

LOAN PREDICTION APP USING POLYNOMIAL REGRESSION

Arya Prawira¹, Catur Ariya²

^{1,2} Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara,
Jln. Letjen S. Parman No. 1, Jakarta, 11440, Indonesia
E-mail: ¹aryaprawira85@gmail.com, ²ariyacatur12@gmail.com

Abstrak.

Laporan ini mendeskripsikan sebuah algoritma untuk prediksi pinjaman hutang menggunakan Polynomial Regression untuk menghitung jumlah hutang yang dapat dipinjam oleh debitur. Keuntungan penggunaan algoritma ini adalah pemilihan degree yang dapat disesuaikan dengan bentuk penyebaran data sehingga mendapatkan akurasi yang maksimal. Kami menggunakan bahasa pemrograman python bersama dengan library scikit-learn, numpy, dan pandas untuk prediksi dan transformasi data serta seaborn dan matplotlib untuk visualisasi.

Kata kunci—Utang, Prediksi pinjaman hutang, Polynomial regression, Machine learning

Abstract

This paper describes an algorithm for loan prediction using Polynomial Regression to calculate the amount of debt that can be borrowed by debtors. The advantage of using this algorithm is that the degree selection can be adjusted to the shape of the data distribution so as to get maximum accuracy. We use python programming language along with scikit-learn, numpy, and pandas libraries for prediction and data transformation as well as seaborn and matplotlib for visualization.

Keywords: Debt, Debt loan prediction, Polynomial regression, Machine learning

1. PENDAHULUAN

1.1. Latar Belakang

Utang adalah sesuatu yang dipinjam seseorang baik itu berupa benda ataupun uang. Dengan berhutang seseorang mempunyai sejumlah uang yang dapat digunakan untuk memenuhi kebutuhannya ataupun membuka sebuah bisnis. Tentunya hutang perlu disesuaikan dengan pendapatan kita sehingga pada saat hutang jatuh tempo kita dapat membayarnya tepat waktu. Banyak orang yang belum mengetahui kemampuan pinjaman hutang mereka sehingga pada saat meminjam hutang pinjaman yang diajukan tidak sebanding dengan pendapatan mereka. Akibatnya hutang tersebut tidak dapat dibayar tepat waktu sehingga peminjam dapat dikenakan sanksi-sanksi yang semakin memberatkan peminjam dalam membayar hutangnya tersebut.

Selama ini, banyak aplikasi yang telah menggunakan prediksi pinjaman hutang dengan *machine learning*. Namun, pinjaman tersebut hanya dapat memprediksi status peminjam yang ada di bank. Aplikasi prediksi yang kami buat ini dapat digunakan oleh seluruh golongan masyarakat baik yang mempunyai rekening di bank maupun yang tidak mempunyai rekening di bank.

1.2. Tujuan Penelitian

Memprediksi Jumlah hutang yang dapat dipinjam oleh *user* dengan menggunakan metode *Polynomial Regression*.

1.3. Manfaat Penelitian.

Aplikasi ini dibuat dengan harapan untuk membuat *user* lebih berhati-hati dalam meminjam sebuah uang yang sesuai dengan status ekonomi mereka agar saat mengembalikan hutang tersebut tidak terjadi gagal bayar.

1.4. Studi Literatur Terhadap Penelitian Relevan

Setelah melakukan pencarian terhadap jurnal lain yang sesuai dengan topik proyek kami, kami menemukan beberapa jurnal yang menarik untuk dianalisis lebih lanjut. Jurnal pertama yang kami analisis berjudul "*Loan Payment Prediction Using Rough Sets*". Topik dari jurnal tersebut adalah memprediksi pembayaran hutang menggunakan algoritma *Rough Sets*. Tujuan dari peneliti mengambil topik tersebut karena peneliti melihat bahwa perusahaan kecil yang berhutang akan selalu ada resiko gagal bayar ataupun ketidakmampuan pengembalian hutang meskipun pemeriksaan yang dilakukan sudah sangat rinci. Peneliti ingin untuk mengurangi resiko terjadinya gagal bayar bukan mengurangi besaran pinjaman [1]. untuk mendukung penelitian ini peneliti menggunakan data yang diberikan oleh sebuah perusahaan. Data tersebut kemudian dibagi menjadi 2 group. Pada group pertama berisikan nama perusahaan yang ingin meminjam hutang beserta penjaminnya. Pada group kedua berisikan struktur finansial dari perusahaan yang meminjam hutang. Algoritma yang digunakan peneliti adalah algoritma *Rough Sets Theory* (RST). Algoritma RST memiliki pendekatan yang berbeda dengan algoritma yang lainnya dimana algoritma RST tidak membutuhkan anggota di dalam fungsinya sehingga algoritma ini bisa menghindari analisis pra-asumsi dan informasi satu sisi [1].

