

SYSTEMIC: Information System and Informatics Journal

ISSN: 2460-8092, 2548-6551 (e)

Vol 5 No 2 - Desember 2019

Klasifikasi Kelancaran Kredit Dengan Metode Random ForestMuhammad Irhamna Putra¹, Ahmad Yusuf², Nita Yalina³^{1,2,3} Universitas Islam Negeri Sunan Ampel, Surabaya, Indonesiairhamna.putraa@gmail.com¹ ahmadyusuf@uinsby.ac.id² nitayalina@uinsby.ac.id³**Kata Kunci***Data Mining, Random Forest, Sistem Pendukung Keputusan, Resiko Kredit***Abstrak**

Penelitian ini membahas tentang penggunaan machine learning untuk melakukan prediksi terhadap kelancaran kredit dengan menggunakan algoritma random forest. Prediksi ini akan menjadi sebuah tolok ukur bagi pihak bank untuk menindak lanjuti proses permohonan kredit oleh nasabah. Pada saat ini, belum adanya sistem pendukung keputusan guna melakukan prediksi terhadap kelancaran kredit menjadi sebuah permasalahan dalam upaya mengurangi resiko kredit pada bank. Untuk itu dikembangkannya sebuah sistem rekomendasi dengan pemodelan dari machine learning menggunakan algoritma random forest untuk melakukan prediksi kelancaran kredit. Berdasarkan hasil penelitian ini, model perhitungan prediksi yang telah melalui beberapa skenario training memiliki rata-rata accuracy sebesar 96,47%.

Keywords*Data Mining, Random Forest, Decision Support System, Credit Risk***Abstract**

This research contains the discussion the use of machine learning for doing prediction toward a good loan using random forest algorithm. This prediction will become basic reference for the bank to continue in evaluating credit risk. At this time, the absence of decision support system for doing prediction toward a good loan became a problem to the bank in attempt to reduce the credit risk. Therefore, a decision support system with machine learning modelling using random forest algorithm was built in predicting a good loan. Based on the result of this research, the prediction model being evaluated in several scenarios and having an average result 96,47% of accuracy.

1. Pendahuluan

Menurut UU No. 10 Tahun 1998 tanggal 10 November 1998 yang membahas tentang perbankan, jasa atau usaha perbankan, bank memiliki tiga kegiatan utama, yaitu menghimpun dana, menyalurkan dana serta memberikan jasa bank lainnya. Pada kegiatan bank lebih tepatnya menyalurkan dana, merupakan fokus utama bagi bank untuk melakukan pendistribusian kredit secara lebih luas. Hal ini dikarenakan laba yang diperoleh dari pendistribusian kredit ini merupakan asset utama dari bank. Dewasa ini, masih banyak perusahaan keuangan atau bank yang mengalami kesulitan dalam meminjamkan aset berupa kredit kepada nasabah yang terpercaya dan sesuai ketentuan [1].

Bank BRI Kantor cabang Pelaihari menjadi tempat studi kasus untuk penelitian ini. Dalam wawancara pribadi diketahui bahwa kantor cabang ini memiliki kesulitan dalam hal penilaian

terhadap nasabah yang akan mengajukan pinjaman. Untuk memberikan pinjaman, bank terlebih dahulu melakukan penilaian untuk menentukan kelancaran kredit seseorang. Penilaian ini dengan menggunakan skor kredit yang bertujuan untuk mengklasifikasikan nasabah, apakah nantinya kredit nasabah tersebut termasuk kategori *Good Loan* (memiliki kemungkinan kredit lancar) atau kategori *Bad Loan* (yang tidak memiliki kemungkinan kredit lancar). Untuk kebutuhan klasifikasi inilah, pihak bank membutuhkan sebuah sistem pendukung keputusan yang dapat mengklasifikasikan nasabah kedalam kategori yang sudah ada.

