untuk
    regularisasi JTJ
    shuffle=True,
    seed=42,
    verbose=True,
    log_val_every_batch=True,     # kalkulasi
    val_akurasi setiap batch (bisa dimatikan jika ingin
    lebih cepat)
    print_every=1                 # cetak tiap berapa
    batch (1=semua batch)
):
    """
        rng = np.random.default_rng(seed)

        # Tambah bias
        Xtr = np.hstack([np.ones((Xtr_s.shape[0], 1)),
Xtr_s]).astype(np.float64)
        Xva = np.hstack([np.ones((Xva_s.shape[0], 1)),
Xva_s]).astype(np.float64)
        Xte = np.hstack([np.ones((Xte_s.shape[0], 1)),
Xte_s]).astype(np.float64)

        ytr = ytr.reshape(-1, 1).astype(np.float64)
        yva = yva.reshape(-1, 1).astype(np.float64)
        yte = yte.reshape(-1, 1).astype(np.float64)

        N, d = Xtr.shape
        theta = np.zeros((d, 1), dtype=np.float64)

        # history per-epoch (seperti sebelumnya)

```

```

        hist_epoch = {"train_loss": [], "val_loss": [],
"train_akurasi": [], "val_akurasi": []}
        # history per-batch (baru)
        hist_batch = [] # list of dicts: {epoch, batch,
mb_size, train_batch_akurasi, val_akurasi_after
(optional)}
# TRAIN GAUSS-NEWTON+QR MINI-BATCH DENGAN TRACKING
WAKTU
print("\n🟢 Training GAUSS-NEWTON+QR Mini-Batch...")
print("="*100)

# Mulai timer
start_time = time.time()

theta_Gauss-Newton_mb, hist_Gauss-Newton_mb,
hist_Gauss-Newton_mb_batches, (akurasiG, pG, rG, f1G,
cmG, probG) = run_Gauss-Newton_qr_minibatch(
    X_train_s, y_train, X_val_s, y_val, X_test_s,
y_test,
    epochs=EPOCHS_GAUSS-NEWTON_MB, batch_size=GAUSS-
NEWTON_BATCH, lr_epoch=LR_GAUSS-NEWTON_EPOCH,
    target_akurasi=TARGET_AKURASI, damping=DAMPING,
shuffle=True, seed=SEED,
    verbose=True, log_val_every_batch=True,
print_every=1
)

# Hitung waktu total
end_time = time.time()
training_time_Gauss-Newton = end_time - start_time

# Tampilkan waktu pelatihan
print("\n" + "="*100)
print(f"🕒 GAUSS-NEWTON+QR TRAINING TIME:
{training_time_Gauss-Newton:.2f} seconds
({training_time_Gauss-Newton/60:.2f} minutes)")
print("="*100)

def plot_keras_hist(hist, title):
    fig, ax = plt.subplots(1,2, figsize=(12,4))
    ax[0].plot(hist.history['loss'], label='Train')
    ax[0].plot(hist.history['val_loss'], label='Val',
linestyle='--')
    ax[0].set_title(f'Loss - {title}')
    ax[0].set_xlabel('Epoch');
ax[0].set_ylabel('Binary Cross-Entropy')
    ax[0].grid(True, alpha=0.3); ax[0].legend()
    ax[1].plot(hist.history['akurasiuracy'],
label='Train')

```

```

        ax[1].plot(hist.history['val_akurasiuracy'],
label='Val', linestyle='--')
        ax[1].axhline(TARGET_AKURASI, color='gray',
linestyle=':', label=f'Target {TARGET_AKURASI}')
        ax[1].set_title(f'Akurasiuracy - {title}')
        ax[1].set_xlabel('Epoch');
ax[1].set_ylabel('Akurasiuracy');
ax[1].set_ylim([0,25,1,0])
        ax[1].grid(True, alpha=0,3); ax[1].legend()
        plt.tight_layout(); plt.show()

plot_keras_hist(hist_adam, "Adam")
plot_keras_hist(hist_sgd, "SGD")

