

**ANALISIS PENGEMBANGAN MODEL PREDIKSI KESUKSESAN
KICKSTARTER MENGGUNAKAN *ALGORITMA BACKPROPAGATION*
ARTIFICIAL NEURAL NETWORK(ANN) DAN *RANDOM FOREST***

(Skripsi)

Oleh

Indria Agustina

NPM 1815061021



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG**

2022

ABSTRAK

ANALISIS PENGEMBANGAN MODEL PREDIKSI KESUKSESAN KICKSTARTER MENGGUNAKAN ALGORITMA BACKPROPAGATION ARTIFICIAL NEURAL NETWORK(ANN) DAN RANDOM FOREST

Oleh

Indria Agustina

Pertumbuhan pendanaan proyek melalui *Crowdfunding*, khususnya salah satu portal *reward-based Crowdfunding* yaitu kickstarter memiliki kecenderungan untuk meningkat setiap tahunnya. Berdasarkan pernyataan tersebut, penelitian ini berfokus untuk mengembangkan model prediksi tingkat kesuksesan kickstarter supaya *fundraiser* dapat merancang strategi untuk meningkatkan kesuksesan proyek dan juga melakukan antisipasi terhadap potensi risiko yang timbul akibat tingkat kesuksesan yang dicapai misalnya kecurangan pendanaan dana untuk *funder*. 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. Ada dua percobaan yang dilakukan, yang mana pada percobaan pertama model prediksi dikembangkan dengan 10 variabel (9 variabel sebagai *input* dan 1 variabel sebagai label *output*) yang ada pada dataset sedangkan pada percobaan kedua hanya 8 variabel (7 variabel sebagai *input* dan 1 variabel sebagai label *output*), dikurangi variabel tentang *pledge* dan *backers* pada pengembangan model prediksi. Hasil untuk percobaan pertama menggunakan algoritma *Backpropagation* ANN dan *Random Forest* memberikan akurasi tertinggi masing-masing 89% dan 98%, sedangkan model kedua memberikan akurasi tertinggi masing-masing 69% dan 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. Sehingga berdasarkan data hasil yang sudah dimiliki dapat disimpulkan bahwa model yang akan digunakan untuk memprediksi kesuksesan kickstarter adalah model pertama dengan menggunakan algoritma *Random Forest* dengan *n_estimator* 500 yang merupakan model terbaik dengan nilai akurasi 98% dan nilai *f1-score* 98%.

Kata Kunci: Kickstarter, *Crowdfunding*, Prediksi, *Artificial Neural Network*, *Random Forest*.

ABSTRACT

ANALYSIS OF THE DEVELOPMENT OF THE KICKSTARTER SUCCESS PREDICTION MODEL USING THE BACKPROPAGATION ARTIFICIAL NEURAL NETWORK(ANN) DAN RANDOM FOREST ALGORITHM

By

Indria Agustina

The growth of project funding through Crowdfunding, especially one of the reward-based Crowdfunding portals, namely Kickstarter, has a tendency to increase every year. Based on this statement, this research focuses on developing a kickstarter success rate prediction model so that fundraisers can design strategies to increase project success and also anticipate potential risks that arise due to the level of success achieved, such as fraudulent funding of funds for funders. This prediction model was built with a classification approach to data mining with 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. There were two experiments carried out, where in the first experiment a prediction model was developed with 10 variables (9 variables as input and 1 variable as output label) in the dataset while in the second experiment only 8 variables (7 variables as input and 1 variable as label). output), minus the variables regarding pledges and backers in the development of the prediction model. The results for the first experiment using the ANN Backpropagation algorithm and Random Forest gave the highest accuracy of 89% and 98%, respectively, while the second model gave the highest accuracy of 69% and 65.7%, respectively. The results of this study indicate that the reduction of the pledge and backers variables can reduce the performance of the prediction model. In terms of utilization, the first and second stage 2 prediction models are for comparison. So, based on the existing data, 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 with n_estimator 500 which is the best model with an accuracy value of 98% and an f1-score value of 98%.

Keywords: Kickstarter, Crowdfunding, Prediction, Artificial Neural Network, Random Forest.