Ketika bank memiliki banyak pinjaman yang termasuk kategori *bad loan*, hal ini akan mempengaruhi rasio *Non Performing Loan* (NPL) bank tersebut. NPL adalah presentase dari perbandingan antara total kredit bermasalah terhadap total kredit yang disalurkan. NPL dapat difungsikan sebagai indikator resiko kredit, apabila tingkat rasio semakin rendah, maka semakin rendah pula tingkat kredit yang

bermasalah yang ditanggung oleh bank, begitu pula sebaliknya. Sehingga, rasio NPL ini digunakan untuk mengukur kemampuan bank dalam mengatasi kegagalan pengambilan kredit oleh debitur [2].

Sistem Pendukung Keputusan merupakan sistem berbasis komputer yang bertujuan untuk membantu dalam pengambilan keputusan dengan cara memanipulasi dan memanfaatkan data dengan bentuk model tertentu. Secara umum sistem pendukung keputusan cenderung berperan penting untuk memberikan bantuan yang efektif terhadap pengambilan keputusan [3].

Untuk kebutuhan model dalam pengambilan keputusan pada sistem pendukung keputusan, maka, berdasarkan penelitian – penelitian sebelumnya, algoritma *random forest* merupakan algoritma terbaik jika dibandingkan dengan algoritma – algoritma *machine learning* lainnya dalam hal memberikan rekomendasi [4][5][6]. Metode *random forest* pada saat ini telah banyak di implementasikan untuk melakukan klasifikasi terhadap berbagai permasalahan dalam berbagai situasi penelitian [7][8].

Pada penelitian terdahulu, Putri melakukan perbandingan terhadap beberapa algoritma guna klasifikasi yang ada pada *machine learning*. Penelitiannya memberikan hasil bahwa algoritma klasifikasi terbaik adalah *random forest* dengan menambahkan beberapa modifikasi pada algoritma tersebut [5].

Pada penelitian yang dilakukan oleh Budi Adnyana, untuk kebutuhan prediksi dengan menggunakan metode *random forest*. Hasil akhir menunjukkan nilai *accuracy* yang cukup baik yakni sebesar 83.54%. Sebuah penelitian tentang perbandingan metode klasifikasi juga mengatakan *random forest* masuk dalam urutan kedua jika diurutkan berdasarkan peringkat tinggi *accuracy*, [4][9].

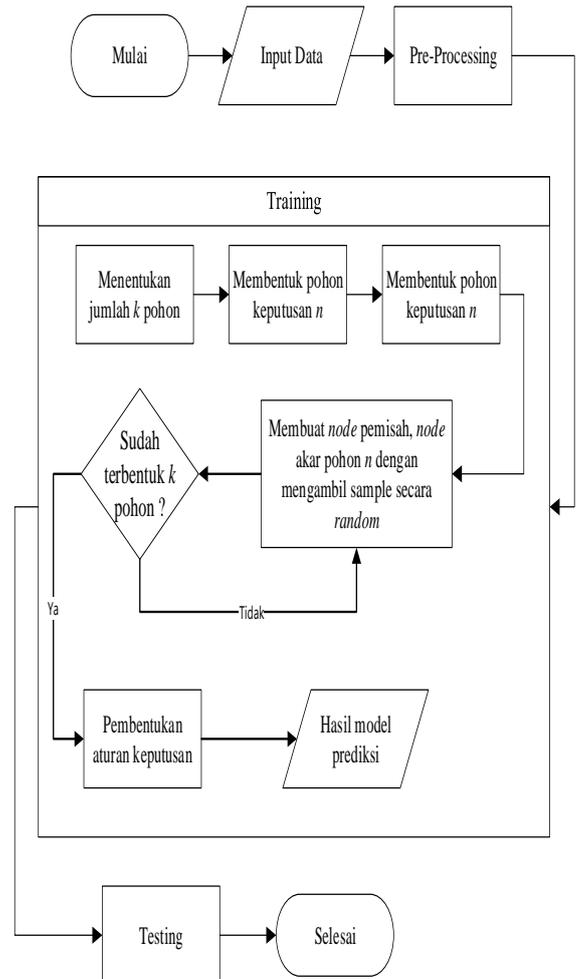
Berdasarkan paparan diatas, penelitian ini akan menggunakan algoritma *random forest* untuk memberikan rekomendasi terhadap prediksi kelancaran kredit pada bank BRI kantor cabang Pelaihari. Sehingga, dengan digunakannya *random forest* dalam prediksi kelancaran kredit tersebut diharapkan dapat mengetahui seberapa efektif metode tersebut.

