OPTIMASI OPTIMAL BINNING MENGGUNAKAN ALGORITMA GENETIKA UNTUK MODEL PENILAIAN KREDIT

Rahmat Rian Hidayat*), Dwiki Jatikusumo[2], Yusuf Priyo Anggodo[3]
[1, 2] Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana, Jakarta
[3]Computer Sciense Departement, Binus Graduate Program Master of Computer Science, Bina Nusantara University, Jakarta
[1]Jln Meruya Selatan No 1 Kembangan, Jakarta Barat, DKI Jakarta, 11650, Indonesia
[3]Jln. Kebon Jeruk No 27, Jakarta Barat, DKI Jakarta, 11480, Indonesia
email: [1] rahmat.rian@mercubuana.ac.id, [2]dwiki.jatikusimo@mercubuana.ac.id, [3]yusuf.anggodo@binus.ac.id

Abstrak Credit scoring in financial institutions/banks is very important in determining whether it is feasible or not to receive borrowed funds. Therefore, credit scoring is the most important part of the company and has a great impact on the company's profits. Therefore, there are very strict regulations in the credit scoring process. This credit evaluation uses customer data collected by filling out forms, customer financial history data, and various other data sources. The Optimal Binning Algorithm is an industry-standard method used by data scientists to build credit scoring models. However, data scientists need to analyze each variable or feature individually to understand how the pattern of each variable affects the goal. When there are hundreds or even thousands of variables, it will take a lot of time, From a business perspective, it usually requires a solution that is as fast and effective as possible. To solve this problem, we propose the implementation of evolutionary algorithms to find the best solution of variable value constraints to form optimal binning. Genetic algorithms as a branch of *adaptive evolutionary* algorithms commonly used to solve value searches in optimization problems. Through this research, *genetic algorithms* can solve the problem of determining the limitations in the formation of *optimal binning ariables* very well. The results of the comparison between *the classifier* methods provide evidence that *Logistic Regression* can consistently have stable performance. In addition, it can maintain the performance of the predictive model despite changes in trends in *the test* data, while other methods get a decrease in performance results. In the *Logistics Regression t* test data, he received an AUC of 0.71.
Kata Kunci Credit Scoring, Optimal Binning, Genetic Algorithm, Data Scientist**.**

Abstrak- Credit scoring di lembaga keuangan/bank sangat penting dalam menentukan apakah layak atau tidak menerima dana pinjaman. Oleh karena itu, credit scoring merupakan bagian terpenting dari perusahaan dan berdampak besar terhadap keuntungan perusahaan. Oleh karena itu, ada peraturan yang sangat ketat dalam proses credit scoring. Evaluasi kredit ini menggunakan data nasabah yang dikumpulkan dengan mengisi formulir, data riwayat keuangan nasabah, dan berbagai sumber data lainnya. Algoritma Binning Optimal adalah metode standar industri yang digunakan oleh ilmuwan data untuk membangun model penilaian kredit. Namun, ilmuwan data perlu menganalisis setiap variabel atau fitur secara individual untuk memahami bagaimana pola setiap variabel mempengaruhi tujuan. Ketika ada ratusan bahkan ribuan variabel, itu akan memakan banyak waktu, Dari perspektif bisnis, biasanya membutuhkan solusi yang secepat dan seefektif mungkin. Untuk mengatasi masalah ini, kami mengusulkan implementasi algoritma evolusioner untuk menemukan solusi terbaik dari kendala nilai variabel untuk membentuk binning yang optimal. Algoritma genetika sebagai cabang dari *algoritma evolusi adaptif* yang biasa digunakan untuk memecahkan pencarian nilai dalam masalah optimasi. Melalui penelitian ini, *Algoritma Genetika* dapat memecahkan masalah dalam menentukan keterbatasan dalam pembentukan *binning variables yang optimal* dengan sangat baik. Hasil perbandingan antara metode *pengklasifikasi* memberikan bukti bahwa *Regresi Logistik* dapat secara konsisten memiliki kinerja yang stabil. Selain itu, dapat mempertahankan kinerja model prediksi meskipun ada perubahan tren dalam data *pengujian*, sementara metode lain mendapatkan penurunan hasil kinerja. Dalam data tes*t Regression Logistik*, ia menerima AUC sebesar 0,71.

Kata Kunci Kredit Skoring, Optimal Binning, Genetic Algorithm, Data Scientist**.**

*) **penulis korespondensi**: Rahmat Rian Hidayat
Email: rahmat.rian@mercubuana.ac.id

I. PENDAHULUAN

Penilaian kredit atau *credit scoring* adalah sistem atau cara yang dipakai oleh suatu lembaga pembiayaan/bank dalam menentukan penilaian layak atau tidaknya debitur dalam menerima pinjaman. Penilaian kredit ini berdasarkan beberapa data baik dari aplikasi nasabah maupun data histori finansial. Histori data ini berisi informasi tentang pinjaman dan pembayaran pinjaman sebelumnya sehingga data ini cukup berpengaruh terhadap *credit scoring* model. Ketika ada kesalahan dalam membuat model dan mengakibatkan prediksi yang tidak tepat ini berpotensi besar terjadi kerugian dalam perusahaan, sehingga *credit scoring* adalah bagian paling penting dalam perusahaan finansial.

Pengolahan data menggunakan *Optimal Binning* cukup populer digunakan di industri, pendekatan ini bertujuan untuk mengurangi *noise* atau *outlier* pada suatu data [1]. *Optimal Binning* adalah proses kategorisasi untuk merubah variabel kontinu menjadi kelompok atau *bins*. Setelah terbentuk *bins* dapat menghitung *information value* (*IV*) yang dijadikan metrik untuk melihat seberapa pengaruh variabel terhadap target[2]. Target atau label permasalahan ini ada dua kelas "Good"

dan "Bad" yang diambil berdasarkan *non performing loan* (NPL) yaitu pembayaran lewat dari 90 hari setelah tagihan. *Optimal Binning* sangat diandalkan dalam industri karena mudah menjelaskan bagaimana pengaruh setiap variabel terhadap hasil skor kredit debitur kepada *upper management* ataupun regulasi. Selain itu menurut [3] pendekatan ini dapat menjaga potensi pola dan distribusi variabel sehingga *classifier* dapat memprediksi dengan baik pada waktu depan.

Peningkatan yang sangat pesat dalam *financial technology* (Fintech) di Indonesia mendorong perkembangan *credit scoring* model juga, terlebih dalam eksplorasi data baru. Berdasarkan hal tersebut potensi data baru untuk membangun *credit scoring* akan sangat besar. Akan tetapi seorang data scientist atau risk scoring akan memerlukan waktu lama untuk mengeksplorasi tiap variabel untuk melihat bagaimana pengaruh terhadap target juga batasan paling terbaik dalam pengelompokan menggunakan *Optimal Binning*, terlebih ketika terdapat IIratusan hingga ribuan variabel.