**ANALISIS PENGEMBANGAN MODEL PREDIKSI KESUKSESAN
KICKSTARTER MENGGUNAKAN ALGORITMA *BACKPROPAGATION*
ARTIFICIAL NEURAL NETWORK(ANN) DAN *RANDOM FOREST***

Oleh

Indria Agustina

Skripsi

**Sebagai Salah Satu Syarat Untuk Mencapai Gelar
SARJANA TEKNIK**

Pada

**Program Studi Teknik Informatika
Jurusan Teknik Elektro
Fakultas Teknik Universitas Lampung**



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2022**

Judul Skripsi : ANALISIS PENGEMBANGAN MODEL PREDIKSI
KESUKSESAN KICKSTARTER MENGGUNAKAN
ALGORITMA BACKPROPAGATION
ARTIFICIAL NEURAL NETWORK (ANN)
DAN RANDOM FOREST

Nama Mahasiswa : *Indria Agustina*

Nomor Pokok Mahasiswa : 1815061021

Program Studi : Teknik Informatika

Jurusan : Teknik Elektro

Fakultas : Teknik



1. Komisi Pembimbing

Yessi Mulyani, S.T., M.T.
NIP 19731226 200012 2 001

Trisya Septiana, S.T., M.T.
NIP 19900921 201903 2 025

2. Mengetahui

Ketua Jurusan
Teknik Elektro

Herlinawati, S.T., M.T.
NIP 19710314 199903 2 001

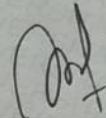
Ketua Program Studi
Teknik Informatika

Mona Arif Muda, S.T., M.T.
NIP 19711112 200003 1 002

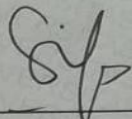
MENGESAHKAN

1. Tim Penguji

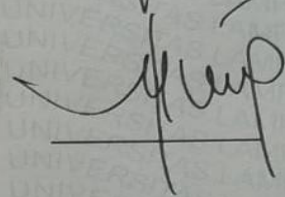
Ketua : **Yessi Mulyani, S.T., M.T.**



Sekretaris : **Trisya Septiana, S.T., M.T.**



Penguji : **Dr. Eng. Mardiana, S.T., M.T.**



2. Dekan Fakultas Teknik



Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc. ✎
NIP. 19750928 200112 1 002

Tanggal Lulus Ujian Skripsi : **18 Juli 2022**

SURAT PERNYATAAN

Saya yang bertandatangan dibawah ini, menyatakan bahwa skripsi saya yang berjudul "ANALISIS PENGEMBANGAN MODEL PREDIKSI KICKSTARTER MENGGUNAKAN ALGORITMA BACKPROPAGATION ARTIFICIAL NEURAL NETWORK(ANN) DAN RANDOM FOREST" dengan ini menyatakan bahwa skripsi saya dibuat oleh saya sendiri. Semua hasil yang tertuang dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti bahwa skripsi ini merupakan salinan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi sesuai dengan ketentuan hukum atau akademik yang berlaku.

Bandar Lampung, 11 agustus 2022

Penulis,



Indria Agustina
Indria Agustina

1815061021

RIWAYAT HIDUP



Penulis dilahirkan di Karang Anyar, Marga Sekampung Lampung Timur pada tanggal 17 Agustus 1999, sebagai anak kedua dari empat bersaudara, Bapak Teguh Antoni dan Ibu Risma Sari. Penulis menyelesaikan pendidikan Sekolah Dasar di SDN 1 Bungkok, Marga Sekampung pada tahun 2011, Sekolah Menengah Pertama di SMPN 1 Marga Sekampung, Lampung Timur pada tahun 2014, dan Sekolah Menengah Atas (SMA) di SMAN 1 Waway Karya, Lampung Timur pada tahun 2017. Tahun 2018, penulis terdaftar sebagai mahasiswa Program Studi Teknik Informatika Universitas Lampung melalui jalur SBMPTN. Selama menjadi mahasiswa, penulis pernah menjadi bagian dari Organisasi Himpunan Mahasiswa Teknik Elektro (HIMATRO) Unila serta di Unit Kegiatan Mahasiswa (UKM) Sains dan Teknologi Unila sebagai anggota Departemen Komunikasi dan Informasi. Pada tahun 2019, penulis pernah mengikuti Program Kreativitas Mahasiswa (PKM) di bidang kewirausahaan. Pada tahun 2021, penulis mendapat kesempatan untuk mengikuti program Bangkit Academy dengan bidang Machine Learning. Pada tahun yang sama, penulis mengikuti Studi Independen di Dicoding dengan bidang Pengembangan Machine Learning dan Front End.