Pada akhir penelitian proyek tersebut, peneliti menemukan bahwa dengan menggunakan algoritma RST dapat mengklasifikasikan pinjaman hutang pada masing-masing perusahaan dengan sangat baik. Algoritma tersebut menyatakan bahwa mustahil ketika suatu perusahaan dengan penjualan yang buruk untuk melakukan pembayaran hutang sebaliknya jika penjualan perusahaan itu bagus maka sangat mungkin untuk perusahaan tersebut mengembalikan hutangnya.

Kami juga menemukan jurnal dengan penelitian yang cukup menarik untuk kami analisis dan teliti. Jurnal ini berjudul "*Loan Default Prediction using Machine Learning Techniques*". Dimana sesuai dengan judulnya, topik yang ingin diteliti adalah memprediksi apakah sebuah hutang layak dipinjamkan kepada seorang pelanggan atau tidak berdasarkan data finansial mereka [2].

Data yang digunakan pada penelitian tersebut adalah data yang diambil dari respon terhadap isu hutang individu dari tahun 2007 hingga tahun 2015. Metode algoritma yang digunakan dalam penelitian tersebut adalah algoritma *Logistic Regression* dan *Random Forest Classifier* yang kemudian dibandingkan untuk mendapatkan hasil yang terbaik. Peneliti mengklaim bahwa mereka berhasil membuat sebuah algoritma machine learning untuk memprediksi orang - orang yang sesuai dengan pinjaman hutang mereka dengan akurasi model algoritma *Logistic Regression* sebesar 99% dan algoritma *Random Forest* sebesar 78%.

Jurnal ketiga yang kami temukan mempunyai topik yang memiliki kemiripan dengan jurnal yang sublunar. Jurnal dengan judul "*Loan Approval Prediction Using Machine Learning*" berfokus kepada mengatasi ancaman terhadap keamanan kredit [3]. untuk mendukung penelitian ini peneliti telah mengumpulkan 1500 kasus dengan 8 karakteristik. Algoritma yang digunakan

oleh peneliti untuk melakukan analisis adalah algoritma SVM (*Support Vector Machine*) yang didukung oleh algoritma *feature extraction*. Hasil yang diperoleh oleh peneliti berupa prediksi terhadap *user* apakah pinjaman hutang diterima atau ditolak.

Selanjutnya kami juga menemukan sebuah jurnal berjudul “*Loan Prediction using Decision Tree and Random Forest*” yang memiliki kemiripan dengan penelitian yang akan kami lakukan. Pada proyek ini peneliti menggunakan algoritma *Decision Tree* dan *Random Forest* untuk memprediksi kredibilitas peminjam dalam mengembalikan hutang yang akan mereka pinjam [4]. Dataset yang digunakan sama dengan dataset yang kami gunakan saat ini yaitu *loan prediction dataset* yang diambil dari situs *kaggle.com*.

Alasan peneliti memilih untuk menggunakan algoritma *Decision Tree* dan *Random Forest* karena peneliti menganggap bahwa metode ini merupakan metode yang paling cocok dengan model data yang peneliti gunakan. Hasil yang didapatkan oleh peneliti berupa sebuah kelas target yang dapat mengklasifikasikan kredibilitas dari peminjam hutang dengan akurasi prediksi untuk algoritma *Decision Tree* sebesar 63,39% dan algoritma *Random Forest* sebesar 85,75%.

Jurnal terakhir yang kami teliti berjudul “*A Study on Machine Learning Algorithm for Enhancement Of Loan Prediction*” dimana topik yang dipilih oleh peneliti adalah melakukan prediksi terhadap seorang pelanggan apakah mereka dapat membayar kembali uang yang mereka pinjam atau tidak [5]. Data yang digunakan adalah dataset *loan prediction dataset* yang didapatkan dari situs *kaggle.com*. Metode yang digunakan adalah metode *supervised learning* dengan algoritma *Logistic Regression*, *Decision Tree*, dan *Random Forest*. Hasil yang diperoleh adalah sebuah akurasi prediksi dimana algoritma *Logistic Regression* merupakan algoritma yang paling baik digunakan untuk model ini dengan akurasi 89,7 %, *Random Forest* dengan akurasi 77,4% dan *Decision Tree* dengan akurasi 85,4%.

2. METODE PENELITIAN

2.1 Penjelasan Data.

Untuk mendukung proses prediksi, tentunya kami memerlukan data yang digunakan untuk melakukan pelatihan sistem. Data tersebut kami ambil di situ *kaggle* dengan nama *loan prediction dataset* (URL: <https://www.kaggle.com/altruistdelhite04/loan-prediction-problem-dataset>). Dataset tersebut terdiri atas beberapa kolom seperti yang terlihat pada tabel 1 :

Tabel 1. Kolom pada dataset

Nama	Tipe	Nama	Tipe
Loan Id	Object	ApplicantIncome	Int64
Gender	Object	CoapplicantIncome	Float64
Married	Object	Loan Amount	Float64
Dependents	Object	Loan Amount Term	Float64
Education	Object	Credit History	Object
Self Employed	Object	Property Area	Object
Loan Status	Object		

Setelah mengetahui kolom - kolom yang ada di dalam dataset (tabel 1), kami terlebih dahulu menentukan kelas target yang ingin diprediksi yaitu *loan_amount*. Setelah mengetahui kelas target yang ingin kami prediksi, kami kemudian menentukan fitur-fitur apa saja yang akan

kami gunakan dalam melakukan prediksi. Berikut adalah fitur-fitur yang kami pilih beserta tipe datanya sesuai dengan tabel 2.

