

ANALISIS PENGEMBANGAN MODEL PREDIKSI KESUKSESAN KICKSTARTER MENGGUNAKAN ALGORITMA BACKPROPAGATION DAN RANDOM FOREST

Indria Agustina¹, Yessi Mulyani², Trisya Septiana³, Mardiana⁴

^{1,2,3,4} Teknik Informatika, Universitas Lampung; Jl. Prof. Sumantri Brojonegoro No.1, Bandar Lampung, Lampung 35141, Indonesia.

Riwayat artikel:

Received: 8 Juli 2022

Accepted: 8 Agustus 2022

Published: 15 Agustus 2022

Keywords:

Kickstarter

Crowdfunding

Prediksi

Backpropagation Artificial Neural Network

Random forest.

Correspondent Email:

indriaagustina17@gmail.com

How to cite this article:

Agustina, Indria. (2022). Analisis Pengembangan Model Prediksi Kesuksesan Kickstarter Menggunakan Algoritma Backpropagation Artificial Neural Network (ANN) dan Random forest. *Jurnal Informatika dan Teknik Elektro Terapan*, 10(3).

© 2022 JITET (Jurnal Informatika dan Teknik Elektro Terapan). This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY NC)

Abstrak. Model prediksi ini dibangun dengan pendekatan klasifikasi pada data mining dengan algoritma *backpropagation* ANN dan *random forest*. Kategori tingkat kesuksesan dibagi menjadi dua yaitu tidak sukses, dan sukses. Selanjutnya model prediksi dikembangkan berdasarkan variabel-variabel informasi yang ada, dimana model dikembangkan dalam dua tahap. Tahap pertama, model prediksi dikembangkan dengan 10 variabel (9 variabel sebagai *input* dan 1 variabel sebagai *label*) yang ada pada dataset. Kemudian pada tahap kedua hanya 8 variabel (7 variabel sebagai *input* dan 1 variabel sebagai *label*), dikurangi variabel-variabel tentang *pledge* dan *backers* pada pengembangan model prediksi. Pengembangan model prediksi ini menggunakan 5723 data *campaign* kategori teknologi dari portal Kickstarter. Model pertama menggunakan algoritma *backpropagation* ANN dan *random forest* memberikan akurasi tertinggi masing-masing sebesar (89%, 98%) Sedangkan model kedua memberikan akurasi tertinggi masing-masing sebesar (69%, 65.7%) Hasil penelitian tersebut menunjukkan bahwa pengurangan variabel *pledge* dan *backers* dapat menurunkan performa model prediksi. Berkaitan dengan pemanfaatan, 2 model prediksi tahap pertama dan kedua adalah sebagai perbandingan. Dan dari data yang sudah didapat dapat disimpulkan bahwa model yang akan dipakai untuk memprediksi kesuksesan *kickstarter* adalah model pertama dengan menggunakan algoritma *random forest* yang merupakan model terbaik dengan nilai akurasi 98% dan nilai *f1-score* nya 98%.

Abstract. This prediction model was built using a classification approach to data mining with the ANN *backpropagation* algorithm and *random forest*. The success rate category is divided into two, namely *unsuccessful* and *successful*. Furthermore, the prediction model was developed based on the available information variables, where the model was developed in two stages. The first stage, the prediction model is developed with 10 variables (9 variables as *input* and 1 variable as *label*) that exist in the dataset. Then in the second stage there are only 8 variables (7 variables as *input* and 1 variable as *label*), minus the variables regarding *pledges* and *backers* in the development of the prediction model. The development of this prediction model uses 5723 technology category *campaign* data from the Kickstarter portal. The first model using the ANN *backpropagation* algorithm and *random forest* gives the highest accuracy of (89%, 98%) respectively. While the second model provides the highest accuracy of each (69%, 65.7%). The results of this study indicate that the reduction in the *pledge* and *backers* variables can reduce the performance of the predictive model. In terms of utilization, the first and second stage 2 prediction models are for comparison. And from the data that has been obtained, it can be concluded that the model that will be used to predict the success of the *kickstarter* is the first model using the *random forest* algorithm which is the best model with an accuracy value of 98% and an *f1-score* value of 98%.

1. PENDAHULUAN

Perkembangan teknologi berperan penting dalam mendorong semua sektor, termasuk sektor keuangan. Penggunaan teknologi dibidang keuangan sering disebut dengan istilah fintech atau *financial technology*. Munculnya teknologi keuangan (*financial technology*) akan membawa perubahan besar di sektor keuangan, seperti model bisnis keuangan, dan perangkat lunak yang terkait dengan keuangan dan komunikasi[1].

Diawali dengan sulitnya menggalang dana untuk proyek bisnis komunitas, kegiatan sosial, usaha mikro dan *start-up*. Dalam beberapa tahun terakhir, munculah sebuah *platform* yang disebut penggalangan dana atau *Crowdfunding*. Situs ini mempertemukan pemilik dana (*funder*) dengan pemilik ide atau gagasan yang membutuhkan dana (*fundraiser*)[2]. Pada dasarnya, istilah tersebut mengacu pada metode pembiayaan bersama oleh individu dan kelompok masyarakat (pendukung), yang biasanya dilakukan pada *platform crowdfunding* untuk proyek-proyek di bawah aspek kemanusiaan, sosial, atau bisnis.