MOTO

“Boleh jadi kamu membenci sesuatu padahal ia amat baik bagimu, dan boleh jadi pula kamu menyukai sesuatu padahal ia amat buruk bagimu, Allah mengetahui sedang kamu tidak” (QS: Al-baqarah ayat 216)

“Allah tidak membebani seseorang itu melainkan sesuai dengan kesanggupannya”
(QS: Al-baqarah ayat 286)

“dan bersabarlah. Sesungguhnya Allah beserta orang-orang yang bersabar”
(QS: Al-anfaal ayat 46)

“Hidup ini bukan tentang seberapa cepat kamu berlari atau seberapa tinggi kamu mendaki, tetapi seberapa baik kamu melambung” (Vivian komori)

“Banyak kegagalan hidup terjadi pada orang-orang yang tidak menyadari seberapa dekat mereka dengan kesuksesan Ketika mereka menyerah”
(Thomas A. Edison)

PERSEMBAHAN



Skripsi ini kupersembahkan untuk orang yang menjadi motivasi terbesar dalam segala kegiatan dalam hidupku, ibu dan bapak. Karya kecil ini kupersembahkan kepada mu sebagai tanda bakti, hormat, dan rasa terima kasih atas segala dukungan selama hidupku

Ibuku, yang telah melahirkanku, merawatku, membesarkanku, dan yang telah sepenuh hati mendidikku.

Ayahku tercinta, yang telah membesarkanku dengan seluruh kasih dan sayangnnya, memberikan pengetahuannya, dan selalu mendukung serta mendoakan untuk keberhasilanku.

Ibu, bapak skripsi ini salah satu bukti kegigihan dan kerja keras kalian dalam mendukung keinginan dan cita-cita anakmu.

Serta, almamater yang saya sangat banggakan

UNIVERSITAS LAMPUNG

SANWACANA

Puji syukur penulis ucapkan kehadirat Tuhan Yang Maha Esa, karena atas rahmat dan hidayah-Nya skripsi ini dapat diselesaikan. Skripsi dengan judul “Analisis Pengembangan Model Kickstarter Menggunakan Algoritma *Backpropagation Artificial Neural Network(ANN)* dan *Random Forest*” adalah salah satu syarat untuk memperoleh gelar Sarjana Teknik di Universitas Lampung.

Dalam kesempatan ini penulis mengucapkan terima kasih kepada:

1. Dr. Eng. Helmy Fitriawan, S.T., M.Sc. selaku Dekan Fakultas Teknik Universitas Lampung.
2. Ibu Herlinawati, S.T.,M.T. selaku Ketua Jurusan Teknik Elektro Universitas Lampung.
3. Bapak Mona Arif Muda, S.T., M.T. selaku Ketua Program Studi Teknik Informatika Jurusan Teknik Elektro Universitas Lampung.
4. Ibu Yessi Mulyani, S.T., M.T. selaku Pembimbing Utama yang telah bersedia meluangkan waktu untuk memberikan arahan dan bimbingan hingga skripsi ini selesai.
5. Ibu Trisya Septiana, S.T., M.T. selaku Pembimbing pendamping yang telah bersedia meluangkan waktu untuk memberikan arahan dan bimbingan hingga skripsi ini selesai.
6. Ibu Dr. Eng. Mardiana, S.T., M.T selaku dosen penguji yang telah bersedia meluangkan waktu untuk memeriksa dan memberikan arahan yang baik untuk skripsi ini.
7. Bapak Wahyu Eko Sulistiono, S.T., M.Sc, selaku Pembimbing Akademik yang telah bersedia meluangkan waktu untuk memberikan pengarahan dan bimbingan selama menempuh pendidikan Program Studi Teknik Informatika Universitas Lampung.
8. Seluruh Dosen Program Studi Teknik Informatika yang telah membagikan banyak ilmunya kepada penulis.