Penelitian ini akan menggunakan publik dataset Fintech Indonesia. Data ini akan digunakan untuk simulasi pencarian variabel terbaik *Optimal Binning* menggunakan GAs. Hasil *Automatic Optimal Binning* akan dibandingkan dengan hasil manual untuk melihat apakah penggunaan GAs memberikan solusi mendekati hasil manual atau bahkan lebih baik. Metode *classifier* akan menggunakan *Logistics Regression* yang merupakan metode standar industri karena semua hal dapat dijelaskan dengan mudah dan setiap variabel yang digunakan dapat dikontrol [4]. Selain itu, juga akan membandingkan dengan metode *classifier* lain seperti *Random Forest* [5], *K-Nearest Neighbors* [6], *Xgboost* [7], *Naïve Bayes* [8], *Support Vector Machine* [9], *Classification and Regression Tree* [10], *Area Under Curve* (AUC) adalah metrik yang digunakan untuk membandingkan hasil prediksi dari tiap metode *classifier.*

Berdasarkan hal tersebut, peneliti akan mengusulkan penerapan algoritma genetika untuk mencari solusi terbaik untuk membentuk binning variabel terbaik dalam hal kredit skoring dengan judul" Optimasi Optimal Binning Menggunakan Algoritma Genetika untuk Penilaian Kredit". Penelitian ini hanya berfokus pada variabel numerik karena memiliki trend secara keseluruhan, sedangkan variabel kategoris masih memerlukan wawasan atau pengetahuan yang mendetail tentang variabel tersebut.

II. PENELITIAN YANG TERKAIT

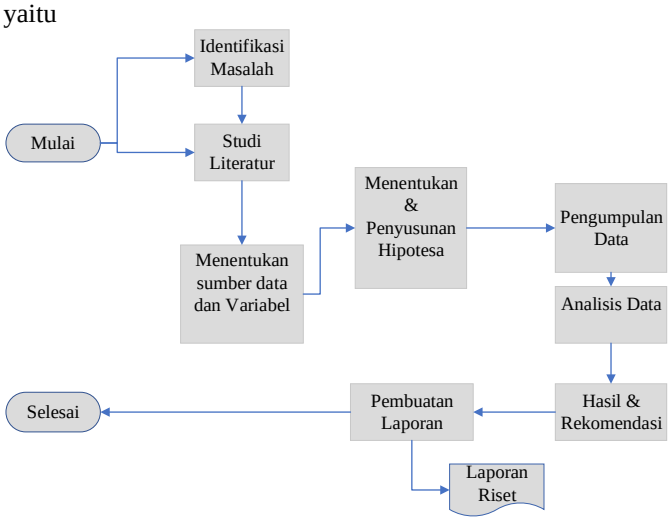
Penelitian [11] tentang penggunaan data *mobile device* dan *social network* untuk membagun model *credit scoring*. Selain itu, *credit scoring* telah sukses dibangun menggunakan data social media [8].

Pencarian solusi terbaik dalam pembentukan variabel *Optimal Binning* ini merupakan permasalahan *multi-objective*, yaitu mencari batasan terbaik berdasarkan beberapa kriteria dan terdapat metrik jelas untuk mengukur hasil terbaik. *Genetics Algorithm* (GAs) adalah salah satu metode paling populer dalam bidang pencarian solusi, GAs dapat diimplementasikan dalam banyak area baik itu penjadwalan, pengiriman, logistics, dan banyak industri. Dibandingkan metode lainnya GAs dapat mencari solusi yang sifatnya lebih global sehingga dalam jarak data yang panjang GAs jauh lebih unggul [12]. Pada bidang finansial GAs telah diterapkan dalam berbagai kasus. Penelitian [13] GAs dapat memberikan solusi terbaik dalam pemilihan variabel yang digunakan untuk membangun model *credit scoring* menggunakan *Neural Network*. Selain itu, GAs juga memberikan solusi pencarian lebih cepat dan paling optimum dalam membentuk struktur parameter *Neural Network* [14]. Sehingga pada penelitian ini akan fokus dalam implementasi GAs dalam membentuk variabel *Optimal Binning* dan penelitian difokuskan pada tipe variabel numerik karena memiliki makna nilai yang sama tanpa harus tau *insight* lebih seperti tipe data kategori.

III. METODE PENELITIAN

A. *Keterangan Gambar*

Tahapan penelitian yang digunakan penulis dalam riset ini yaitu



Gbr. 1. Tahapan penelitian

B. *Keterangan Tabel*

Langkah 1: Identifikasi masalah yaitu Untuk melakukan identifikasi masalah dilakukan interview, observasi kepada tim data scientist yang mengelola data skoring pada salah satu perbankan.
Langkah 2: Studi Literatur yaitu Mempelajari konsep, metode, teknik, maupun informasi dari berbagai sumber seperti internet, buku, jurnal, artikel ilmiah, ataupun referensi-referensi yang berkaitan dengan penelitian ini.
Langkah 3: Menentukan Sumber data & Variabel

ISSN: 2477-5126
e-ISSN: 2548-9356

Pada tahap ini menentukan sumber data skoring dengan menggunakan real dataset dan untuk simulasi penelitian dari salah satu fintech di Indonesia yaitu
DOI:10.17632/bsgrrzxsnp.1

Langkah 4: Menentukan & Penyusunan Hipotesis

Pada tahap ini menentukan hipotesis yang mana Hipotesa awal menggunakan genetic algorithms akan membuat pembentukan optimal binning akan lebih cepat dan hasilnya maksimal

Langkah 5: Pengumpulan Data

Untuk mengkategorikan data dibutuhkan berbagai macam tipe data yang dapat digunakan dalam pengolahan data untuk menghasilkan sebuah model yang digunakan untuk acuan pada tools yang digunakan.

Langkah 6: Analisis Data

Pada tahap ini di lakukan analisa data yang di dapat dengan menggunakan algoritma binning optimal dan Genetika.

Langkah 7: Hasil & Rekomendasi

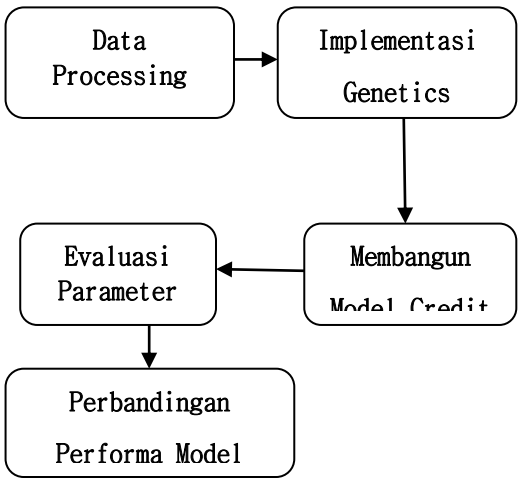
Hasil dari penelitian ini adalah model dari penggunaan algoritma Optimal Binning dan Genetika yang nantinya dapat memberikan rekomendasi untuk memberikan solusi dalam menentukan data skoring yang tepat dan efisien.