Tabel 2. Kolom data yang terpilih

Nama	Tipe
Married	Object
Dependents	Object
Education	Object
ApplicantIncome	Int64
CoapplicantIncome	Float64

Penjelasan masing-masing fitur :

Married : Status pernikahan peminjam
 Dependents : Tanggungan dari peminjam
 Education : Menyatakan lulus atau tidaknya peminjam dari sekolah
 ApplicantIncome : Pemasukan pertama peminjam
 CoapplicantIncome : Pemasukan kedua peminjam.

Setelah menentukan fitur, kami melakukan pengecekan data hilang(*missing value*) dari fitur-fitur yang akan kami gunakan. Pengecekan ini bertujuan untuk mengetahui fitur mana yang harus diisi valuenya. Berikut adalah bentuk *code* yang telah kami lakukan seperti yang terlihat pada gambar 1 untuk pengecekan *missing value* pada *data train* dan gambar 2 untuk pengecekan *missing value* pada *data test*.

Melihat Missing data dari fitur-fitur yang akan digunakan baik itu yang berada di train dataset maupun test dataset

In [10]:

```
dftr.isnull().sum()
```

Out[10]:

```
Married          3
Dependents       15
Education         0
ApplicantIncome  0
CoapplicantIncome 0
LoanAmount       22
dtype: int64
```

Gambar 1. Missing data pada data train

In [11]:

```
dfte.isnull().sum()
```

Out[11]:

```
Married          0
Dependents       10
Education         0
ApplicantIncome  0
CoapplicantIncome 0
LoanAmount       5
dtype: int64
```

Gambar 2. Missing data pada data test

Setelah mengetahui *missing value* dari masing-masing fitur, pada data train fitur yang perlu dilakukan pengisian data hilang adalah *Married*, *Dependents*, dan *Loan Amount*. Sementara pada data test fitur yang perlu dilakukan pengisian data hilang adalah *Dependents* dan *Loan Amount*. Untuk data dengan tipe *object* seperti *Married* dan *Dependents* dilakukan dengan metode modus (*Mode*), sementara untuk tipe data *float* seperti *Loan Amount* dilakukan dengan metode rata-rata (*Mean*). Berikut adalah bentuk *code* yang telah kami lakukan seperti yang terlihat pada gambar 3.

Melakukan pengisian data kosong pada fitur loan amount menggunakan metode mean

In [12]:

```
dftr['LoanAmount'].fillna(dftr['LoanAmount'].mean(),inplace=True)
dfte['LoanAmount'].fillna(dftr['LoanAmount'].mean(),inplace=True)
```

Melakukan pengisian data kosong pada fitur married dan dependents menggunakan metode modus

In [13]:

```
dftr['Married'].fillna(dftr['Married'].mode()[0],inplace=True)
dftr['Dependents'].fillna(dftr['Dependents'].mode()[0],inplace=True)
dfte['Dependents'].fillna(dfte.Dependents.mode()[0],inplace=True)
```

Gambar 3. Coding Pengisian data

Untuk melakukan prediksi dengan metode regresi, kami terlebih dahulu melakukan transformasi tipe data dari bentuk objek ke bentuk integer. Ada 3 tipe data yang perlu kami ubah yaitu *married*, *dependents*, dan *education*. Pengubahan bentuk data dilakukan dengan metode *mapping*. Berikut adalah bentuk *coding* yang telah kami lakukan seperti yang terlihat pada gambar 4.

Melakukan transformasi data dengan metode mapping dari bentuk object menjadi bentuk integer

In [16]:

```
dftr.Married=dftr.Married.map({'No':0,'Yes':1}).astype(int)
dftr.Dependents = dftr.Dependents.map({'0':0,'1':1,'2':2,'3+':3}).astype(int)
dftr.Education = dftr.Education.map({'Graduate':1,'Not Graduate':0}).astype(int)
```

In [17]:

```
dfte.Married=dfte.Married.map({'No':0,'Yes':1}).astype(int)
dfte.Dependents = dfte.Dependents.map({'0':0,'1':1,'2':2,'3+':3}).astype(int)
dfte.Education = dfte.Education.map({'Graduate':1,'Not Graduate':0}).astype(int)
```

Gambar 4. Coding Transformasi Data

Setelah data terintegrasi, langkah selanjutnya adalah menentukan fitur dan target. Untuk fitur kami memutuskan untuk menggunakan data *Married*, *Dependents*, *Education*, *ApplicantIncome*, dan *CoapplicantIncome*. Berikut adalah bentuk *code* yang telah kami lakukan seperti yang terlihat pada gambar 5.