Semakin populer sebuah *platform*, maka semakin banyak celah yang dapat dimanfaatkan oknum tidak bertanggung jawab untuk menyalahgunakan uang yang diberikan pendukung (juga dikenal sebagai investor/pendukung/donor) [3].

Ada beberapa *platform Crowdfunding* raksasa, seperti Indiegogo, Kickstarter, dan GoFundMe yang tidak luput dari penipuan. Seperti dilansir The Verge (29/8), FTC melakukan survei atau investigasi sebuah *start-up* dibalik produk iBackPack yang didanai oleh dua raksasa *platform crowdfunding*, Indiegogo dan Kickstarter, jumlahnya sekitar \$800.000 kemudian menghilang tanpa menepati janjinya pada pendukung yang sudah menginvestasikan uang [4].

iBackPack sendiri adalah ransel yang menjanjikan teknologi inovatif untuk memenuhi berbagai kebutuhan pengguna laptop, seperti klaim kompartemen antipeluru dan rahasia. *Start-up* dibalik produk ini berkomitmen untuk mempromosikan ide bisnis dan menawarkan produk kepada pendukung melalui akun nama pengguna Monahan, namun pada saat itu *website* dan *email* iBackPack sedang *down* dan Monahan tidak dapat dihubungi, sehingga tidak ada satupun

supporter yang menerima produk yang dijanjikan.

Oleh sebab itu, selain dilakukan kewaspadaan perlu adanya suatu alat yang bisa memprediksi sebuah proyek kickstarter, model yang bisa memprediksi sebuah proyek tersebut itu sukses atau tidak, dengan model *machine learning* yang dibuat dengan menggunakan 2 algoritma yang sering digunakan dalam membuat model prediksi yaitu algoritma *backpropagation artificial neural network* dan *random forest*. algoritma *backpropagation* disebut algoritma yang bisa memperkecil nilai *error* atau *loss* sehingga membuat akurasi model lebih tinggi dan bisa memprediksi lebih akurat. Sedangkan algoritma *Random forest* yang bisa memproses data *training* dalam jumlah banyak secara efisien, dapat menghasilkan nilai *error* yang lebih rendah karena merupakan algoritma *ensemble learning* serta dapat memberikan hasil yang bagus dalam klasifikasi prediksi kickstarter yang diharapkan bisa membuat model yang akurat dalam memprediksi kesuksesan sebuah proyek kickstarter.

2. TINJAUAN PUSTAKA

2.1 Crowdfunding

Crowdfunding berasal dari istilah *Crowdsourcing*, yaitu proses untuk mendapatkan *source* (sumber daya) dari *crowd* (kerumunan/orang banyak). Menurut howe menyatakan bahwa kerumunan (*crowds*) akan lebih efisien dibandingkan individu. Dalam *crowdfunding* sumber daya yang dicari dari luar adalah sumber daya berupa pendanaan. Menurut Lambert & Schwiendbacher, menyatakan bahwa *Crowdfunding* adalah cara mendanai keuangan sebuah proyek atau perusahaan melalui internet[5].

2.2 Kickstarter

Kickstarter adalah *platform* pendanaan yang memungkinkan pengembang untuk terhubung dan membangkitkan minat pada proyek kreatif tertentu yang ingin dimulai. *Platform* ini sepenuhnya didukung oleh *crowdfunding* yang berarti masyarakat umum selaku *funder* dapat membantu merealisasikan proyek atau ide dari *fundraiser* sehingga proyek atau ide tersebut bisa diproduksi atau direalisasikan.[6]

2.2 Artificial Neural Network

Artificial Neural Network (ANN) atau Jaringan Syaraf Tiruan (JST) merupakan set algoritma yang berkerja seperti jaringan syaraf otak manusia, dimana *neuron* saling terhubung satu dengan lainnya, bekerja untuk memproses informasi[7]. *Backpropagation* (Propagasi balik) adalah salah satu dari jaringan saraf tiruan (*Neural Network*) merupakan metode pelatihan yang terawasi (*Supervised Learning*) dengan jaringan *multilayer* dan memiliki ciri khusus meminimalkan *error* pada output yang dihasilkan oleh jaringan. klasifikasi ini berkerja dengan cara melakukan dua tahap perhitungan yaitu perhitungan maju yang akan menghitung nilai kesalahan (*error*) antara nilai *output* sistem dengan nilai yang seharusnya dan perhitungan mundur untuk memperbaiki bobot berdasarkan nilai *error* tersebut[8].

2.4 Stochastic Gradient Descent

Gradient descent yang umumnya digunakan pada model *machine learning* adalah *stochastic gradient descent*. *Stochastic gradient descent* adalah proses pembelajaran yang melakukan update untuk setiap 1 data. Hal ini berarti, setiap melakukan 1 kali *forward-propagation* dengan 1 data, maka akan terjadi 1 kali *back-propagation* juga.[9]