9. Mbak Rika yang telah membantu penulis dalam segala urusan administrasi serta membantu dalam pencarian data untuk penelitian ini.
10. Ibu Risma Sari dan Bapak Teguh Antoni selaku orang tua ku tercinta yang telah mendukungku baik secara mental dan materi selama ini.
11. Beni Aris Munandar, Ahmad Bastari, Risky Habib Maulana selaku saudara kandungku serta seluruh keluarga besar yang selalu mendukungku serta telah menjadi keluarga yang mampu membuatku tumbuh dan berkembang dengan baik hingga sekarang.
12. Riky Rahmadi selaku teman dekat yang telah banyak membantu serta memberi dukungan secara mental dan materi sejak proses pendaftaran perkuliahan hingga saat ini.
13. Ahmad Arbain, Hilmi Hermawan, Sawiti Fina Kartika, Uji Khaidah Resti, Nur Halimah dan Novtrilla Putri Amanda yang telah banyak membantu selama proses perkuliahan di Program Studi Teknik Informatika dan telah menjadi teman, sahabat, sekaligus keluarga yang paling berharga selama di Lampung.
14. Seluruh teman teman Teknik Elektro dan Teknik Informatika Angkatan 2018 atas dukungan yang telah diberikan selama menempuh studi di Program Studi Teknik Informatika Universitas Lampung.

Semoga Allah SWT membalas segala bentuk kebaikan hati dan jasa yang telah kalian berikan kepada saya. Saya menyadari meskipun skripsi ini sudah disusun dengan sebaik mungkin, skripsi ini masih terdapat kekurangan dan masih jauh dari kata sempurna, namun saya sangat berharap melalui skripsi ini akan memberikan manfaat bagi siapapun yang membacanya dan bagi penulis dalam mengembangkan dan mengamalkan ilmu pengetahuan yang telah ditempuh selama ini.

Bandar Lampung, 11 Agustus
2022

Penulis,



Indria Agustina
NPM 181506102

DAFTAR ISI

	Halaman
DAFTAR ISI.....	XI
DAFTAR TABEL.....	II
DAFTAR GAMBAR.....	III
I. PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	4
1.6 Sistematika Penulisan.....	4
II. TINJAUAN PUSTAKA.....	6
2.1 <i>Crowdfunding</i>	6
2.2 Kickstarter.....	7
2.3 <i>Artificial Neural Network (ANN)</i>	9
2.4 <i>Stochastic Gradient Descent(SGD)</i>	18
2.5 <i>Random Forest</i>	19
2.6 <i>Confussion Matrix</i>	24
2.9 Penelitian Terkait.....	29
III. METODOLOGI PENELITIAN.....	33
3.1 Waktu dan Tempat.....	33
3.2 Jadwal Penelitian.....	33
3.3 Alat dan Bahan.....	35
3.4.1 <i>Flowchart Penelitian</i>	35
3.4.2 Tahapan Penelitian.....	37

IV. HASIL DAN PEMBAHASAN	44
4.1 Hasil	44
4.1.1 Hasil Algoritma ANN	45
4.1.2 Hasil dengan Algoritma <i>Random Forest</i>	52
4.2 Pembahasan.....	54
V. KESIMPULAN DAN SARAN.....	57
5.1 Kesimpulan	57
5.2 Saran	58
DAFTAR PUSTAKA	60
LAMPIRAN.....	60