Langkah 8: Pembuatan Laporan Riset

Langkah terakhir dari riset ini adalah membuat laporan riset. Laporan ini berisi hal – hal yang dikerjakan selama riset dan hasil yang didapatkan pada saat melakukan riset. Dalam penulisannya, format yang digunakan adalah berdasarkan format yang telah diterapkan oleh Pusat Penelitian Universitas Mercu Buana.

IV. HASIL DAN PEMBAHASAN

Pada tahapan pembahasan ini melakukan analisa Data yang mana secara tahapan sebagai berikut:



Gbr. 2 : Tahapan Analisa Data

Pertama data preprocessing, yaitu membahas pendekatan yang digunakan untuk memproses data sebelum masuk tahapan feature engineering. Kedua implementasi GAs dalam melakukan optimasi pembentukan variabel optimal binning, didalamnya akan dibahas lebih detail untuk setiap proses dalam GAs. Ketiga terkait pengujian parameter GAs sehingga mendapatkan metode GAs dengan parameter terbaik dalam membentuk variabel optimal binning. Keempat membangun model credit scoring menggunakan variabel optimal binning dari hasil GAs. Terakhir membahas dan analisa terkait hasil evaluasi performa untuk setiap model credit scoring.

3.1 **Data *Preprocessing***

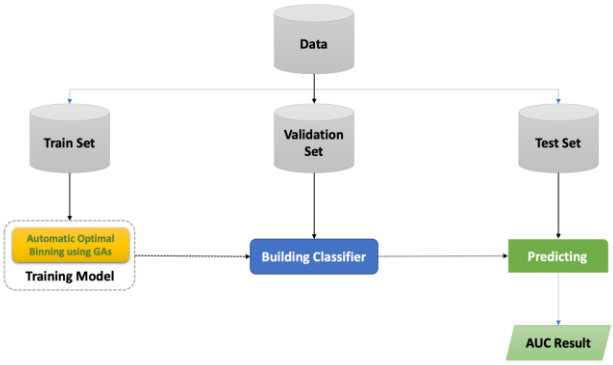
Penelitian ini menggunakan ril dataset salah fintech di Indonesia, dimana data dapat diakses di

Total data	82.492
Timeframe	Januari.2019 - April.2020
Total Variabel Numerik	356
Total Variabel Kategori	38
Definisi *Bad*	NPL
Persentase *Bad*	19.5 %

DOI:10.17632/bsgrrzxsnp.1.

Tabel 3.1. Informasi Dataset

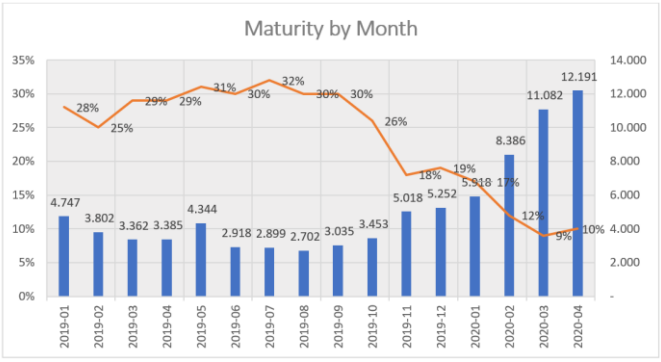
Sebelum masuk ke dalam proses pemodelan pertama adalah membagi data menjadi 3 bagian *train*, *validation*, dan *test.*



Gbr. 3. Pembagian Data Sampel

Masing-masing sampel memiliki fungsi yang penting dalam proses membangun model credit scoring. Sampel *train* digunakan untuk melatih model agar dapat memprediksi dengan baik dan presisi. Selanjutnya model akan divalidasi apakah model dapat bekerja dengan baik menggunakan

sampel *validation*. Kedua sampel diambil dalam waktu yang sama, sedangkan sampel *test* merupakan sampel yang diambil diluar *timeframe* sampel *test* maupun *validation* yang mana sampel *test* digunakan untuk simulasikan bahwa model credit scoring dapat bekerja dengan baik diwaktu yang akan datang.



Gbr 4. *Maturity* Dataset

Data mulai Januari 2020 sampai April 2020 akan digunakan sebagai data *test*. Sedangkan data tahun 2019 akan digunakan sebagai data *modeling* yang masing-masing diacak dengan proporsi 80 dan 20 persen untuk data *train* dan *test*.

Sebelum memasukan variabel pada proses optimal binning, perlu memastikan bahwa variabel-variabel tersebut cukup berkualitas. Ada beberapa penyaringan secara statistik untuk memastikan variabel yang tidak berkualitas telah dihapus, antara lain:

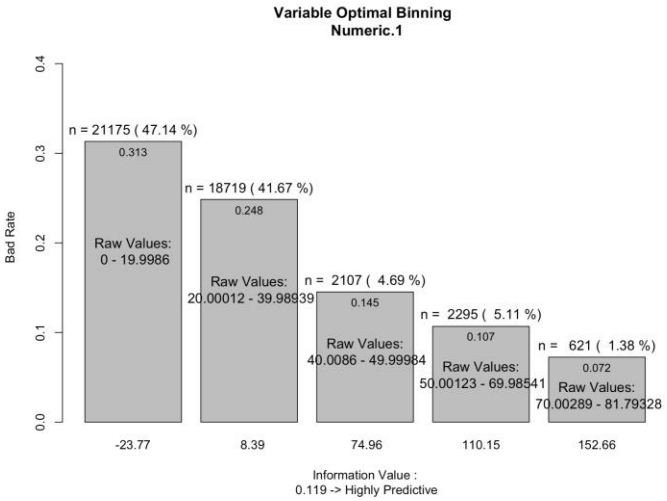
1. Menghapus variabel yang kosong semua atau yang nilainya sama semua, hal ini pasti tidak akan memiliki pengaruh apapun terhadap pemodelan. Dari 356 variabel terdapat 69 variabel nilai kosong semua dan 6 variabel nilainya sama semua. Sehingga terdapat 281 variabel yang dapat lanjut proses.
2. Menghitung *summary* statistik dari 281 variabel, dimana *summary* statistik ini akan digunakan mengetahui secara general bagaimana karakter tiap variabel. Informasi tersebut dapat mengetahui beberapa hal seperti tipe data, total data kosong, nilai paling sering muncul atau mode, total mode, dan juga persentase data kosong, nilai mode, dan persentase nilai diluar kosong dan mode. Informasi ini akan digunakan untuk filter variabel-variabel yang dirasa tidak akan berpengaruh besar terhadap pembentukan model, antara lain:
    - Variabel yang memiliki persentase kosong lebih dari 45%.
    - Variabel yang memiliki persentase mode lebih dari 90%.

    Filter ini membuat 124 variabel dihapus karena tidak akan berpengaruh besar terhadap modeling.

Dari proses ini memberikan hasil 158 variabel yang cukup berkualitas untuk masuk dalam proses pemodelan.