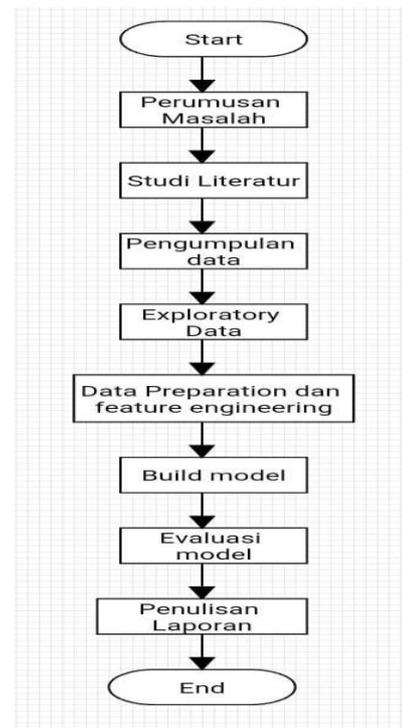
2.5 Random forest

Random forest (RF) adalah suatu algoritma yang digunakan pada klasifikasi data dalam jumlah yang besar. Proses klasifikasi pada *random forest* berawal dari memecah data sampel yang ada kedalam *decision tree* secara acak. Setelah pohon terbentuk, maka akan dilakukan *voting* pada setiap kelas dari data sampel. Kemudian, mengkombinasikan *vote* dari setiap kelas lalu diambil *vote* yang paling banyak. Dengan menggunakan *random forest* pada klasifikasi data maka, akan menghasilkan *vote* yang paling baik[10].

2.6 Kaggle

Menurut InfoWorld, Kaggle adalah sebuah komunitas online yang dibentuk oleh Anthony Goldbloom sebagai CEO dan Ben Hamner sebagai CTO di tahun 2010. Komunitas *online* ini menampung para pegiat *data science* yang ingin belajar lebih dalam tentang *machine learning* dan ilmu-ilmu terkait lainnya.[11]

3 METODE PENELITIAN



Gambar 3.1 Flowchart Tahapan Penelitian

Berikut ini tahapan-tahapan yang dilakukan pada penelitian sebagaimana dibahas diatas tentang *flowchart* penelitian.

3.1 Perumusan Masalah

Adapun rumusan masalah pada penelitian ini yaitu tentang bagaimana membuat model yang bisa membantu para *fundraiser* dan *funder* dalam program penggalangan dana dengan melakukan prediksi apakah proyek kickstarter yang akan dikerjakan bisa sukses atau tidak.

3.2 Studi Literature

Pada tahapan ini mengumpulkan literasi serta teori seperti jurnal dan materi terkait penelitian ini.

3.3 Pengumpulan Data

Tahapan ini dilakukan dengan mengumpulkan data yang terkait dengan program kickstarter. Dan dalam penelitian ini dataset yang digunakan merupakan data yang didapat dari *platform* kaggle yang dirilis oleh

pengamat kickstarter. Dataset yang digunakan ini memiliki 15 variabel yang diantaranya adalah sebagai berikut:

Table 3.1 Variabel dataset

No	Nama Variabel	Deskripsi
1.	ID	ID dari masing-masing proyek kickstarter
2.	Name	Nama dari sebuah proyek
3.	Category	Kategori yang lebih detail
4.	Main_category	Kategori untuk kampanyenya secara umum
5.	Currency	Mata uang yang digunakan untuk memberikan dukungan finansial
6.	Deadline	Batas waktu untuk pengumpulan dana
7.	Goal	Jumlah dana yang ingin dikumpulkan, yang dibutuhkan oleh para pembuat untuk menyelesaikan proyeknya
8.	launched	Tanggal proyek tersebut diluncurkan
9.	pledged	Jumlah dana yang sudah dikumpulkan
10.	State	Status atau kondisi proyek pada saat ini
11.	Backers	Jumlah orang yang mendukung proyek ini.
12.	Usd_goal_real	Jumlah uang yang ingin dikumpulkan dalam satuan dolar AS yang dikonversi menggunakan API fixer.io
13.	Usd_pledged	Jumlah uang yang telah dikumpulkan dalam satuan dolar AS yang dikonversi dalam situs KS.
14.	Usd_pledged_real	Jumlah uang yang telah dikumpulkan dalam satuan dolar AS yang dikonversi menggunakan API fixer.io
15.	Country	Negara Asal pembuat proyek.

3.4 Exploratory Data

Tahapan selanjutnya adalah *exploratory data analysis*. Pada tahapan ini penulis melakukan Analisa data, seperti melihat jenis tipe data data dan kualitas data dengan mencari apakah ada variabel yang memiliki nilai *null* atau kosong yang nantinya akan dibersihkan pada tahapan selanjutnya. Kemudian menjabarkan data dalam bentuk grafik dan diagram supaya dapat melihat kategori dan main kategori yang paling banyak mendapat dukungan dan sukses dalam proyek kickstarter.

3.5 Data Preparation & Feature Engineering

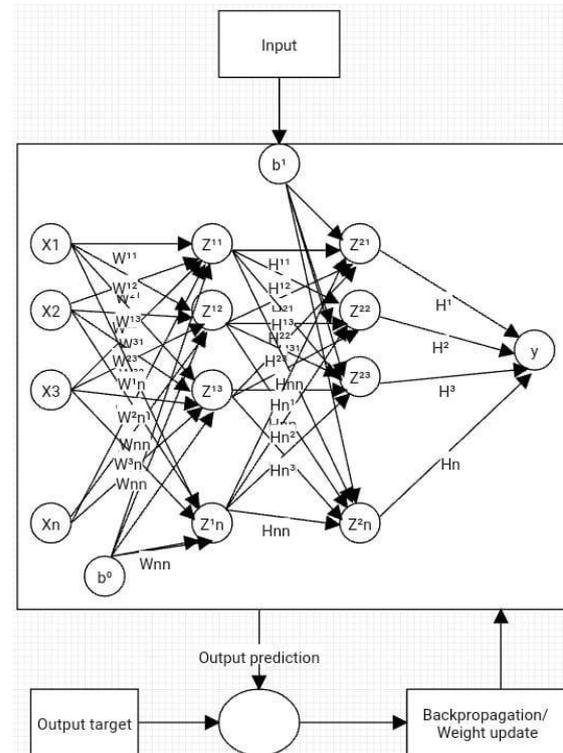
Pada tahapan ini data yang sudah dikumpulkan diolah menjadi data yang siap untuk digunakan dalam membuat model yang nantinya akan dilatih dan diuji. *Preparation* data yang dilakukan diantaranya menghapus