2. Data Penelitian

Data yang digunakan dalam penelitian ini berupa 104 *row record data* tagihan kredit nasabah pada bank BRI kantor cabang Pelaihari pada periode tahun 2019. Tabel 1 menunjukkan variabel yang dimiliki oleh data yang diteliti.

3. Metodologi

Pada Gambar 1 menunjukkan tahapan – tahapan proses yang dilakukan pada penelitian ini.



Gambar 1. Metodologi penelitian

Tabel 1. Bentuk data yang diperoleh

Nama Variabel Data	Tipe Skala Data	Deskripsi
Usia	Rasio	Untuk kebutuhan klasifikasi perihal rentang usia
Alamat	Nominal	Untuk kebutuhan klasifikasi perihal daerah
Jenis Pekerjaan	Nominal	Untuk kebutuhan klasifikasi perihal jenis pekerjaan
Jumlah Kredit yang diajukan	Rasio	Untuk kebutuhan klasifikasi perihal rentang jumlah kredit yang diajukan
Jangka Waktu Kredit	Rasio	Untuk kebutuhan klasifikasi jangka waktu kredit
Status Pinjaman	Nominal	Untuk kebutuhan klasifikasi status pinjaman

Tabel 2. Bentuk tipe data sebelum dan sesudah *pre-processing*

Sebelum <i>Pre-processing</i>		Sesudah <i>Pre-processing</i>	
Nama Data	Tipe Data	Nama Data	Tipe data
Usia	Integer	Usia	Integer
Alamat	Text	Jumlah Pinjaman	Integer
Pekerjaan	Text	Jangka Waktu	Integer
Jumlah Pinjaman	Integer	Alamat_Labeled	Integer
Jangka Waktu	Integer	Pekerjaan_Labeled	Integer
Status Kredit	Text	Status Kredit	Integer

Dimulai dengan melakukan *input data* sebagai tahapan awal, data yang digunakan merupakan data *record* tagihan pinjaman yang telah diperoleh sebelumnya. Kemudian dilanjutkan dengan *pre-processing*, dimana data tersebut dipersiapkan sebelum masuk kedalam perhitungan. Selanjutnya masuk kedalam tahapan *training*, pada tahapan ini algoritma *random forest* sudah mulai dilakukan untuk menghasilkan sebuah *model*. Lalu tahapan terakhir yaitu *testing*, dimana dilakukannya pengujian terhadap *model* yang telah dihasilkan.

3.1 Pre-Processing

Metode yang dapat membantu dalam memperbaiki data untuk kebutuhan *machine learning* disebut sebagai *pre-processing*.

Label encoder adalah sebuah alternatif untuk penyandian variabel. *Label encoder* akan mengubah data yang bersifat kategorikal (dalam hal ini berbentuk *string*) dan menetapkan sebuah angka terhadap masing – masing *value*.

Tipe data awal ketika data pertama kali diperoleh ditunjukkan pada tabel 2, ada tiga kolom yang memiliki tipe data yang berbeda, yakni kolom 'alamat', 'pekerjaan' dan 'status kredit'. Dikarenakan *machine learning* tidak dapat memproses *multiclass / multi variable*, maka tiga kolom tersebut dikonversi menjadi *integer* pada proses *label encoder*, sehingga tipe data yang telah melalui proses *label encoder* disajikan pada tabel 2.

Pada tabel 3, dapat dilihat kolom yang telah diproses dari *label encoder* yang isinya awal berupa *string* dinotasikan menjadi *integer*. kolom 'alamat', berisikan nama – nama daerah yang dapat ditemui di Kecamatan Pelaihari, Kabupaten Tanah Laut, Provinsi Kalimantan Selatan. Nama – nama daerah ini merupakan daerah yang didalam pengawasan oleh Bank BRI dalam hal ini Kantor Cabang Pelaihari untuk kegiatan perbankannya. Setelah melakukan pengelompokan terhadap masing – masing daerahnya, kemudian dilakukan *label encoder* untuk menotasikan nama daerah tersebut menjadi sebuah *integer*.