DAFTAR TABEL

Tabel	Halaman
Tabel 3.1. Jadwal Penelitian.....	33
Tabel 3.2. Alat Penelitian.....	35
Tabel 3.3. Bahan Penelitian	35
Tabel 3.4. Variabel yang digunakan	38
Tabel 4.1. Hasil perbandingan jumlah <i>neuron hidden layer</i> dg <i>learning rate</i> 0.1	47
Tabel 4.2. Hasil perbandingan jumlah <i>neuron hidden layer</i> dg <i>learning rate</i> 0.5.....	47
Tabel 4.3. Hasil perbandingan jumlah <i>neuron hidden layer</i> dg <i>learning rate</i> 0.1	48
Tabel 4.4. Hasil perbandingan jumlah <i>neuron hidden layer</i> dg <i>learning rate</i> 0.5.....	49
Tabel 4.5. Hasil perbandingan jumlah <i>neuron hidden layer</i> dg <i>learning rate</i> 0.1	49
Tabel 4.6. Hasil perbandingan jumlah <i>neuron hidden layer</i> dg <i>learning rate</i> 0.5.....	50
Tabel 4.7. Hasil perbandingan jumlah <i>neuron hidden layer</i> dg <i>learning rate</i> 0.1	51
Tabel 4.8. Hasil perbandingan jumlah <i>neuron hidden layer</i> dg <i>learning rate</i> 0.5.....	51
Tabel 4.9. Hasil perbandingan <i>n_estimator</i> dengan <i>random state</i>	54
Tabel 4.10. Hasil perbandingan <i>n_estiimator</i> dengan <i>random state</i>	54

DAFTAR GAMBAR

Gambar	Halaman
Gambar 2.1 <i>Crowdfunding</i>	7
Gambar 2.2 Tampilan Halaman Kickstarter	8
Gambar 2.3 Representasi ANN.....	10
Gambar 2.4 Arsitektur ANN.....	11
Gambar 2.5 <i>Backpropagation</i> dengan 1 <i>hidden layer</i>	12
Gambar 2.6 Kurva Fungsi Aktivasi Relu.....	17
Gambar 2.7 Kurva Fungsi Aktivasi Sigmoid.....	17
Gambar 2.8 Diagram <i>Random Forest</i>	22
Gambar 3.1 <i>Flowchart</i> Penelitian	36
Gambar 3.2 Struktur model <i>multilayer</i> pada Algoritma ANN.....	40
Gambar 3.3 Blok diagram algoritma ANN.....	40
Gambar 3.4 Blok diagram algoritma <i>Random Forest</i>	43
Gambar 3.5 Algoritma <i>Random Forest</i>	43
Gambar 4.1 Variabel yang digunakan pada <i>training</i> pertama	45
Gambar 4.2 Variabel yang digunakan pada <i>training</i> kedua.....	46
Gambar 4.3 Model <i>Backpropagation</i> terbaik	53

I. PENDAHULUAN

1.1 Latar Belakang

Saat ini dunia berada di era 4.0, dimana semuanya dilakukan secara digital. Hampir semua aspek kehidupan manusia telah didukung dan dicapai oleh teknologi. Semuanya otomatis digital, mesin cerdas akan diajarkan untuk meniru otak manusia dan memfasilitasi pekerjaan manusia, hal ini disebut pembelajaran mesin atau *machine learning*. 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 (Chalimah, 2018).

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*) (Chairunisa, 2018). 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) (Indonesia, 2018). 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 (Charman, 2018). 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.

Kasus serupa pernah terjadi di Indonesia dimana *fundraiser* kegiatan kemanusiaan yang dituduh menggunakan *crowdfunding* di *platform* kitabisa untuk membeli *smartphone* dan mobil mewah, dengan berdalih bahwa *smartphone* dan mobil digunakan untuk mendanai kegiatan kemanusiaannya. Meskipun pada akhirnya semua diserahkan kembali kepada organisasi kemanusiaan yang disetujui secara hukum, namun kasus tersebut menarik perhatian publik pada kerentanan penyalahgunaan dana *crowdfunding*.

Meskipun saat ini, hampir semua *platform crowdfunding* telah diatur dan diawasi oleh pemerintah, namun sebagai pendukung atau donatur, dapat melakukan uji kelayakan sendiri sebelum memutuskan untuk mendukung sebuah proyek. Misalnya, ketika mengevaluasi sebuah proyek bisnis, dapat dilihat jika dana yang terkumpul tidak mencapai target dalam batas waktu yang ditentukan dan ada risiko kegagalan, donatur juga harus mempertimbangkan risiko ini serta mencari akuntabilitas untuk pemilik proyek sebanyak mungkin kemudian perlu memverifikasi keabsahan organisasi, nomor kontak organisasi, situs web dan alamat *email*, nomor kontak pemilik proyek, dan apakah proyek yang diusulkan cukup layak untuk direalisasikan.