variabel yang tidak diperlukan dalam penelitian, kemudian melakukan *feature engineering* dengan teknik *one-hot encoder* pada variabel yang akan digunakan yaitu proses mengubah variabel dengan tipe data kategorikal menjadi numerikal, dikarenakan model prediksi yang akan dilakukan hanya menerima nilai numerik bukan nilai kategorikal, selanjutnya persiapan data dengan menerapkan *feature scaling* pada variabel yang sudah diubah menjadi data numerik menggunakan standar *scaler* dari scikit-learn untuk membuat numerik data yang ada pada dataset memiliki rentang nilai yang sama dan tidak terlalu jauh sehingga tidak ada satu variabel data yang mendominasi variabel data lainnya..

3.6 Build Model

Pada tahapan ini data yang sudah dikumpulkan dan disiapkan akan digunakan untuk membuat model *machine learning* dengan menggunakan 2 algoritma, antara lain:

3.6.1 Menggunakan Teknik jaringan syaraf tiruan atau ANN.



Gambar 3.2 blok diagram alur algoritma ANN

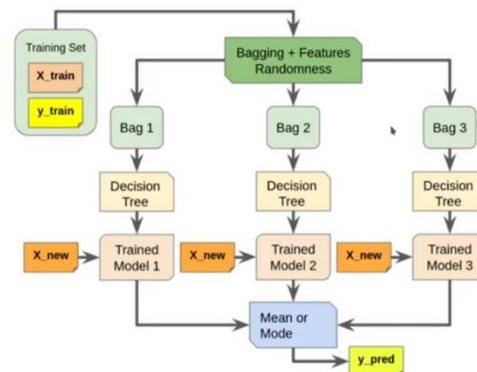
Berdasarkan gambar 3.2 yaitu alur algoritma *artificial neural network*, Pada algoritma ini, yang dilakukan pertama kali adalah menentukan jumlah *feature* yang akan digunakan untuk nilai *input* pada model prediksi ini terdapat 218 kemudian setelah itu tentukan nilai atau jumlah *neuron* yang ada pada *hidden layer* beserta fungsi aktivasi yang akan digunakan model ini diantaranya yaitu 4, 16, 32, 64 dan 128 *neuron* dengan fungsi aktivasi relu. Setelah menentukan jumlah *hidden layer* beserta jumlah *neuron*.

selanjutnya menentukan *neuron* pada *layer output* atau keluaran beserta fungsi aktivasi nya yaitu pada *layer* ini terdapat 1 *neuron* sebagai hasil dengan fungsi aktivasi sigmoid. Setelah menentukan parameter yang akan digunakan selanjutnya *build model* dengan parameter variabel *input* dan *output*. Kemudian model yang sudah dibuat akan di-*compile* dengan beberapa parameter seperti *optimizer*, *loss* dan *metric* penilaian yang akan digunakan untuk melihat hasil model yang sudah dilatih.

Untuk model prediksi ini karena menggunakan *backpropagation* maka *optimizer* yang digunakan adalah *stochastic gradient descent* dengan *learning rate* 0,1 dan 0,5 sebagai parameter yang digunakan untuk *update* nilai bobot dan bias pada saat melakukan algoritma *backpropagation* dan untuk nilai *loss* yang digunakan adalah MSE atau *Mean Squared Error* untuk menghitung nilai *loss* pada model dan untuk *metric* menggunakan *accuracy* yang nantinya hasil *training* yang akan dilihat adalah nilai *accuracy* dan *loss* nya. a jika nilai *accuracy* tinggi diatas 80% serta nilai *loss* yang rendah dibawah 0.1 atau 10% menandakan kualitas model yang baik dan mampu melakukan prediksi dengan baik dan benar.

Kemudian jika nilai tersebut belum sesuai yang diinginkan maka akan dilakukan peningkatan bobot dengan algoritma *backpropagation* lalu setelah bobot diperbarui, model akan melakukan pengulangan *running neural network* sampai nilai *error* yang didapatkan sesuai yang ditentukan.

3.6.1 Menggunakan Teknik *Random forest*



Gambar 3.3 diagram blok alur algoritma *random forest*

Berdasarkan gambar 3.3 diatas yang merupakan alur kerja algoritma *random forest*. Setelah data dipecah menjadi dua yaitu *training* dan *testing* menjadi *training set*, kemudian data akan diambil untuk dibentuk sebuah pohon dengan cara acak untuk setiap node pada tree yang sudah ditentukan. Kemudian dari hasil *trained model* yang sudah ditentukan akan dilakukan *testing* menggunakan data *testing* setelah hasilnya didapat kemudian akan dilakukan *majority voting* untuk mendapatkan model prediksi terbaik.