3.2 Implementasi Genetics Algorithm

Pada bagian ini telah selesai melakukan implementasi GAs untuk melakukan optimasi pada pembentukan variabel optimal binning. Sebelum membahas GAs pertama akan membahas bagaimana optimal binning bekerja dalam merubah sebuah variabel numerik menjadi variabel yang memiliki kelompok nilai. Gambar 7 menunjukan hasil variabel optimal binning pada *Numeric.1* dengan *cut-point* 20, 40, 50, dan 70.



Gbr 5. Variabel Optimal Binning *Numeric.1*

Hasil itu terbentuk 5 kelompok nilai yang mana linier menurun *bad-rate* nya untuk nilai semakin besar. Selain itu, variabel memiliki IV sebesar 0.119 yang artinya sangat bagus untuk pemodelan. Ada banyak potensi dalam membentuk optimal binning mulai jumlah kelompok nilainya dan juga *cut-point* yang digunakan. Intinya membentuk variabel yang pola kreditinya terlihat jelas dan memudahkan model untuk memprediksi. Pada *Numeric.1* terlihat semakin kecil nilainya maka semakin besar potensi untuk NPL. Hal lain yang perlu diperhatikan adalah batas bawah distribusi pada tiap bin yaitu 5% (Siddiqi, 2006).

GAs diimplementasikan untuk mencari *cut-point* terbaik untuk menghasilkan variabel optimal binning. Pada penelitian ini akan dibatasi jumlah *cut-point* mulai dari 1 sampai 10, artinya akan ada potensi kelompok mulai dari 2 sampai 11 kelompok.

3.2.1 Initialization chromosome

```
Initialization chromosome
While (stop condition){
      // reproduction step
      Crossover using one cut point
      Mutation using one cut point
      Calculate Fitness
```

Gbr 6. Prosedur GAs

Selain GAs memiliki beberapa tahapan, juga terdapat parameter yang digunakan untuk melakukan kontrol pada setiap tahapnya, antara lain:

- *Population size* adalah total solusi yang akan digunakan.
- *Generation* adalah nilai perulangan yang digunakan untuk kondisi GAs berhenti.
- *PC* adalah nilai untuk koefisien yang digunakan untuk mengatur jumlah *child* dalam tahapan *crossover*.
- *MC* adalah nilai untuk koefisien yang digunakan untuk mengatur jumlah *child* dalam tahapan *mutation*.

	Chromosome									
Type 1	10									
Type 2	36	85								
Type 3	34	43	43							
Type 4	9	41	44	74						
Type 5	27	41	65	68	90					
Type 6	24	41	57	60	65	76				
Type 7	4	7	44	54	55	56	66			
Type 8	4	22	25	51	68	73	85	88		
Type 9	6	23	25	38	43	59	88	95	98	
Type 10	1	8	38	42	58	70	71	83	93	98

Gbr 7. Chromosome GAs

Chromosome ini akan menggunakan binary chromosome dalam proses GAs, hal ini digunakan agar proses pencarian solusi lebih luas dibandingkan menggunakan *real number* pada proses reproduksi GAs.

3.2.2 Reproduksi

Reproduksi adalah tahapan untuk memproduksi *child* dari kombinasi 2 chromosome ataupun dari 1 chromosome. Proses ini adalah bagian penting dalam proses GAs dalam mencari potensi solusi yang lebih baik ataupun mencari *child* yang akan menghasilkan *child* baru yang lebih baik. Penelitian terdapat tipe chromosome yang berbeda-beda sehingga proses reproduksi hanya untuk satu tipe chromosome. Proses menghasilkan *child* chromosome tipe 1 hanya bisa dilakukan dari kombinasi chromosome tipe 1 saja, tidak dapat dilakukan kombinasi dengan tipe yang berbeda. Terdapat 2 tahapan yaitu *crossover* dan *mutation*. *Crossover* merupakan proses menghasilkan *child* dengan kombinasi 2 *chromosome parent* yang berbeda. Sedangkan *mutation* menghasilkan *child* dari satu *chromosome parent*. Masing-masing memiliki fokus dalam proses pencarian solusi, *crossover* lebih mencari solusi secara global dan *mutation* lebih fokus ke pencarian lokal area. Jumlah *child* yang dihasilkan akan sangat tergantung dengan nilai *pc* dan *mc*. Misalkan jumlah populasi sebesar 10, *pc* 0.4, dan *mc* 0.2. Maka jumlah *child* yang diproduksi pada *crossover* dan *mutation* masing-masing 8 dan 2 untuk setiap tipe chromosome.

- Proses *crossover*

  Katakan terdapat 10 chromosome tipe 1 dan diperlukan 8 *child* pada tahap ini. Pertama akan dibangkitakan secara acak 2 chromosome, misalkan memiliki nilai 20 dan 55. Terbentuk binary chromosome sebagai berikut.

P1		**1**	**1**	**1**	**1**	**0**	
P2		**1**	**1**	**0**	**1**	**1**	**1**

Metode yang digunakan adalah *one cut point* sehingga diperlukan 1 nilai secara acak antara 1 sampai 6 (total digit pada chromosome) yang digunakan untuk memproduksi *child*. Katakan 4, maka dapat membentuk *child* dari kombinasi *parent* pertama mulai dari digit 1 sampai *cut-point* yaitu 4 dan *parent* kedua dari digit 5 sampai terakhir.

					Cut-Point		
P1		**1**	**1**	**1**	**1**	**0**	
P2		**1**	**1**	**0**	**1**	**1**	**1**
C1		**1**	**1**	**1**	**1**	**1**	**1**

Begitu juga akan membentuk *child* kedua dari kombinasi *parent* 2 dan *parent* 1, sebagai berikut.

					Cut-Point		
P1		**1**	**1**	**1**	**1**	**0**	
P2		**1**	**1**	**0**	**1**	**1**	**1**
C2		**1**	**1**	**0**	**1**	**0**	

Sehingga dari proses ini didapat 2 *child* masing-masing memiliki nilai 63 dan 26. Proses akan diulangi mulai dari pemilihan secara acak *chromosome* sampai terpenuhi total *child* yang dibutuhkan.

Tahapan ini juga akan berlaku untuk tipe chromosome lain, dimana akan diproses setiap digitnya. Juga, dilakukan pengurutan nilai ulang pada chromosome lebih dari 1 untuk memastikan solusi yang diberikan sesuai dengan kebutuhan dari optimal binning.

- Proses *mutation*

Proses *mutation* ini diperlukan 2 *child*. Pertama akan dibangkitakan secara acak 1 chromosome, misalkan memiliki nilai 43. Terbentuk binary chromosome sebagai berikut.

P1		**1**	**0**	**1**	**0**	**1**	**1**

Pilih secara acak 1 angka mulai dari 1 sampai 6 (maksimal digit pada chromosome). Misalkan 4, maka merubah digit ke 4 menjadi kebalikannya jika 1 rubah ke 0 dan sebaliknya.