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.

1.2 Rumusan Masalah

Adapun rumusan masalah dari penelitian ini adalah sebagai berikut :

1. Bagaimana model dapat memprediksi kesuksesan sebuah proyek kickstarter dari *platform* penggalangan dana atau *Crowdfunding*
2. Algoritma manakah yang akan memberikan performa terbaik untuk proyek prediksi ini
3. Variabel manakah yang berpengaruh dalam kesuksesan sebuah proyek kickstarter

1.3 Batasan Masalah

Adapun batasan masalah dalam penelitian ini adalah sebagai berikut :

1. Data yang digunakan adalah data yang terkait dengan kickstarter pada tahun 2018 seperti data yang berisi *id*, *name*, *goal*, *deadline*, *status*, jumlah pendukung proyek.
2. Model Prediksi menggunakan beberapa algoritma *machine learning* yaitu, *Artificial Neural Network (ANN)* *backpropagation* dan *random forest* untuk menjadi pembanding model satu dengan yang lain nya.
3. Menggunakan bahasa pemrograman python dan menggunakan google colab untuk *running* program.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut :

1. Menganalisa algoritma yang cocok dalam memprediksi proyek kickstarter.
2. Menganalisa variabel mana yang sangat berpengaruh terhadap kesuksesan proyek kickstarter.
3. Menganalisa parameter model prediksi terbaik dari setiap algoritma yaitu *Random Forest* dan *Backpropagation*

1.5 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah sebagai berikut :

1. Mengetahui algoritma yang baik untuk melakukan prediksi pada kesuksesan proyek kickstarter.
2. Mengetahui variabel yang paling berpengaruh terhadap suksesnya proyek kickstarter.
3. membuktikan bahwa banyaknya estimator (pohon yang terbentuk) mempengaruhi nilai akurasi atau tidak pada algoritma *Random Forest*.
4. Model mampu memprediksi kesuksesan kickstarter dengan akurat dan baik.

1.6 Sistematika Penulisan

Adapun sistematika penulisan untuk penelitian ini adalah sebagai berikut :

1. BAB I : PENDAHULUAN

Pada Bab 1 ini merupakan pendahuluan yang didalamnya menjelaskan tentang latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika penulisan pada penelitian ini.

2. BAB II : TINJAUAN PUSTAKA

Pada Bab II ini merupakan penjabaran dari tinjauan pustaka terkait yang digunakan pada penelitian ini meliputi landasan teori terkait istilah-istilah yang ada dalam laporan penilitan ini seperti algoritma yang digunakan untuk proyek ini.

3. BAB III : METODOLOGI PENELITIAN

Pada Bab III ini berisikan tentang keterangan jadwal penelitian, terkait waktu dan tempat selama melakukan penelitian tersebut serta keterangan alat dan bahan yang digunakan.

4. BAB IV: HASIL DAN PEMBAHASAN

Pada Bab IV ini merupakan penjelasan dari hasil yang sudah didapat selama penelitian dan juga pembahasan secara rinci hasil yang sudah didapat.

5. BAB V : PENUTUP

Pada Bab V ini berisi tentang kesimpulan yang didapat dan juga saran yang digunakan untuk menjadi parameter sebagai penelitian lanjutan yang lebih baik lagi sehingga bisa mendapatkan hasil yang sempurna.