# Konversi hist_Gauss-Newton_mb ke format yang benar
Gauss-Newton_history_for_plot = []
for i in range(len(hist_Gauss-
Newton_mb['train_akurasi'])):
    Gauss-Newton_history_for_plot.append({
        'iteration': i + 1,
        'train_akurasi': hist_Gauss-
Newton_mb['train_akurasi'][i],
        'val_akurasi': hist_Gauss-
Newton_mb['val_akurasi'][i],
        'train_loss': hist_Gauss-
Newton_mb['train_loss'][i],
        'val_loss': hist_Gauss-
Newton_mb['val_loss'][i]
    })

histories = {
    'Adam': hist_adam,
    'SGD': hist_sgd,
    'GAUSS-NEWTON+QR (Mini-Batch)': Gauss-
Newton_history_for_plot
}

# Plot gabungan (asumsikan hist_adam dan hist_sgd
sudah ada)
plot_combined_methods(iterations_hist, hist_adam,
hist_sgd, target_akurasi=TARGET_AKURASI)

```


RIWAYAT HIDUP



Alvin Aura Einst, lahir di Tuban pada 12 Januari 2002. Penulis merupakan putra dari pasangan Bapak Prayudianto dan Ibu Siswati. Pendidikan formal penulis dimulai di TK Muslimat NU 3, kemudian melanjutkan pendidikan dasar di SD Kutorejo 1 dan menyelesaikannya pada tahun 2015.

Pendidikan tingkat menengah pertama ditempuh di SMPN 1 Tuban dan lulus pada tahun 2018. Selanjutnya, penulis menyelesaikan pendidikan menengah atas di MA Tarbiyatul Banin Banat dan lulus pada tahun 2021. Pada tahun yang sama, penulis melanjutkan pendidikan tinggi di Universitas Islam Negeri Maulana Malik Ibrahim Malang dengan memilih Program Studi Matematika, Fakultas Sains dan Teknologi. Selama menempuh pendidikan di perguruan tinggi, penulis aktif berpartisipasi dalam berbagai kegiatan akademik maupun non-akademik. Penulis juga terlibat dalam kepanitiaan berbagai kegiatan kemahasiswaan yang memberikan pengalaman dalam bekerja sama, mengelola kegiatan, serta mengembangkan keterampilan organisasi. Selain itu, penulis aktif mengikuti seminar dan pelatihan, baik di dalam maupun di luar kampus, guna meningkatkan wawasan dan kompetensi di bidang akademik dan profesional.



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 : Alvin Aura Einst
NIM : 210601110012
Fakultas / Jurusan : Sains dan Teknologi / Matematika
Judul Skripsi : Optimasi Regresi Logistik menggunakan Gauss-Newton dan Dekomposisi QR untuk Penyelesaian Klasifikasi Biner
Pembimbing I : Dr. Mohammad Jamhuri, M.Si.
Pembimbing II : Dr. Ach. Nashichuddin, M.A.

No	Tanggal	Hal	Tanda Tangan
1.	11 Februari 2025	Konsultasi Bab I, II, dan III	1.
2.	12 Maret 2025	Konsultasi Bab I, II, dan III	2.
3.	9 April 2025	Konsultasi Bab I, II, dan III	3.
4.	11 April 2025	ACC Bab I, II, dan III	4.
5.	14 April 2025	Konsultasi Kajian Agama Bab I dan II	5.
6.	15 April 2025	Konsultasi Kajian Agama Bab I dan II	6.
7.	18 April 2025	Konsultasi Kajian Agama Bab I dan II	7.
8.	18 April 2025	ACC Kajian Agama Bab I dan II	8.
9.	16 Mei 2025	ACC Seminar Proposal	9.
10.	12 Agustus 2025	Konsultasi Revisi Seminar Proposal	10.
11.	28 Oktober 2025	Konsultasi Bab IV	11.
12.	31 Oktober 2025	Konsultasi Bab IV	12.
13.	4 November 2025	Konsultasi Bab IV dan V	13.
14.	20 November 2025	ACC Bab IV dan V	14.
15.	21 November 2025	Konsultasi Kajian Agama Bab IV	15.



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

16.	25 November 2025	Konsultasi Kajian Agama Bab IV	16.
17.	26 November 2025	ACC Kajian Agama Bab IV	17.
18.	3 Desember 2025	ACC Seminar Hasil	18.
19.	9 Desember 2025	Konsultasi Revisi Seminar Hasil	19.
20.	15 Desember 2025	Sidang Skripsi	20.
21.	22 Desember 2025	ACC Keseluruhan	21.

Malang, 22 Desember 2025

Mengetahui,

Ketua Program Studi Matematika



Rozi, M.Si.

NIP. 19800527 200801 1 012