					Cut-Point		
P1		**1**	**0**	**1**	**0**	**1**	**1**
CI		**1**	**0**	**1**	**1**	**1**	**1**

Sehingga terbentuk *child* dengan nilai 47. Proses akan diulangi mulai dari pemilihan secara acak *chromosome* sampai terpenuhi total *child* yang dibutuhkan. Tahapan ini juga akan berlaku untuk

tipe chromosome lain, dimana akan diproses setiap digitnya. Juga, dilakukan pengurutan nilai ulang pada chromosome lebih dari 1 untuk memastikan solusi yang diberikan sesuai dengan kebutuhan dari optimal binning.

Crossover memproses kombinasi chromosome 20 dan 55 menghasilkan *child* dengan nilai cukup jauh yaitu 63 dan 26. Sedangkan *mutation* menggunakan *parent* dengan nilai 43 menghasilkan *child* yang nilainya tidak terlalu jauh yaitu 47. Hal ini sesuai dengan teori bahwa *crossover* lebih fokus pada pencarian global dan *mutation* lebih ke lokal. Kesimpulannya proses reproduksi ini menyeimbangkan proses pencarian global dan lokal untuk mencari solusi terbaik.

3.2.3 Fitness

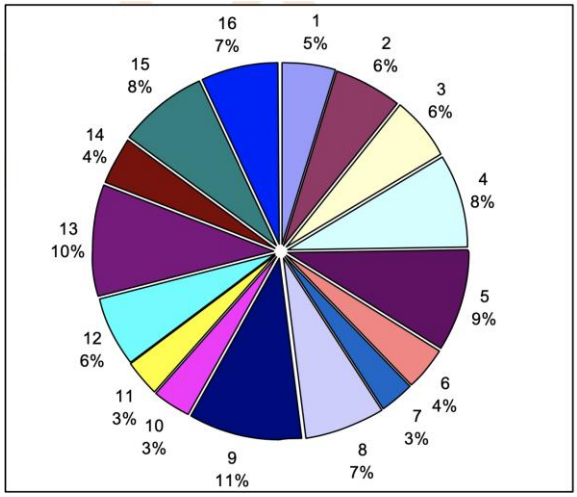
Tahapan ini berfokus pada proses menghitung seberapa bagus chromosome yang dihasilkan baik dari *parent* maupun *child*.

Fitness = Fit - Cost

3.2.4 Selection

Setelah didapatkan *child* maka akan ada chromosome yang jumlahnya lebih dari populasi. Sehingga akan diseleksi sebanyak populasi dikali 10 untuk semua tipe chromosome.

Menggunakan probabilitas kumulatif akan terbentuk sebuah *roulette wheel* yang akan digunakan untuk melakukan seleksi chromosome

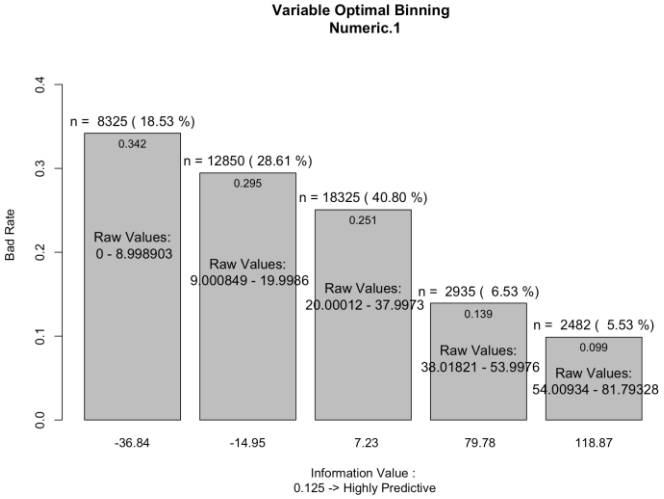


Gbr 8. Probabilitas Chromosome Terpilih (Mahmudy, 2013)

Hasil chromosome yang terseleksi akan berulang mulai dari reproduksi dan seterusnya sampai kondisi berhenti yaitu ketika perulangan sebanyak total generation. Di Akhir GAs akan memberikan solusi terbaik dari prosesnya. Menggunakan parameter GAs berikut:

- *Population size*: 10
- *Generation*: 5
- *PC*: 0.4
- *MC*: 0.2
- *Chromosome type*: All

pada variabel *Numeric.1* dihasilkan solusi *cut-point* optimal binning 9, 20, 38, dan 54.



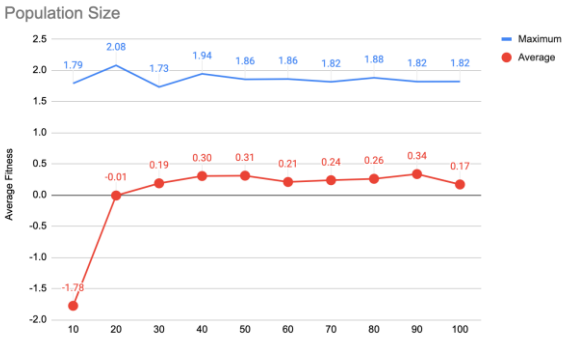
Gbr 9. Hasil Optimal Binning *Numeric.1*

3.3 Evaluasi Parameter Gas

Pada tahapan ini membahas lebih detail terkait evaluasi parameter pada GAs yang mana ini digunakan untuk mengetahui nilai parameter paling optimal dalam GAs. GAs termasuk dalam algoritma stokastik artinya terdapat beberapa variabel yang dibangkitkan secara acak. Oleh sebab itu, pengujian dilakukan sebanyak lima kali agar hasil yang didapatkan merupakan hasil rata-rata yang digunakan untuk mengambil parameter terbaik. Terdapat 4 parameter GAs yang dilakukan pengujian antar lain:

- *Population Size*: jumlah populasi yang digunakan dalam pencarian solusi.
- *Generation*: jumlah iterasi yang digunakan untuk pencarian solusi.
- *pc:* variabel kontrol dalam menentukan jumlah *child* ketika proses *crossover* direproduksi.
- *mc*: variabel kontrol dalam menentukan jumlah *child* ketika proses *mutation* direproduksi.

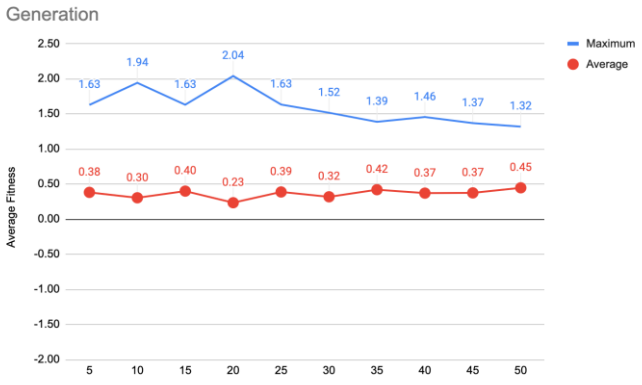
Pengujian dilakukan sebanyak lima kali dan diambil nilai rata-rata dari kelima hasil pengujian yang ditunjuk pada gambar berikut.