II. TINJAUAN PUSTAKA

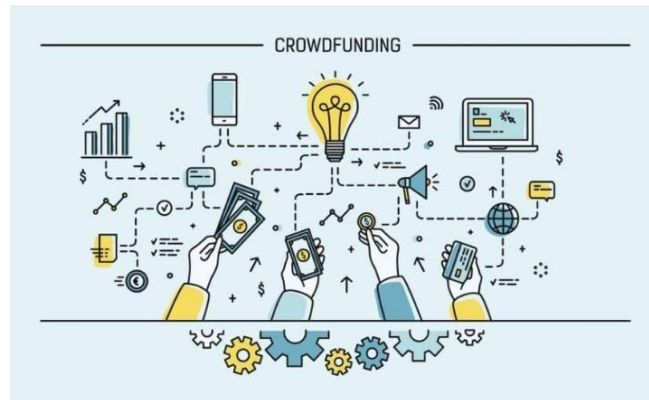
2.1 *Crowdfunding*

Istilah dari *Crowdfunding* sendiri berasal dari kata *crowdsourcing*, yaitu proses untuk mendapatkan *source* (sumber daya) dari *crowd* (kerumunan/orang banyak). Menurut Howe, menyatakan bahwa kerumunan lebih efisien daripada individu. Dalam *crowdfunding*, sumber daya yang dibutuhkan dari luar adalah sumber daya berupa pembiayaan. Kemudian Menurut Lambert & Schwenbacher, menemukan bahwa *crowdfunding* adalah cara untuk mendanai proyek dan bisnis melalui internet. Sumber pendanaan dapat berasal dari berbagai individu yang bersedia menjadi donatur atau yang bersedia mendanai dengan imbalan kepentingan *financial* dan *non-financial*. Menurut Ramsey, *crowdfunding* itu proses menggalang dana untuk mengubah ide menjadi bisnis nyata dengan menghubungkan *fundraiser* yang membutuhkan investasi dan penyandang dana (pendukung/pendukung).

Manfaat dari *Crowdfunding* bagi *fundraiser* adalah mampu memperoleh dana melalui saluran-saluran distribusi yang dapat memenuhi pendanaan, tidak seperti penggalangan dana, pemberi pinjaman memiliki beberapa manfaat untuk berpartisipasi dalam *crowdfunding*. Motivasi utama donatur adalah keterlibatan emosional *funder* dalam proyek yang ingin mereka dukung. Motivasi lainnya adalah untuk mendapatkan *reward*, hal ini juga berlaku untuk *crowdfunding*. Sejak awal, *crowdfunding* ditujukan untuk mendapatkan imbalan, baik imbalan *non-moneter* maupun keuntungan finansial. Selain manfaat yang dapat diperoleh melalui *crowdfunding*, ada beberapa risiko yang mungkin dihadapi baik oleh pencari dana maupun pemberi pinjaman. Risiko utama adalah risiko penipuan oleh penyelenggara atau *platform*, yang dapat mengambil dua bentuk. Bentuk penipuan pertama adalah *platform* menyediakan informasi penipuan tentang situasi keuangan perusahaan yang disembunyikan dari investor, atau dana yang diterima

tidak dialokasikan untuk tujuan yang dimaksudkan. Jenis penipuan ini disebabkan oleh asimetri informasi. Risiko berikut adalah akibat dari pencurian data, pencucian uang dan pendanaan teroris, dan penggunaan internet yang mengancam kejahatan privasi. Risiko lain yang mungkin terjadi adalah kegagalan proyek *crowdfunding* yang berdampak pada imbalan atau *reward* yang dijanjikan untuk *funder*.

Perusahaan dan individu yang meminjam uang tidak dapat membayar kembali dana pinjaman atau membayar *dividen* atau keuntungan dari investasi pemberi pinjaman. Faktor utama yang mengekspos investor/penyandang dana pada risiko penipuan dan kegagalan adalah kurangnya informasi tentang individu atau perusahaan yang mencari pendanaan, hanya terbatas pada informasi yang tersedia di *platform*. Risiko utama bagi pemberi pinjaman, terutama yang terlibat dalam model *crowdfunding* ekuitas, adalah risiko likuiditas, yang membuat sulit untuk menjual kembali investasi ekuitas yang dilakukan melalui *crowdfunding*, tidak seperti investasi ekuitas di pasar saham (Setiawan, Naomi, & Wijanarko, 2020). Pada gambar 2.1 merupakan skema penggalangan dana atau *crowdfunding*.