Hal yang sama terjadi pada kolom 'pekerjaan' dan kolom 'status kredit', pada tabel 3 dapat dilihat pula perubahan notasi dari *string* menjadi *integer*.

Tabel 3 menjelaskan tentang perubahan nilai data pada variabel – variabel tertentu, yang kemudian di interpretasikan sebagai *integer*. Nilai – nilai data yang berbentuk *integer* ini lah yang disebut hasil *encoder*.

Tabel 3. Bentuk kolom 'alamat', 'pekerjaan', dan 'status kredit' setelah *label encoder*

Variabel	Nilai	Hasil Encoder
Alamat	Pelaihari Kota	'1'
	Banjarbaru	'2'
	Batu Ampar	'3'
	Takisung	'4'
	Bati - Bati	'5'
	Tambang Ulang	'6'
	Jorong	'7'
	Kintap	'8'
	Kurau	'9'
	Penyipatan	'10'
Pekerjaan	Karyawan BUMN	'1'
	PNS	'2'
	Wiraswasta	'3'
	Karyawan Swasta	'4'
	Guru	'5'
	Pensiunan PNS	'6'
	TNI AD	'7'
Status Kredit	Lancar	'0'
	Kurang Lancar	'1'

3.2 Training

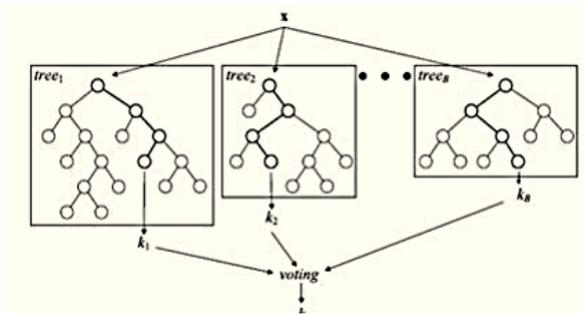
Pada tahapan ini, sudah mulai menggunakan algoritma *random forest* untuk penggunaan *machine learning*. Data yang telah melalui tahapan *pre-processing* selanjutnya masuk ke tahapan *training* yang akan menghasilkan model klasifikasi untuk kebutuhan prediksi. Gambar 2 menunjukkan gambaran umum tentang *random forest*.

Algoritma *random forest* bekerja dengan tahapan – tahapan sebagai berikut [9]:

1. Menentukan akan membangun berapa banyak pohon (n)
2. Melakukan pengambilan sampel data secara acak (*random*) dengan kemungkinan pengambilan sampel data yang sama (tahapan ini disebut tahapan *bootstrap*).
3. Gunakan salah satu dari contoh *Bootstrap tersebut* untuk melakukan perhitungan persamaan 1.

$$Gini=1- \sum_{i=1}^n (p_i)^2 \quad (1)$$

- Ulang kembali langkah 3 sebanyak n kali. Sehingga terbentuk hutan atas n pohon.



Gambar 2. Contoh random forest [8]

3.3 Testing

Tahapan ini dilakukan untuk melakukan pengujian terhadap model klasifikasi. Setelah dilakukan *training*, hal yang selanjutnya dilakukan adalah melakukan *testing* terhadap data tersebut. Tahapan *testing* memiliki beberapa skenario *training - testing* yang dijabarkan pada tabel 4.

Tahapan *testing* ini menggunakan *confusion matrix* untuk menguji algoritma *random forest*. *Confusion matrix* digunakan untuk melakukan pengukuran ketika melakukan analisis terhadap sebuah *classifier* [10].

		Predicted class		
		yes	no	Total
Actual class	yes	TP	FN	P
	no	FP	TN	N
Total		P'	N'	P + N

Gambar 3. Confusion Matrix [11]

Tabel 4. Skenario Training Testing

Skenario	Jumlah Dataset	Jumlah 'Lancar'	Jumlah 'Kurang Lancar'
60% Training - 100% Testing	61	59	2
70% Training - 100% Testing	72	69	3
80% Training - 100% Testing	80	76	4