Gbr 10. Hasil Pengujian *Population Size*

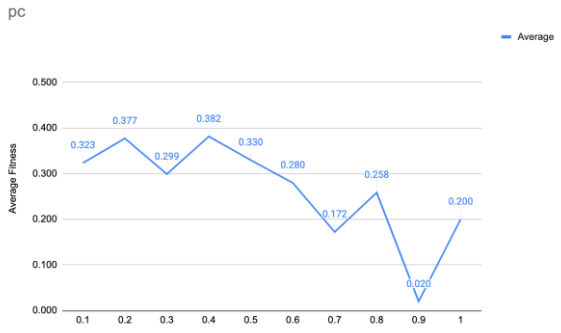
ISSN: 2477-5126
e-ISSN: 2548-9356

Pada nilai *population size* 40 sudah mengalami konvergensi dini sehingga pada jumlah yang lebih besar tidak terjadi peningkatan nilai *average fitness* yang cukup signifikan. Evaluasi ini memberikan bukti bahwa nilai 40 adalah nilai paling optimal untuk *population size*.
Setelah mendapatkan nilai *population size*, maka akan digunakan untuk pengujian selanjutnya yaitu *generation*



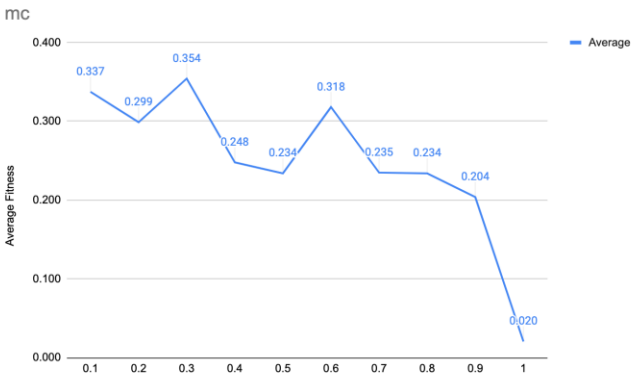
Gbr 11. Hasil Pengujian *Generation*

Average fitness terlihat tidak ada perubahan signifikan artinya dengan jumlah solusi awal yang cukup banyak tidak diperlukan iterasi yang banyak sudah dapat mencapai solusi mendekati optimum. Selain itu, pada jumlah iterasi mulai 25 sampai 50 terlihat terjadi tren penurunan pada *maximum fitness* artinya pencarian solusi terjebak pada permasalahan pencarian lokal area. Sehingga nilai 5 keluar sebagai paling optimal yang digunakan untuk *generation.* Karena nilai iterasi cukup kecil ini akan membuat proses pencarian solusi jauh lebih cepat.
Evaluasi telah memberikan keluaran nilai parameter paling optimal untuk *population size* dan *generation* masing-masing 40 dan 5. Proses selanjutnya adalah melakukan pengujian pada nilai kontrol GAs yaitu *pc* dan *mc*



Gbr 13. Hasil Pengujian *PC*

Nilai 0.4 memberikan hasil *average Fitness* tertinggi dibandingkan dengan nilai *pc* lainnya, artinya proses *crossover* memproduksi *child* sebesar 40% dari *parent* merupakan nilai terbaik untuk mendapatkan solusi yang optimal. Evaluasi terakhir adalah melakukan pengujian nilai kontrol pada *mc* menggunakan parameter GAs terbaik yang telah didapatkan yaitu *population size* 40, *generation* 5, dan *pc* sebesar 0.4.



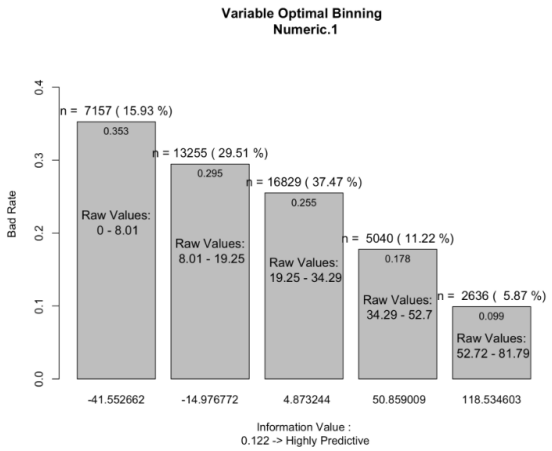
Gbr 12. Hasil Pengujian *MC*

Hasil evaluasi menunjukan nilai 0.3 keluar sebagai *average Fitness* terbesar dibandingkan nilai lainnya. Artinya, 30% produksi *child* dari *parent* pada proses *mutation* merupakan jalan terbaik untuk mendapatkan solusi paling optimum.
Hasil evaluasi ini telah mendapatkan nilai parameter terbaik dari GAs untuk pencarian solusi pada Optimal Binning variabel untuk membentuk *credit scoring* yang ditunjukan pada Tabel 3.2

Tabel 3.2. Parameter terbaik GAs

Parameter	**Nilai**
Population Size	40
Generation	5
pc	0.4
mc	0.3

Sehingga hasil pembentukan Optimal Binning menggunakan GAs dengan parameter terbaik yang telah dilakukan pengujian sebelumnya.



Gbr 14. Hasil Optimal Binning variabel *Numeric.1*

4. Membangun Model Credit Scoring

Setelah mendapatkan parameter terbaik selanjutnya adalah mencari solusi terbaik dari batasan pembentukan Optimal Binning menggunakan GAs untuk keseluruhan variabel.

Tahap *classification* akan diujikan menggunakan beberapa metode juga optimasi variabel terbaik antara lain *Logistics Regression* [4], *Random Forest* [5], *K-Nearest Neighbors* [6], *Xgboost* [7], *Naïve Bayes* [8], *Support Vector Machine* [9], *Classification and Regression Tree* [10].

Logistics Regression

Variable	Estimate	Std. Error	z value	Pr(>|z|)	VIF
(Intercept)	-1.039	9.747	-0.107	0.91508	0.00
Sc_Numeric.1	0.002	0.000	4.421	9.83E-06	1.27
Sc_Numeric.2	0.003	0.000	7.506	6.10E-14	1.95
Sc_Numeric.126	0.003	0.000	11.551	< 2e-16	2.17
Sc_Numeric.133	0.146	0.644	0.227	0.82024	1.00
Sc_Numeric.139	0.001	0.000	2.91	0.00362	1.80
Sc_Numeric.143	0.002	0.000	7.375	1.65E-13	1.92
Sc_Numeric.145	0.004	0.000	16.866	< 2e-16	1.22
Sc_Numeric.148	0.004	0.000	12.728	< 2e-16	1.64
Sc_Numeric.154	0.001	0.000	3.59	0.00033	1.20
Sc_Numeric.159	0.010	0.001	17.22	< 2e-16	1.03
Sc_Numeric.162	0.006	0.001	8.036	9.31E-16	1.33
Sc_Numeric.240	0.003	0.001	2.392	0.01676	2.02
Sc_Numeric.282	0.002	0.001	2.103	0.03543	1.97

Gbr 15. Hasil Pemodelan *Logistics Regression*

Dimana 32 variabel yang digunakan dalam model *Logistics Regression* setelah dilakukan proses seleksi menggunakan *Stepwise.*

Random Forest

35871 samples
32 predictor
2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 28697, 28697, 28696, 28697, 28697
Resampling results across tuning parameters:

mtry	Accuracy	Kappa
2	0.8104321	0.3942963
17	0.7999499	0.4168962
32	0.7902206	0.4023577

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 2.