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
rf = RandomForestClassifier(n_estimators=100, random_state=0)
rf.fit(X_train, y_train)

prediction_rf = rf.predict(X_test)
print("RandomForest Accuracy Score -> ", accuracy_score(y_test, prediction_rf))
```

Gambar 3.4 algoritma *random forest*

Berdasarkan gambar 3.4 yaitu Algoritma *Random forest* dimana pada model algoritma ini setelah melakukan *splitting* data *train* dan *testing* kemudian membangun model *random forest* dengan beberapa parameter yang digunakan seperti nilai *n_estimators* atau jumlah pohon keputusan yang akan dibuat sebelum melakukan *voting* untuk *random forest* dan untuk model kali ini menggunakan nilai *n_estimators* 50, 100, 200 dan 500 dengan nilai *random_state* nya 0 untuk membantu memastikan hasil dengan nilai yang konsisten. Setelah menentukan parameter pada bagian

initialization model kemudian dilanjutkan dengan menjalankan model yang sudah dibuat dengan parameter data *training* yang sudah di split yaitu *X_train* dan *y_train* setelah model dijalankan maka kemudian akan dilakukan prediksi pada data *testing* atau uji untuk melihat hasil atau nilai akurasi nya. Selain itu menggunakan *confussion matrix* sebagai *metric* penilaian kualitas model. Dengan *confussion matrix* ini maka akan menampilkan *accuracy*, *precision*, *recall* dan *f1-score*. Dimana terdapat dua kategori *output* pada model ini yaitu 0 dan 1. Model baik yang memiliki nilai *f1-score* tinggi, karena nilai *f1-score* mempengaruhi nilai lainnya seperti nilai *accuracy*, *precision* dan *recall*.

3.7 Evaluasi Model

Tahapan ini merupakan lanjutan dari tahapan sebelumnya yaitu *build model*, setelah model selesai dibuat dan dilatih maka akan dievaluasi model mana yang terbaik dan bagian mana yang harus diperbaiki serta variabel mana saja yang mempengaruhi prediksi pada kickstarter. Jika model sudah memenuhi dan memberi hasil yang sesuai maka model tersebut yang akan digunakan. Tetapi jika belum sesuai yang diharapkan maka akan mengulang tahapan persiapan data untuk diolah lagi supaya model yang dilatih bisa memberikan hasil yang baik.

3.8 Penulisan Laporan

Tahapan terakhir yaitu penulisan laporan, tahapan ini dilakukan jika sudah mendapat hasil dari penelitian ini dan tentunya jika sudah mendapatkan model prediksi yang baik dan memiliki nilai akurasi yang tinggi. Laporan ini nantinya berisi hasil dari penelitian yang sudah dilakukan, mulai dari permasalahan yang dihadapi sampai mendapatkan hasil yang diinginkan terkait prediksi program kickstarter.

4 HASIL DAN PEMBAHASAN

4.1 Hasil

Pada penelitian ini menggunakan 2 algoritma pada model prediksinya supaya ada perbandingan antara satu model dengan model lainnya, dan pada pengerjaan penelitian ini sebagai bahan perbandingan maka ditentukan variabel atau fitur yang digunakan untuk setiap modelnya, untuk pelatihan atau *training*

pertama model menggunakan 10 variabel dari 15 variabel yang ada pada dataset yaitu *main_category*, *category*, *country*, *goal*, *backers*, *pledged*, *launched*, *currency* dan *deadline*. Kemudian setelah di-*encoding* menjadi 219 data yang mana 218 variabel sebagai *input* kemudian 1 variabel *state/ status* akan dijadikan fitur label atau target dalam model. Sedangkan untuk pelatihan yang kedua hanya menggunakan 8 fitur atau variabel yang sudah dilakukan *encoding* menjadi 217 data, yang mana 216 sebagai *input* dan 1 variabel sebagai label *output*.

4.1.1 Hasil Algoritma ANN

Algoritma yang pertama adalah model dengan menggunakan algoritma *Artificial Neural Network*(ANN) dengan *Backpropagation*. ANN sendiri merupakan salah satu algoritma jaringan syaraf tiruan yang mana dalam membuat model menggunakan banyak *layer* di dalamnya. Dan dengan algoritma *Backpropagation* ini digunakan untuk memperkecil nilai kesalahan atau *error* serta mengupdate nilai bias dan nilai bobot pada masing-masing perceptron.

4.1.1.1 Hasil *Training* Pertama (percobaan pertama)

- Dengan 1 *Hidden Layer*

Tabel 4.1 Perbandingan jumlah *neuron hidden layer* dengan *learning rate* 0.1

Jumlah <i>neuron</i> Hidden layer	<i>Learning rate</i> 0.1			
	<i>Acc</i>	<i>Val_acc</i>	<i>Loss</i>	<i>Val_loss</i>
4	0.8646	0.8594	0.0998	0.1029
16	0.8679	0.8643	0.0981	0.1006
32	0.8692	0.8624	0.0969	0.1018