Confusion Matrix dapat digunakan untuk melakukan evaluasi terhadap *accuracy*, *precision* dan *recall* [12]. Nilai *accuracy* menggambarkan seberapa akurat sistem dapat mengklasifikasikan data secara benar. Nilai *precision* menggambarkan jumlah data kategori positif yang diklasifikasikan secara benar dibagi dengan total data yang diklasifikasi positif. Sementara itu *recall* menunjukkan berapa persen data kategori positif

yang terklasifikasikan dengan benar oleh sistem. Adapun rumus dari *accuracy*, *precision* dan *recall* dapat dilihat pada persamaan 2, 3 dan 4:

$$Accuracy = \frac{TP+TN}{TP+TN+FN+FP} \quad (2)$$

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

4. Hasil

Bab ini menjelaskan hasil dari evaluasi yang dilakukan pada tahapan *testing* dengan menggunakan beberapa skenario *training-testing*. Evaluasi pada tahapan *testing* ini dengan menggunakan *confusion matrix* yang dapat dilihat hasilnya pada gambar 4.

CONFUSION MATRIX			
Training 60 - Testing 100	Prediksi \ Real	Lancar	Kurang Lancar
	Lancar	98	5
	Kurang Lancar	0	1
Training 70 - Testing 100	Prediksi \ Real	Lancar	Kurang Lancar
	Lancar	98	3
	Kurang Lancar	0	3
Training 80 - Testing 100	Prediksi \ Real	Lancar	Kurang Lancar
	Lancar	98	3
	Kurang Lancar	0	3

Gambar 4. Confusion Matrix [11]

Confusion matrix pada gambar 4 menjelaskan bahwa algoritma *random forest* mampu melakukan klasifikasi secara benar terhadap data yang bernilai 'lancar'. Dapat dilihat data yang bernilai *True Positive* pada masing - masing skenario menunjukkan angka 98. Sehingga pada tabel 5, didapati nilai *precision* 100% pada masing - masing skenario.

Pada *confusion matrix* tersebut dari masing - masing skenario dilakukan dua kali perhitungan dengan mengubah nilai 'lancar' dan 'kurang lancar' sebagai nilai *Positive*.

Pada tabel 5, merupakan hasil dari *accuracy* dengan menggunakan 'lancar' dan 'kurang lancar' sebagai nilai *Positive*. Masing - masing skenario menghasilkan *accuracy* yang memiliki rata - rata sebesar 96,47%.

Namun hal ini berbeda dengan nilai *precision* dan *recall*. Ketika menggunakan 'lancar' sebagai nilai *Positive*, hasil daripada *precision* cenderung meningkat dari angka 95,14% hingga 97,02% beriringan dengan jumlah data yang digunakan pada tahapan *training - testing*, dan nilai *recall* menunjukkan hasil yang stabil pada angka 100%.

Tabel 5. Hasil *accuracy*, *precision* dan *recall*

	Skenario 60%	Skenario 70%	Skenario 80%	Rata - Rata
<i>Accuracy</i>	95,19%	97,11%	97,11%	96,47%
True Positive = 'Lancar'				
<i>Precision</i>	95,14%	97,02%	97,02%	96,39%
<i>Recall</i>	100%	100%	100%	100%
True Positive = 'Kurang Lancar'				
<i>Precision</i>	100%	100%	100%	100%
<i>Recall</i>	16,67%	50%	50%	38,89%

Hal ini terjadi karena *precision* merupakan rasio prediksi benar positif yang dibandingkan dengan keseluruhan hasil yang diprediksi positif. *Precision* menjawab pertanyaan "Berapa persen nasabah yang benar 'lancar' dari keseluruhan nasabah yang diprediksi 'lancar'?" jika nilai *Positive* merupakan 'lancar' maupun sebaliknya.

Nilai *recall* yang cenderung rendah ketika 'kurang lancar' digunakan sebagai nilai *Positive* terjadi dikarenakan sedikitnya data *real* daripada 'kurang lancar' yang didapatkan. Hal ini menyebabkan rentang data yang cukup jauh antara data yang bernilai 'lancar' dan 'kurang lancar' pada tahapan proses *training*. Sehingga *machine learning* menghasilkan model klasifikasi yang dinilai kurang optimal untuk memprediksi nilai 'kurang lancar'.