Gbr 16. Hasil Pemodelan *Random Forest*

Random Forest dilakukan *resampling* sebanyak 5 kali untuk mendapatkan hasil model paling baik. Variabel yang digunakan sebanyak 32 untuk mengklasifikasi 2 kelas.

Berikut ini pemodelan menggunakan *K-Nearest Neighbors.*

K-Nearest Neighbours

Param	Length	Class	Mode
learn	2	-none-	list
k	1	-none-	numeric
theDots	0	-none-	list
xNames	32	-none-	character
problemType	1	-none-	character
tuneValue	1	data.frame	list
obsLevels	2	-none-	character
param	0	-none-	list
variable	All Variable		

Gbr 17. Hasil Pemodelan *K-Nearest Neighbors*

KNN melakukan klasifikasi dengan melihat kelas terbanyak dari titik disekitar data. Hasil pengujian menunjukkan KNN dapat bekerja dengan optimal ketika menggunakan 32 variabel yang masuk dalam proses modeling

Beikut ini pemodelan menggunakan *Xgboost.*

Xgboost

	Length	Class	Mode
handle	1	xgb.Booster.handle	externalptr
raw	278476	-none-	raw
niter	1	-none-	numeric
evaluation_log	2	data.table	list
call	17	-none-	call
params	5	-none-	list
callbacks	2	-none-	list
feature_names	32	-none-	character
nfeatures	1	-none-	numeric

Gbr 18. Hasil Pemodelan *Xgboost*

untuk mendapatkan hasil paling optimal baik pada data *train* maupun *validation* beberapa kali pengujian baik iterasi maupun variabel yang digunakan.

Berikut ini Metode klasifikasi yang melihat berdasarkan probabilitas data menggunakan *Naïve Baye.*

Naive Bayes

35871 samples
32 predictor
2 classes: '0', '1'

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 35871, 35871, 35871, 35871, 35871, 35871, ...
Resampling results across tuning parameters:

usekernel	Accuracy	Kappa
FALSE	0.774310	0.3143497
TRUE	0.758868	0.1381873

Tuning parameter 'laplace' was held constant at a value of 0
Tuning parameter 'adjust' was held constant at
a value of 1
Accuracy was used to select the optimal model using the largest value.
The final values used for the model were laplace = 0, usekernel = FALSE and adjust = 1.

Gbr 19. Hasil Pemodelan *Naïve Bayes*

Model dilakukan *bootstrapped* sebanyak 25 kali juga dilakukan tuning parameter untuk dapatkan hasil paling maksimal dari *Naïve Bayes*.

Berikut ini merupakan hasil pemodelan SVM.

Support Vector Machine
svm(formula = Target1 ~ ., data = dfSc_train, type = "C-classification", kernel = "linear")
Parameters:
SVM-Type: C-classification
SVM-Kernel: linear
cost: 1
Number of Support Vectors: 15826
(7039 8787)
Number of Classes: 2
Levels:
0 1

Gbr 20. Hasil Pemodelan *Support Vector Machine*

Berikut ini Metode *classifier* terakhir adalah *Classification and Regression Trees* yaitu metode regression menggunakan konsep dasar dari *tree*,

Classification and Regression Trees
35871 samples
32 predictor
2 classes: '0', '1'
No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 28697, 28697, 28697, 28697, 28696
Resampling results across tuning parameters:
cp Accuracy Kappa
0.01168969 0.8105434 0.3819165
0.03761955 0.8003120 0.3276552
0.19681190 0.7675554 0.1541558
Accuracy was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.01168969.

Gbr 21. Hasil Pemodelan *Classification and Regression Trees*

Setelah dilakukan pembangunan model, akan dilakukan pengujian performa dari masing-masing model baik dari sampel data *train*, *validation*, dan *test.* Dilakukan pengujian dengan melakukan perbandingan.

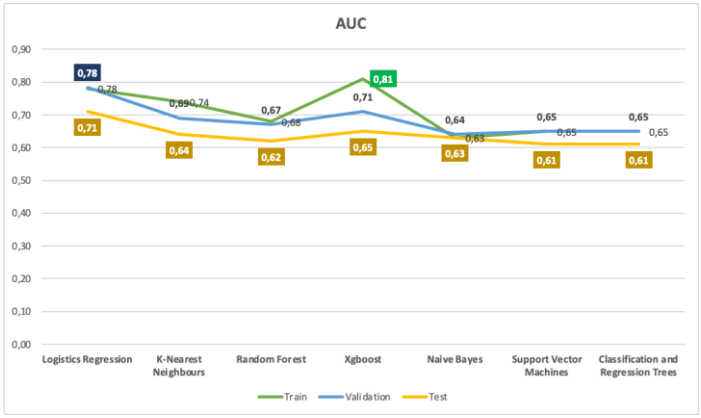
5 Perbandingan Performa Model Prediksi

Performa model dilihat dari hasil AUC, untuk melihat seberapa presisi model *classifier* melakukan prediksi baik untuk kelas peminjam *good* maupun *bad*.

Berikut ini Tabel dari perbandingan dari performa model prediksi.

Tabel 3.3. Perbandingan performa model Prediksi

No	Model Algoritma	Train	Validation	Test
1	Logistics Regression	0,78	0,78	0,71
2	K-Nearest Neighbours	0,74	0,69	0,64
3	Random Forest	0,67	0,68	0,62
4	Xgboost	0,81	0,71	0,65
5	Naïve Bayes	0,63	0,64	0,63
6	Support Vector Machines	0,65	0,65	0,61
7	Classification and Regression Trees	0,65	0,65	0,61



Gbr 22. Performa Model Prediksi

Hasil evaluasi dapat kita ketahui beberapa hal, pertama pada *train* sampel *Xgboost* memberikan hasil AUC tertinggi yaitu 0.81. Di Bawahnya ada *Logistics Regression* dengan AUC sebesar 0.78. Akan tetapi jika kita lihat hasil sampel *validation* juga maka terlihat penurunan performa model. *Xgboost* mengalami penurunan cukup signifikan sampai di angka 0.71. Sedangkan *Logistics Regression* memiliki performa yang sama dengan data *validation* artinya model berkerja dengan sangat. Pada metode lain tidak terjadi penurunan AUC, hanya KNN saja yang terjadi penurunan tetapi nilai sangat kecil. Artinya metode-metode yang memiliki performa stabil pada *train* dan *validation* sudah mencapai model mendekati optimal dan tidak terjadi *overfiting* maupung *underfitting*.