64	0.8694	0.8688	0.0979	0.0990
128	0.8964	0.8838	0.0782	0.0865

Berdasarkan tabel 4.1, yang merupakan tabel perbandingan antara jumlah *neuron* pada *hidden layer* dengan nilai *learning rate* 0.1 pada pelatihan pertama menggunakan 1 *hidden layer* dapat dilihat bahwa hasilnya adalah memiliki nilai terbaik dengan *accuracy* tertinggi dan *loss* terendah dengan jumlah *neuron* 128

dan *learning rate* 0.1 yaitu nilai acc 0.89 dan val_acc 0.88, loss 0.0782, val_loss 0.0865.

Tabel 4.2 Hasil Perbandingan jumlah *neuron hidden layer* dengan *learning rate* 0.5

Jumlah <i>neuron</i> pada <i>hidden layer</i>	<i>Learning rate</i> 0.5			
	Acc	Val_acc	Loss	Val_loss
4	0.8674	0.8674	0.0980	0.0981
16	0.8704	0.8672	0.0962	0.0980
32	0.8714	0.8655	0.0939	0.0981
64	0.8722	0.8626	0.0939	0.0992
128	0.8894	0.8847	0.0823	0.0875

Berdasarkan tabel 4.2 yang merupakan hasil dari perbandingan antara jumlah *neuron hidden layer* dan *learning rate* 0.5 pada *training* pertama menggunakan 1 *hidden layer* dapat dilihat bahwa hasil terbaik yaitu dengan jumlah *neuron* 128 dengan memiliki nilai acc 0.8894, val_acc 0.8847, loss 0.0823 dan val_loss 0.0875.

Dari kedua tabel diatas untuk percobaan pertama menggunakan 1 *hidden layer* dengan jumlah *neuron* yang sama dan 2 *learning rate* yang berbeda dapat disimpulkan bahwa model terbaik untuk percobaan ini adalah *neuron* 128 dan learning rat 0.1 dengan nilai acc 0.8964, val_Acc 0.8838, loss 0.0782 dan val_loss 0.0865.

- Dengan 2 Hidden Layer

Tabel 4.3 Hasil perbandingan jumlah *neuron hidden layer* dengan *learning rate* 0.1

Jumlah <i>neuron</i> ada <i>hidden layer</i>	<i>Learning rate</i> 0.1			
	Acc	Val_acc	Loss	Val_loss
4	0.8737	0.8683	0.0916	0.0951
16	0.8742	0.8690	0.0910	0.0941
32	0.8739	0.8635	0.0915	0.0980
64	0.8759	0.8667	0.0901	0.0962

128	0.8762	0.8684	0.0891	0.0943
-----	--------	--------	--------	--------

Berdasarkan tabel 4.3 yang merupakan hasil dari percobaan pertama menggunakan 2 *hidden layer* yang memiliki nilai yang sama pada *hidden layer* pertama yaitu 128 dan untuk *hidden layer* yang kedua yaitu 4,16,32,64 dan 128 dengan *learning rate* 0.1. dari data diatas didapat model yang terbaik dengan jumlah *neuron* 128 dengan nilai acc 0.8762, val_acc 0.8684, loss 0.0891 dan val_loss 0.0943.

Tabel 4.4 hasil perbandingan jumlah *neuron hidden layer* dengan *learning rate* 0.5

Jumlah <i>neuron</i> pada <i>hidden layer</i>	<i>Learning rate</i> 0.5			
	Acc	Val_acc	Loss	Val_loss
4	0.8736	0.8688	0.0921	0.0968
16	0.8798	0.8731	0.0874	0.0926
32	0.8802	0.8725	0.0870	0.0932
64	0.8829	0.8671	0.0848	0.0943
128	0.8936	0.8911	0.0776	0.0806

Berdasarkan tabel 4.4 yang merupakan hasil dari percobaan pertama menggunakan 2 *hidden layer* yang masing-masing jumlah *neuron* pada layer pertama yaitu 128 dan layer kedua yaitu 4, 16, 32, 64, 128 dengan *learning rate* 0.5. dapa dilihat model terbaik pada percobaan ini didapat dari *hidden layer* yang memiliki jumlah *neuron* 128 yaitu dengan nilai acc 0.8936, val_acc 0.8911, loss 0.0776 dan val_loss 0.0806.

Dari kedua tabel diatas yang merupakan hasil dari percobaan pertama dengan 2 *hidden layer* dan dengan jumlah *neuron* yang sama tetapi dengan nilai *learning rate* berbeda didapat hasil terbaik dari *neuron* 128 dengan *learning rate* 0.5 dengan nilai acc 0.8936, val_acc 0.8911, loss 0.0776 dan val_loss 0.0806.

4.1.1.2 Hasil *Training* kedua (percobaan kedua)

- Dengan 1 *hidden layer*

Dengan menggunakan *learning rate* 0.1 yang mendapat hasil terbaik adalah *neuron* 128 dengan nilai *acc* 0.688, *val_acc* 0.6816, *loss* 0,1997 dan *val_loss* 0.2038. sedangkan, *learning rate* 0.5. hasilnya tidak stabil, nilai *acc* tertinggi dengan jumlah *neuron* 64 tetapi hasil terbaik lain nya dimiliki oleh *neuron* 128. Maka disimpulkan hasil terbaik adalah jumlah *neuron* 128 dengan *acc* 0.6877 dan *val_acc* 0.6841 dan nilai *loss* 0.2.

- Dengan 2 *hidden layer*

Dengan menggunakan *learning rate* 0.1 didapat hasil terbaiknya yaitu pada saat jumlah *neuron* 128 dengan nilai *acc* 0.6942, *val_acc* 0.6849, *loss* 0.1965 dan *val_loss* 0.2025, sedangkan *learning rate* 0.5 yang mana pada lapisan pertama memiliki jumlah *neuron* yang sama yaitu 128 dan jumlah *neuron* kedua yang berbeda yaitu 4,16,32,64,128. Didapat hasil yang terbaik pada percobaan ini yaitu pada jumlah *neuron* 128 dengan nilai *acc* 0.6941 dan *loss* 0.1961.