In [26]:

```
x_train=dftr.drop(["LoanAmount"],axis=1)
y_train=dftr['LoanAmount']
```

In [27]:

```
x_test=dfte.drop(["LoanAmount"],axis=1)
y_test=dfte['LoanAmount']
```

Gambar 5. Pemilihan Fitur dan Target

2.2 Pendekatan Algoritma

Dalam melakukan proyek prediksi pinjaman hutang ini kami membandingkan beberapa algoritma sebelum menentukan algoritma yang paling cocok digunakan untuk model data kami.

2.2.1 Algoritma Multiple Linear Regression

Untuk algoritma yang pertama kami menggunakan algoritma Regresi Linear Berganda (*Multiple Linear Regression*). Algoritma Regresi Linier berganda sering digunakan untuk meramalkan data dengan lebih dari satu variabel [6]. Algoritma Regresi Linier berganda dihitung dengan rumus persamaan (1)[5].

$$Y = a + b_1X_1 + b_2X_2 + b_3X_3 + \dots + b_nX_n + e \quad (1)$$

Dengan,

Y = Variabel *Dependent*

X_n = Variabel *Independen*

a = Konstanta

b_n = Konstanta *variable independent*

e = *error*

2.2.2 Algoritma Support Vector Machine

Untuk algoritma kedua kami menggunakan algoritma *Support Vector Machine*. Proses Pembelajaran SVM adalah menentukan *support vector*, kita harus mengetahui fungsi kernel yang dipakai, namun kita tidak perlu mengetahui wujud fungsi dari persamaan *non-linear*-nya. *Support Vector Machine* dihitung dengan rumus persamaan (2)[7].

$$f(x) = W^T \phi(x) + b \quad (2)$$

Dengan,

$W = (W_0, W_1, \dots, W_d)$ = vektor pembobot berdimensi 1.

$X = (x_1, x_2, \dots, x_d)$ = Variabel Input

$\phi(x)$ = fungsi transformasi fitur

b = bias.

Rumus diatas dapat digunakan Ketika data berbentuk *linear*. Namun Ketika data yang kita gunakan tidak berbentuk *linear* maka kita perlu menggunakan rumus persamaan (3) yaitu fungsi persamaan kernel [7].

$$K(x, y) = (x \cdot y + c)^d \quad (3)$$

Dengan,

(x, y) = Vektor Input

c = Variabel

d = Derajat *polynomial*

2.2.3 Algoritma Polynomial Regression

Untuk algoritma ketiga kami menggunakan algoritma regresi *polynomial*. *Polynomial Regression* dibentuk dari model Regresi Linear dengan menjumlahkan pengaruh masing-masing variabel predictor yang kemudian dipangkatkan sembari meningkatkan orde sampai orde ke- k [8]. *Polynomial Regression* Dihitung dengan menggunakan persamaan (4)[6].

$$Y = b_0 + b_1X + b_2X^2 + \dots + b_kX^k + \varepsilon \quad (4)$$

Dengan,

Y = Variabel Respons

b_0 = Intersep

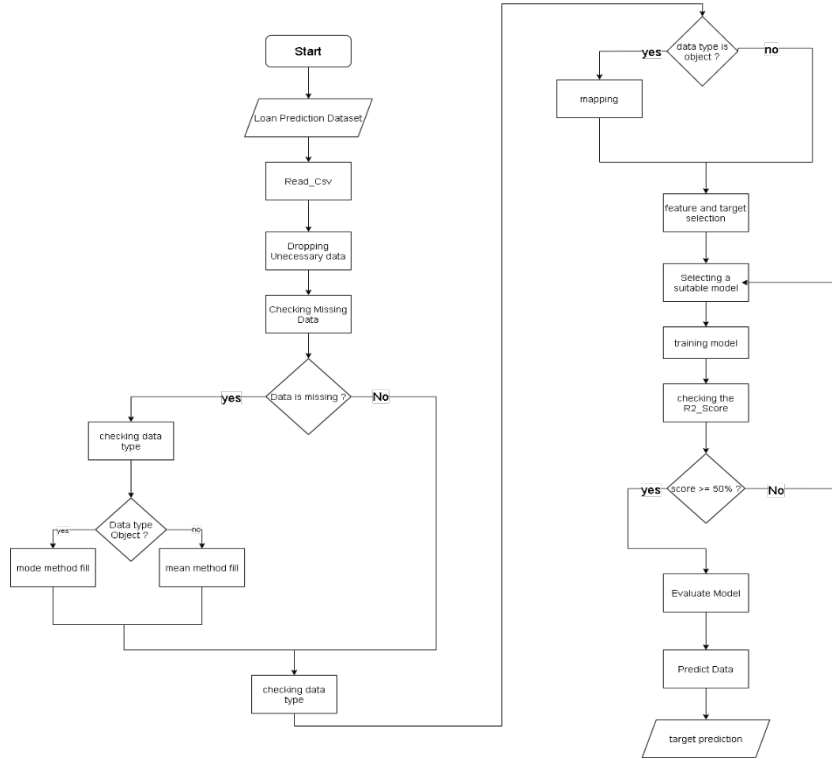
b_1, b_2, \dots, b_k = Koefisien – koefisien regresi

X = Variabel predictor

ε = Faktor pengganggu yang tidak dapat dijelaskan oleh model regresi.