Data *test* adalah data yang tidak masuk sama sekali dalam proses modeling, data ini dipakai untuk simulasi bagaimana performa model diwaktu yang akan datang. Dilihat dari *Maturity by month*, data *test* terjadi perubahan tren peminjam. Tahun 2019 rata-rata bulanan *bad rate* sebesar 28% sedangkan mendekati akhir tahun dan menuju tahun 2020 terjadi penurunan *bad rate* sangat signifikan mencapai belasan persen. Sehingga secara normal model akan ada penurunan performa karena pola dari data terjadi perubahan. Keseluruhan metode mengalami hal ini, akan tetapi hasil evaluasi *Logistics Regression* masih dapat bertahan pada nilai yang cukup tinggi yaitu 0.71. Sedangkan *Xgboost* turun signifikan pada nilai AUC sebesar 0.65, artinya *Xgboost* tidak dapat memberikan hasil prediksi yang maksimal ketika ada perubahan tren data. Hal ini berbeda dengan *Logistics*

Regression yang masih dapat mempertahankan performa prediksi dengan cukup baik. Berdasarkan hasil evaluasi ini, penulis memberikan saran untuk menggunakan *Logistics Regression* karena terbukti dapat mempertahankan performa model prediksi walaupun terjadi perubahan tren data diwaktu yang akan datang.

KESIMPULAN

Genetics Algorithm dapat menyelesaikan permasalahan dalam menentukan batasan dalam pembentukan variabel *Optimal Binning* dengan sangat baik. *Chromosome* dibentuk menggunakan nilai biner dan terdapat 10 tipe untuk merepresentasikan masing-masing jumlah kelompok nilai dari *Optimal Binning* dapat memberikan hasil yang maksimal dalam proses pencarian solusi optimum untuk masing-masing variabel. Selain itu, perhitungan *Fitness* yang melihat dari beberapa variabel dapat memberikan nilai metrik yang tepat untuk sebuah solusi. Variabel yang dilihat dari perhitungan *Fitness* antara lain IV, jumlah bin, perbedaan nilai *bad-rate* bin pertama dan terakhir. Juga melihat aturan teori dalam pembentukan *Optimal Binning* yang baik seperti minimal distribusi bin, batasan nilai IV, nilai asli maupun *bad-rate* yang harus urut.

Pengujian parameter GAs didapatkan nilai parameter terbaik dengan susunan *Population size* sebesar 40, *Generation* sebesar 5, *PC* sebesar 0.4, dan *MC* sebesar 0.3. Menggunakan GAs dengan nilai parameter berikut memberikan hasil 32 variabel terbaik yang digunakan dalam proses pembentukan model prediksi. Hasil perbandingan antara metode *classifier* memberikan bukti bahwa *Logistics Regression* dapat konsisten memiliki performa yang stabil. Selain itu, dapat mempertahankan performa model prediksi walaupun terjadi perubahan tren pada data *test*, sedangkan metode lain mendapatkan hasil performa yang turun. Pada data *test Logistics Regression* mendapatkan AUC sebesar 0.71.

Sedangkan untuk saran dari penelitian ini adalah Validasi menggunakan data kredit scoring lain untuk pengujian GAs dalam pembentukan *Optimal Binning* menjadi saran penulis untuk penelitian kedepan. Hal ini, diperlukan untuk membuktikan bahwa *framework* yang dibuat memang dapat mencari solusi dengan baik. Selain itu, hibridisasi GAs dengan metode lain dapat dilakukan untuk mengatasi permasalahan terjebak dalam pencarian lokal area, sehingga dapat memberikan hasil lebih baik lagi.

UCAPAN TERIMA KASIH

Ucapan terima kasih kepada Biro Penelitian, Pengabdian Masyarakat dan Publikasi dengan perjanjian Nomor: 02-5/303/B-SPK/II/2022 telah memberikan dana untuk publikasi dalam pelaksanaan penilitian.

DAFTAR PUSTAKA

[1] P. Mironchyk. and V. Tchistiakov, "Monotone optimal binning algorithm for credit risk modeling.," *Utr. Work. Pap.*, 2017.

[2] G. Zeng, "Necessary condition for a good binning algorithm in credit scoring," *Appl. Math. Sci.*, vol. 65, pp. 3229–3242., 2014.

[3] J. P. Barddal, L. Loezer, F. Enembreck, and R. Lanzuolo, "Lessons Learned from Data Stream Classification Applied to Credit scoring," *Expert Syst. Appl.*, no. 113899, 2020.

[4] S. Lessmann, B. Baesens, H. V. Seow, and L. C. Thomas, "Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research," *Eur. J. Oper. Res.*, vol. 34(4), pp. 124–136, 2015.

[5] X. Zhang, Y. Yang, and Z. Zhou, "A Novel Credit scoring Model Based on Optimized Random Forest.," *IEEE 8th Annu. Comput. Commun. Work. Conf.*, pp. 60–65, 2018.

[6] L. Junior, F. Nardini, C. Renso, R. Trani, and J. Macedo, "A Novel Approach to Define the Local Region of Dynamic Selection Techniques in Imbalanced Credit scoring Problems.," *Expert Syst. Appl.*, vol. 113351, 2020.

[7] C. Wang, C. Deng, and S. Wang, "Imbalance-XGBoost: Leveraging Weighted and Focal Losses for Binary Label-Imbalanced Classification with XGBoost," *Pattern Recognit. Lett.*, pp. 190–197, 2020.

[8] S. V. Kulkarni and S. N. Dhage, "Advanced Credit Score Calculation using Social Media and Machine Learning.," *J. Intell. Fuzzy Syst.*, vol. 2373-2380., 2019.

[9] P. Pławiak, M. Abdar, and U. . Acharya, "Application of new deep genetic cascade ensemble of svm classifiers to predict the australian credit scoring.," *Appl. Soft Comput. 84*, vol. 105740, 2019.

[10] P. R. Hahn, J. S. Murray, and C. M. Carvalho, "Bayesian regression tree models for causal inference: Regularization, confounding, and heterogeneous effects (with discussion).," *Bayesian Anal. 15 (13)*, pp. 965–1056., 2020.

[11] M. B. Óskarsdóttir, C. Sarraute, J. Vanthienen, and B. Baesens, "The value of big data for credit scoring: Enhancing financial inclusion using mobile phone data and social network analytics.," *Appl. Soft Comput.*, vol. 74, pp. 26–39, 2019.

[12] R. Gen, M., Cheng, "Genetic algorithms and engineering optimization," *John Wiley Son*, vol. (Vol. 7, p. 12, 199AD.

[13] D. Ilter, E. Deniz, and O. Kocadagli, "Hybridized artificial neural network classifiers with a novel feature selection procedure based genetic algorithms and information complexity in credit scoring.," *Appl.*

Stoch. Model. Bus. Ind., vol. 37(2), pp. 203–228, 2021.

[14] H. R. Kazemi, K. Khalili-Damghani, and S. Sadi-Nezhad, "Tuning structural parameters of neural networks using genetic algorithm: A credit scoring application.," *Expert Syst.*, vol. 38(7), no. e12733. ., 2021.