Dari setiap perlakuan dan percobaan sudah didapat masing-masing nilai terbaiknya, dan semua hasil terbaik didapat pada jumlah *neuron* 128 menandakan jumlah *neuron* yang terdapat pada *hidden layer* sangat berpengaruh untuk hasil dari model, semakin banyak maka hasil yang didapat juga semakin baik. Selain itu jumlah *hidden layer* yang digunakan juga berpengaruh, terbukti hasil yang didapatkan menggunakan 2 *hidden layer* lebih baik dibandingkan hanya menggunakan 1 *hidden layer*.

4.1.2 Hasil dengan Algoritma *Random forest*

Algoritma selanjutnya adalah menggunakan *random forest* dengan *n_estimator* yang akan menjadi parameter nya. Nilai dari estimator ini yang menentukan model akan menggunakan berapa banyak pohon keputusan untuk nantinya dilakukan *voting* pohon terbaik yang akan digunakan pada *random forest* dengan *random state* 0 yang digunakan untuk memeriksa dan memvalidasi data saat menjalankan kode

beberapa kali. Mengatur *random_state* nilai tetap akan menjamin urutan nomor acak yang sama dihasilkan setiap kali Anda menjalankan kode. Dan kecuali ada beberapa keacakan lain dalam proses, hasil yang dihasilkan akan sama seperti biasa. Ini membantu dalam memverifikasi *output*.

4.1.2.1 Hasil Percobaan Pertama

Tabel 4.5 hasil perbandingan *n_estimator* dengan *random state* 0

N_e stim ator	Random state 0						
	Acc	Precision		Recall		F1_score	
		0	1	0	1	0	1
50	0.98	0.	0.96	0.	0.98	0.	0.97
	34	99	83	98	57	98	69
		20		21		70	
100	0.98	0.	0.96	0.	0.98	0.	0.97
	37	99	77	98	72	98	74
		28		18		73	
200	0.98	0.	0.96	0.	0.98	0.	0.97
	39	99	82	98	72	98	76
		28		20		74	
500	0.98	0.	0.96	0.	0.98	0.	0.97
	40	99	31	98	77	98	78
		31		20		75	

Berdasarkan tabel 4.5 yang merupakan hasil dari algoritma *random forest* pada percobaan pertama yang menggunakan variabel *input* 218 dapat dilihat hasilnya antara perbandingan jumlah estimator dengan *random state* 0, dari tabel tersebut model terbaik atau nilai tertinggi didapat dari *random forest* dengan jumlah estimator 500 yaitu dengan nilai *acc* 0.9840 dan nilai *f1-score* nya 0,9875 dan 0.9778 untuk masing-masing kategori 0 dan 1. Dan dapat dilihat dari tabel diatas nilai yang didapat akan meningkat dengan bertambah besarnya jumlah estimator.

4.1.2.1 Hasil percobaan kedua

hasil dari algoritma *Random forest* pada percobaan kedua dengan menggunakan *input layer* 216. Dengan hasil perbandingan antara jumlah estimator atau pohon keputusan dengan *random state* didapat model terbaik atau nilai tertinggi pada jumlah estimator 500 dengan

nilai accuracy 0.6577 dan nilai f1-score 0.7425 dan 0.4893 pada masing kategori 0 dan 1.

4.2 Pembahasan

Berdasarkan dari hasil diatas nilai akurasi dari ke-2 algoritma dan 2 pelatihan, dapat dilihat hasil yang sudah diperoleh untuk kedua model didapatkan pada percobaan pertama dengan menggunakan *input* 218. Pada *backpropagation*, hasil yang didapat mengalami peningkatan pada akurasi dan penurunan nilai *loss* hal tersebut membuktikan bahwa algoritma ini dapat menurunkan nilai *error* dengan mengupdate bobot dan bias disetiap pelatihannya. Dan untuk algoritma ini akurasi tertinggi didapat dengan menggunakan parameter *learning rate* 0.5 dan dengan 2 *hidden layer* yang masing-masing *hidden layer* berjumlah 128 *neuron* dengan hasil *accuracy* 0.8936, *val_acc* 0.8911, *loss* 0.0776 dan *val_loss* 0.0806. Pada algoritma *Random forest* berdasarkan hasil yang didapat semakin banyak estimator atau pohon yang digunakan maka nilai akurasinya pun semakin tinggi, terbukti nilai akurasi tertinggi pada algoritma ini didapat dengan menggunakan *n_estimator* 500 dengan *random_state* 0 menghasilkan nilai *accuracy* 0.9840 dan nilai *f1-score* nya 0,9875 dan 0.9778 untuk masing-masing kategori 0 dan 1.

Sedangkan percobaan kedua menggunakan 216 *input*, hanya bisa menghasilkan akurasi 69% pada *Backpropagation* dan 65% pada *Random forest*. Sehingga bisa disimpulkan hasil yang didapat pada percobaan pertama model terbaik adalah menggunakan *Random forest*, sedangkan pada percobaan kedua model terbaik menggunakan *Backpropagation*.

Sehingga model terbaik untuk penelitian ini algoritma *random forest* dengan data input 218 menggunakan parameter estimator 500.