2.3 Rancangan eksperimen

Dalam proses pembuatan aplikasi prediksi pinjaman hutang, kami membuat sebuah diagram alur (*FlowChart*) untuk melihat proses – proses yang akan dilakukan oleh sistem dalam melakukan prediksi. Proses ini meliputi bagian *input*, *process*, dan *output*. Berikut adalah bentuk Diagram alur (*Flow Chart*) yang telah kami buat seperti yang terlihat pada gambar 6.



Gambar 6. Diagram Alir Prediksi Pinjaman Hutang

2.4 Metode evaluasi

Dalam melakukan prediksi, kami melakukan evaluasi terhadap model – model algoritma yang kami gunakan. Berikut adalah metode evaluasi yang kami gunakan.

2.4.1 Metode R Square Score

Koefisien Determinasi atau yang biasa disebut dengan R^2 adalah sebuah metode analisis regresi yang bekerja dengan mengestimasi OLS. R^2 menjelaskan proporsi hubungan antara variasi variabel tak bebas (*dependent variable*) dengan variasi variabel bebas.

Rumus dari R^2 score dihitung dengan menggunakan persamaan (5)[8].

$$R^2 = 1 - \frac{RSS}{TSS} \quad (5)$$

Dimana RSS adalah *Residual Sum of Square* yang dihitung dengan persamaan (6)[8].

$$RSS = \sum_{i=1}^k V_i^2 \quad (6)$$

Dan TSS adalah *Total Sum of Squares* yang merupakan hasil penjumlahan antara RSS dan ESS (*Explained Sum Of Squares*). TSS dihitung dengan menggunakan persamaan (7)[8].

$$TSS = \sum_{i=1}^k (Y_i - \bar{Y})^2 \quad (7)$$

Dengan \bar{Y} adalah rata-rata dari Y_i

Melalui Persamaan (5) kita dapat menghitung besar kecilnya koefisien R^2 dari suatu model yang dipengaruhi oleh dua factor yaitu RSS dan sebaran data Y terhadap rata-ratanya. Semakin kecil RSS maka koefisien R^2 akan semakin besar begitu pula sebaliknya. Di sisi lain, RSS dipengaruhi oleh rata-rata sebaran Y . Semakin lebar sebaran rata-rata Y maka akan semakin besar pula nilai R begitu pula sebaliknya[9].

2.4.2 Metode ScatterPlot

Metode *ScatterPlot* atau penyebaran data grafik adalah sebuah metode yang mengambil dua buah variabel untuk kemudian di representasikan ke dalam bentuk grafik. Untuk membuat grafik ini, kita akan meletakkan variabel tak bebas (*dependent variabel*) di sumbu Y dan variabel bebas (*independent variabel*) ke sumbu X . *ScatterPlot* juga menyatakan hubungan yang terus menerus antara dua variabel yaitu variabel X dan variabel Y .

Sering kali *ScatterPlot* digunakan untuk menganalisis relasi yang terdapat di dalam dua variabel. *Scatterplot* juga dapat membantu para peneliti dalam menilai apakah sebuah metode cocok dengan suatu model statistik atau tidak. Misalnya saja, Jika sebuah hubungan antara dua variabel membentuk sebuah kesatuan linear maka kita dapat menggunakan model linear regression untuk menganalisis data tersebut. Pada python library yang digunakan untuk melakukan metode *scatterplot* adalah *seaborn* dan *matplotlib*.

3. HASIL DAN PEMBAHASAN

3.1. Hasil Penelitian

3.1.1. Hasil Evaluasi R Square

Dalam penelitian ini kami menggunakan 3 algoritma yaitu *Multiple Linear Regression*, *Support Vector Machine*, dan *Polynomial Regression*. Dari ke 3 algoritma tersebut kami lakukan pengujian terhadap model data kami untuk mengetahui algoritma mana yang paling cocok digunakan dalam melakukan prediksi. Pengujian kami lakukan dengan menggunakan metode R^2 Score. Berikut adalah hasil pengujian kami seperti yang terlihat pada tabel 3.

Tabel 3. Perbandingan hasil R^2 Score

No	Algoritma	R^2 Score
1	Multiple Linear Regression	40,35 %
2	Support Vector Machine	19,40 %
3	Polynomial Regression	61,57 %

Dari hasil perbandingan diatas, dapat dilihat bahwa metode yang paling cocok digunakan untuk memprediksi model kami adalah metode *Polynomial Regression* dengan akurasi tertinggi yaitu 61,57 %. Sementara itu metode *Multiple Linear Regression* dan *Support Vector Machine* memiliki akurasi yang lebih rendah yaitu 40,35% (*Multiple Linear Regression*) dan 19,40% (*Support Vector Machine*). Sehingga kami memutuskan untuk menggunakan metode *Polynomial Regression*.