Model klasifikasi untuk prediksi kelancaran kredit ini akan digunakan sebagai acuan dasar oleh pihak bank untuk proses tindak lanjut dari pengajuan kredit. Sehingga, setelah mendapatkan prediksi kelancaran kredit dari calon nasabah, bank dapat melanjutkan proses evaluasi yang lebih mendalam untuk meninjau kelancaran kredit nasabah tersebut secara lebih detail.

5. Kesimpulan dan Saran

Berdasarkan tiga skenario terhadap *training* dan *testing* pada algoritma *random forest*, didapatkan nilai *accuracy* dengan rata - rata sebesar 96,47%. Jika menggunakan nilai 'lancar' sebagai *True Positive*, didapatkan nilai *precision* sebesar 96,39% dan nilai *recall* sebesar 100%. Dengan hasil rata - rata nilai *accuracy* tersebut, algoritma *random forest* dinyatakan mampu melakukan prediksi terhadap kelancaran kredit yang sangat baik. Hal ini dapat dijadikan acuan dasar untuk proses tindak lanjut oleh bank.

Sedangkan jika menggunakan nilai 'kurang lancar' sebagai *True Positive*, didapatkan nilai *precision* sebesar 100% dan nilai *recall* sebesar 38,89%. Dengan nilai *recall* yang cenderung rendah ini, algoritma *random forest* dinyatakan belum dapat untuk melakukan prediksi terhadap kurang lancarnya kredit dari nasabah dengan baik. Namun hal ini terjadi dikarenakan rentang data antara data bernilai 'lancar' dan 'kurang lancar' yang cenderung jauh.

Sehingga, dapat disimpulkan bahwa algoritma

random forest mampu melakukan prediksi kelancaran kredit dengan baik. Sedangkan untuk melakukan prediksi terhadap kurang lancarnya kredit, dapat dilakukan modifikasi atau pengembangan lebih lanjut terhadap data yang akan digunakan.

Daftar Pustaka

- [1] K. Arun, G. Ishan, and K. Sanmeet, "Loan Approval Prediction based on Machine Learning Approach," *IOSR J. Comput. Eng.*, pp. 18–21, 2016.
- [2] M. Ali, *Asset Liability Management, Menyiasati Risiko Pasar dan Operasional dalam Perbankan*. Jakarta: PT. Elex Media Kompetindo Kelompok Gramedia, 2004.
- [3] P. Manurung, "Sistem Pendukung Keputusan Seleksi Penerima Beasiswa Dengan Metode AHP dan TOPSIS," Universitas Sumatera Utara, 2010.
- [4] C. Zhang, C. Liu, X. Zhang, and G. Almpandis, "An up-to-date comparison of state-of-the-art classification algorithms," *Expert Syst. Appl.*, vol. 82, pp. 128–150, 2017.
- [5] C. B. Putri, "Klasifikasi Nasabah Thera Bank Membeli Personal Loan Menggunakan Metode Klasifikasi Dalam Machine Learning Pendahuluan Metodologi Penelitian," Institut Teknologi Sepuluh Nopember, 2018.
- [6] T. K. Ho, "Random Decision Forest," vol. 47, pp. 4–5, 1995.
- [7] C. D. Sutton, "Classification and Regression Trees, Bagging, and Boosting," *Handb. Stat.*, vol. 24, no. 04, pp. 303–329, 2004.
- [8] M. van Wezel and R. Potharst, "Improved customer choice predictions using ensemble methods," *Eur. J. Oper. Res.*, vol. 181, no. 1, pp. 436–452, 2007.
- [9] I. M. Budi Adnyana, "Prediksi Lama Studi Mahasiswa Dengan Metode Random Forest (Studi Kasus: Stikom Bali)," *CSRID (Computer Sci. Res. Its Dev. Journal)*, vol. 8, no. 3, pp. 201–208, 2016.
- [10] J. Han, M. Kamber, and J. Pei, *Data Mining Concept and Techniques*. 2011.
- [11] R. Kohavi and F. Provost, "Glossary of Terms," *Mach. Learn.*, vol. 30, pp. 271–274,

- 1998.
- [12] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manag.*, vol. 45, no. 4, pp. 427–437, 2009.