5 KESIMPULAN

1. Berdasarkan hasil yang diperoleh dari penelitian ini, algoritma yang memiliki performa terbaik untuk model prediksi kesuksesan kickstarter ini adalah *Random forest* dengan nilai akurasi 98%, dan hasil *f1-score* untuk kedua kategori yaitu sukses (1) dan tidak sukses (0) adalah 97% dan 98%.
2. Variabel yang sangat berpengaruh untuk kesuksesan atau tidaknya proyek kickstarter adalah *pledge* (jumlah uang

yang dijanjikan *supporter* untuk proyek tersebut) dan *backer* (pendukung)

3. Berdasarkan hasil yang diperoleh model prediksi terbaik dari kedua algoritma adalah pada data *input* 218, untuk *Backpropagation* menggunakan parameter 2 *hidden layer* dengan jumlah *neuron* pada *hidden layer* 128, dan untuk *Random Forest* menggunakan parameter 500 estimator.
4. Berdasarkan hasil penelitian, pada algoritma *random forest* banyaknya estimator (banyak pohon yang terbentuk) dapat mempengaruhi tingkat akurasi dimana semakin banyak pohon semakin baik akurasinya.
5. Pada algoritma *backpropagation*, nilai parameter *learning rate* mempengaruhi tingkat akurasi pada model prediksi dimana semakin besar nilai *learning rate* atau semakin mendekati nilai 1 maka semakin baik akurasi yang didapat.

UCAPAN TERIMA KASIH

Terimakasih saya ucapkan kepada semua pihak Teknik Informatika yang telah membantu saya dalam mengerjakan jurnal penelitian ini hingga selesai

DAFTAR PUSTAKA

- [1] Chalimah, Claudia Vilahi. "Model Penjelasan Keputusan Berdonasi Melalui Layanan Pendanaan Berbasis Financial Technology (Crowdfunding)". 2016. [online]. Available: <https://dspace.uui.ac.id/bitstream/handle/123456789/11100/05.%20BAB%20I.pdf?sequence=5&isAllowed=y>, [accessed 10 februari 2022]
- [2] Chairunisa. "Pengetahuan, Kepercayaan, Informasi dan Teknologi yang Mempengaruhi Motivasi Masyarakat dalam Investasi Menggunakan Crowdfunding Syariah" 2018. [online]. Available: <https://repository.uinjkt.ac.id/dspace/bitstream/123456789/41199/1/CHAIRUNISA-FEB.pdf>, [diakses pada 10 februari 2022 pukul 14.15]
- [3] Integrity, Indonesia, "Crowdfunding: Resiko Kecurangan pada dana Patungan Online" 2018. [online]. Available: <https://www.integrity->

- indonesia.com/id/blog/2018/09/13/crowdfunding-risiko-kecurangan-pada-dana-patungan-online/, [diakses pada 10 Februari 2022 pukul 14.50]
- [4] Charman, Ashley, "The FTC is investigating a crowdfunding campaign that disappeared with more than \$700K" 2018. [online]. Available: <https://www.theverge.com/circuitbreaker/2018/8/29/17793720/ftc-ibackpack-investigation-indiegogo-kickstarter>, [diakses pada 10 Februari pukul 15.00]
- [5] Setiawan, Betania Jezamin, dkk, "Crowdfunding: Aspek Kemitraan pada Penyelenggaraan (Studi pada platform gandengtangan)" 2020. [online]. Available: <https://journal.paramadina.ac.id/index.php/madani/article/download/491/214/1977>, [diakses pada 10 Februari 2022 pukul 15.45]
- [6] Go Travels, "Apa Itu Kickstarter" 2022. [online]. Available: <https://id.go-travels.com/17574-what-is-kickstarter-3486258-9020894>, [diakses pada 10 Februari 2022 pukul 16.00]
- [7] Rajendra, Laksmana, "Artificial Neural Network" 2022. [online]. Available: <http://sistem-informasi-s1.stekom.ac.id/informasi/baca/Artificial-Neural-Network/b1c26e9347ef547ff06845ca38cc443aedc4fa86>, [diakses pada 10 Februari 2022 pukul 16.00]
- [8] Prasetyo, Hendro, "Backpropagation Neural Network(Jaringan Syaraf Tiruan)" 2019. [online]. Available: <https://hendroprasetyo.com/backpropagation-neural-network-jaringan-saraf-tiruan/#.YmAOWShBzIU>, [diakses pada 10 Februari 2022 pukul 19.45]
- [9] Novindasari, Ida, "Pengaruh Ukuran Batch dan *Learning rate* dalam Konvergensi Gradient Descent" 2020. [online]. Available: <https://idanovinda.medium.com/pengaruh-ukuran-batch-dan-learning-rate-dalam-konvergensi-gradient-descent-ebfe6f4cae75>, [diakses pada 10 Februari 2022 pukul 22.15]
- [10] Wikipedia, "Random forest " 2021. [online]. Available: https://id.wikipedia.org/wiki/Random_forest, [diakses pada 12 Februari 2022 pukul 10.06]
- [11] Rahmalia, Nadiyah, "Kaggle, komunitas belajar data science yang bisa menghasilkan uang" 2021. [online]. Available: <https://glints.com/id/lowongan/kaggle-adalah/#.YmAjuihBzIU>, <https://www.kaggle.com/datasets/kemical/kickstarter-projects>, [diakses pada 13 februari 2022 pukul 10.40]