Setelah melakukan perhitungan R^2 score untuk mengetahui metode yang paling cocok digunakan model kami, kami kemudian mencoba melakukan prediksi dengan memasukkan data baru yang sebelumnya tidak ada di dataset. Berikut adalah data yang kami masukkan seperti yang terlihat pada tabel 4.

Tabel 4. Variabel Predictor

Nama Fitur	Nama Variabel	Isi Variabel
ApplicantIncome	new_ApplicantIncome	6000
CoapplicantIncome	new_CoapplicantIncome	1000
Married	new_Married	1
Dependents	new_Dependents	2
Education	New_Education	1

Setelah mengetahui nilai dari variabel predictor kita terlebih dahulu harus mengetahui nilai intersep dan koefisien dari fitur yang kami gunakan. Berikut adalah nilai koefisien dan intercept dari masing-masing fitur seperti yang dapat dilihat pada tabel 5.

Tabel 5. Nilai Koefisien masing - masing fitur

Fitur	Nilai Koefisien	Nilai Intercept
Married	13,7575828	67.48898273718787
Dependents	5,52828412	
Education	15,490699	
ApplicantIncome	0.00785756192	
CoapplicantIncome	0.0069424816	

Setelah kami mengetahui nilai intersep dan koefisien dari fitur-fitur yang akan kami gunakan kami dapat menggunakan rumus persamaan *Polynomial Regression* (4) untuk melakukan prediksi dengan memasukkan nilai koefisien dan intersep tersebut ke dalam persamaan (8).

$$f(x) = y = 67,49 + 13,75(x) + 5,53(x)^2 + 15,49(x)^3 + 0.0079(x)^4 + 0.0069(x)^5 \quad (8)$$

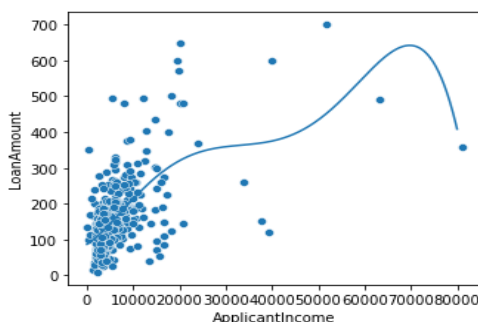
Langkah selanjutnya adalah memasukkan variabel predictor ke dalam persamaan (8).

$$f(x) = y = 67,49 + 13,75(1) + 5,53(2)^2 + 15,49(1)^3 + 0.0079(6000)^4 + 0.0069(1000)^5 \quad (9)$$

Dan kami mendapatkan hasil prediksi jumlah pinjaman hutang sebesar \$ 142,41.

3.1.2 Hasil Evaluasi *Scatter Plot*

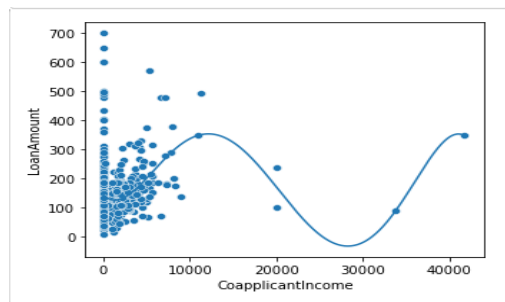
Untuk mengevaluasi model yang telah kami buat kami membuat 2 *plot* yang menyatakan korelasi fitur terhadap kelas target kami. Kedua fitur ini kami pilih karena tipe data yang digunakan bukanlah tipe data binary sehingga data dapat tersebar dengan baik. Metode yang kami gunakan adalah metode *Scatter Plot* dengan model *Polynomial Regression* ber-degree 5. Berikut adalah bentuk dari *scatter plot* beserta *line plot* dari fitur *ApplicantIncome* dan *Loan Amount* seperti yang terlihat pada gambar 7.



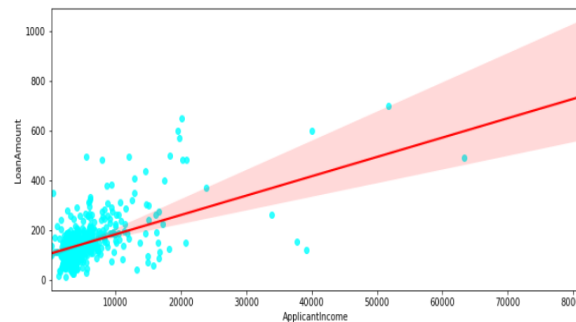
Gambar 7. Scatter Plot Fitur *Applicant Income* terhadap *Loan Amount*

Pada grafik *plot ApplicantIncome* terhadap *LoanAmount* seperti yang terlihat pada gambar 7, dapat dilihat bahwa garis model *Polynomial Regression* yang telah kami buat memiliki tingkat akurasi yang cukup baik dimana titik-titik yang tersebar banyak yang bersentuhan ataupun berdekatan dengan garis meskipun ada beberapa data yang jauh dari garis(*outlier data*).

Selanjutnya untuk grafik *CoapplicantIncome* terhadap *LoanAmount* dapat dilihat bahwa model *Polynomial Regression* memiliki tingkat efektifitas yang cukup tinggi karena banyak titik data yang menyentuh ataupun berdekatan dengan garis model. Berikut adalah bentuk dari *scatter plot* beserta *line plot* dari fitur *CoapplicantIncome* dan *Loan Amount* seperti yang terlihat pada gambar 8.

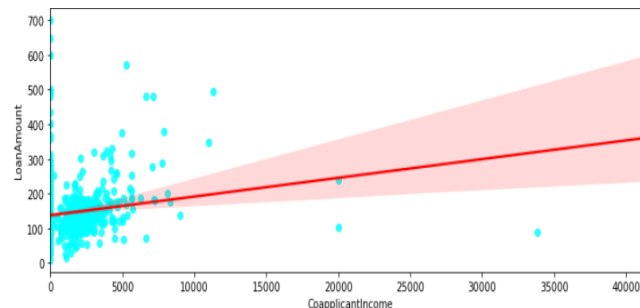


Gambar 8. Grafik Scatter Plot Fitur Coapplicant Income terhadap Loan Amount



Gambar 9. ScatterPlot Multiple Linear Regression Applicant Income

Kami juga melakukan evaluasi *ScatterPlot* terhadap model *Multiple Linear Regression* kami. Dapat dilihat pada gambar 9, model yang dihasilkan kurang baik karena titik data fitur *applicantIncome* berada jauh dari garis sehingga akurasi yang dihasilkan model kurang baik.

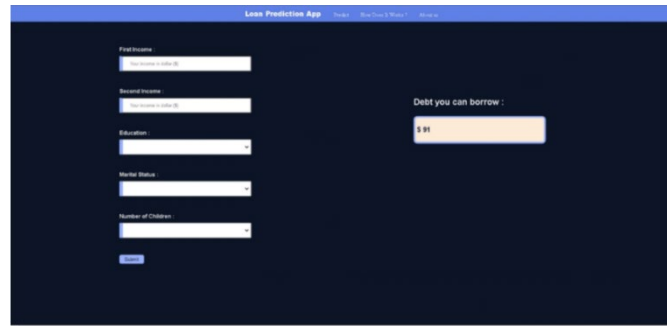


Gambar 10. ScatterPlot Multiple Linear Regression CoapplicantIncome

Dapat dilihat juga pada gambar 10, Penyebaran fitur *CoapplicantIncome* dengan menggunakan metode algoritma *Multiple Linear Regression* menghasilkan hasil yang kurang baik karena bentuk penyebaran data yang kurang sesuai dengan model algoritma.

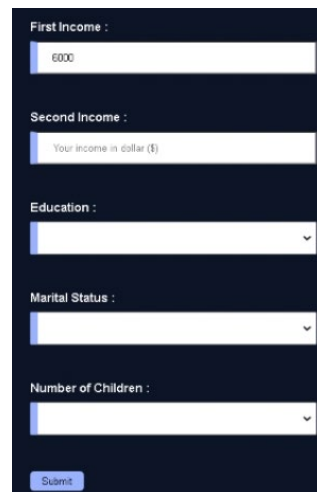
3.2. Pembuatan User Interface.

Berikut adalah bentuk keseluruhan web aplikasi yang telah kami kerjakan seperti yang terlihat pada gambar 9. Pembuatan aplikasi web ini bertujuan untuk mempermudah *user* dalam melakukan prediksi pinjaman hutang.

The image shows a web application interface titled "Loan Prediction App". On the left side, there are five input fields: "First Income" (with a placeholder "Your income in dollar (\$)"), "Second Income" (with a placeholder "Your income in dollar (\$)"), "Education" (a dropdown menu), "Marital Status" (a dropdown menu), and "Number of Children" (a dropdown menu). Below these fields is a blue "Submit" button. On the right side, there is a text label "Debt you can borrow :" followed by a yellow rectangular box containing the value "\$ 91".

Gambar 11 *Interface Aplikasi Prediksi Jumlah Pinjaman Hutang*

User kemudian mengisi *form* sesuai dengan instruksi yang diberikan. *First Income* seperti yang terlihat pada gambar 10 adalah pemasukan utama *user*.

This image is a close-up of the input form. The "First Income" field is filled with the number "6000". The "Second Income" field has a placeholder "Your income in dollar (\$)". The "Education", "Marital Status", and "Number of Children" fields are dropdown menus. A blue "Submit" button is at the bottom.

Gambar 12 *input First Income*

Second Income seperti yang terlihat pada gambar 11 adalah *form* untuk melakukan *input* pemasukan sekunder *user*.

This image is a close-up of the input form. The "Second Income" field is filled with the number "1000". The "First Income" field has a placeholder "Your income in dollar (\$)". The "Education", "Marital Status", and "Number of Children" fields are dropdown menus. A blue "Submit" button is at the bottom.

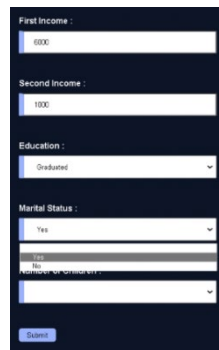
Gambar 13 *Input Second Income*

Education seperti yang terlihat pada gambar 12 merupakan sebuah pilihan yang menyatakan *user* telah lulus dari sekolah atau masih bersekolah.



Gambar 14 *Input Education*

Marital Status seperti yang terlihat pada gambar 13 adalah sebuah pilihan yang menyatakan status pernikahan *user* (sudah menikah atau belum).



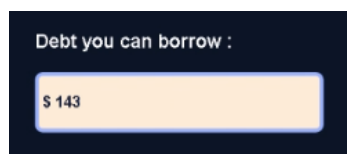
Gambar 15 *Input Marital Status*

Dan *form Number of Children* seperti yang terlihat seperti pada gambar 14 menyatakan jumlah anak atau tanggungan yang *user* punya.



Gambar 16 *Input Number of Children*

Lalu setelah menekan submit maka akan muncul tampilan pada bagian output seperti yang terlihat pada gambar 15.



Gambar 17 *Tampilan Output*

4. KESIMPULAN

Dari penelitian dan pengolahan data yang telah kami lakukan maka dapat ditarik kesimpulan bahwa Metode *Polynomial Regression* terbukti mempunyai kecocokan model yang lebih bagus terhadap dataset yang kami gunakan dibandingkan dengan metode *Multiple Linear Regression* dan *Support Vector Machine*. Hal ini dapat diketahui dengan menghitung R^2 Score dari masing-masing metode lalu membandingkannya. Metode *Polynomial Regression* mempunyai akurasi yang bagus terhadap model yang digunakan dikarenakan jumlah *degree* yang dapat diatur dan disesuaikan dengan bentuk penyebaran data sehingga data yang berdekatan dan bersentuhan dengan garis model menjadi lebih maksimal.

Penambahan fitur juga menjadi faktor penting dalam meningkatkan akurasi sebuah model. Semakin banyak fitur yang digunakan dalam suatu prediksi, maka semakin tinggi akurasi yang didapatkan oleh sistem prediksi tersebut dengan catatan memperhatikan fitur-fitur yang relevan terhadap target yang akan kita prediksi. Tidak dapat dipungkiri pada penelitian kami kali ini masih banyak kekurangan dan ruang untuk perkembangan lebih lanjut. Sangat banyak fitur yang masih dapat ditambahkan agar prediksi pinjaman dapat menjadi lebih akurat seperti penambahan fitur yang dapat menghitung jumlah pinjaman beserta bunganya ataupun prediksi berapa lama waktu bayar seseorang berdasarkan pendapatannya.

DAFTAR PUSTAKA

- [1] B. Ruzgar and N. S. Ruzgar, "Loan Payment Prediction Using Rough Sets," 2008.
- [2] Y. Diwate, P. Rana. and P. Chavan, "Loan Approval Prediction Using Machine Learning," vol. 9, no. 5, 2021.
- [3] K. Gautam., A. P. Singh, K. Tyagi and M. S. Kumar, "Loan Prediction using Decision Tree and Random Forest," vol. 7, no. 8, 2020.
- [4] D. Prateek, "A Study on Machine Learning Algorithm for Enhancement Of Loan Prediction," vol. 3, no. 1, 2021.
- [5] W. Anggraini and T. Irwinsyah, "Analisis Model Multiple Regression untuk Prediksi Nilai Kurs Rupiah Terhadap Dolar Amerika Berdasarkan Studi Makroekonomi," *Jurnal Teknik Industri*, vol. 3, no. 1, 2017.
- [6] V. Vikash and M. A. Ahmed., "Loan Default Prediction using Machine Learning Techniques," 2019.
- [7] I. C. R. Drajana, "Metode Support Vector Machine dan Forward Selection Prediksi Pembayaran Pembelian Bahan Bakukopra," *ILKOM Jurnal Ilmiah*, vol. 9, no. 2, p. 119, 2017.
- [8] J. S. Malensang, H. Komalig and D. Hatidja, "PENGEMBANGAN MODEL *POLYNOMIAL REGRESSION* BERGANDAPADA KASUS DATA PEMASARAN," *Jurnal Ilmiah Sains*, vol. 12, no. 2, p. 150, 2012.
- [9] Dr.R. Mohsen" Getting to Know The Scatter Plot"*Middle East Journal of Age and Ageing*, vol.6, no.2, 2009.