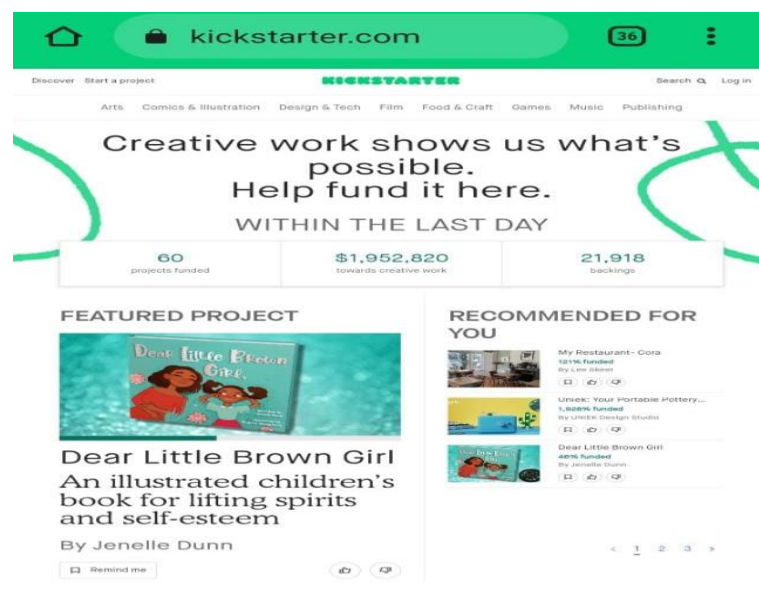
Gambar 2.1 *Crowdfunding*

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. Setiap proyek dibuat secara independent oleh fundraiser yang kemudian *funder* yang mengenal ataupun tidak *fundraiser* dapat menawarkan untuk mendanai proyek itu sendiri dengan imbalan hadiah dan barang jadi. *Fundraiser* dapat menggunakan teks, video, dan foto untuk menyiapkan halaman yang menunjukkan semua detail proyek dan *prototype* untuk memberi tahu *funder*. Pencipta proyek menetapkan tujuan dan tenggat waktu pendanaan, serta berbagai tingkat kompensasi yang dapat diterima penyandang dana ketika *funder* mendonorkan dalam jumlah tertentu (Semakin banyak *funder* menginvestasikan uang, maka semakin besar hadiah atau *reward* yang bisa didapatkan).

Ketika cukup banyak orang mendanai proyek dengan menjanjikan sejumlah uang untuk mencapai tujuan pencipta pada tenggat waktu, proyek dapat dikembangkan dan diproduksi. Bergantung pada kerumitan proyek, pendukung yang menjanjikan uang mungkin harus menunggu berbulan-bulan untuk menerima dan mengakses produk jadi (Travel, 2022). Berikut ini pada gambar 2.2 merupakan tampilan pada halaman utama *platform* kickstarter.



Gambar 2.2 Tampilan halaman kickstarter(<https://www.kickstarter.com/>)

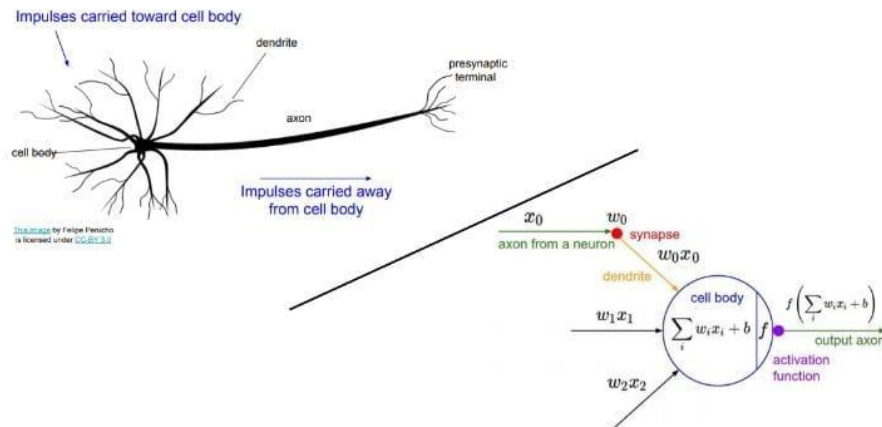
2.3 *Artificial Neural Network (ANN)*

Artificial Neural Network(ANN) adalah sekumpulan algoritma yang bertindak seperti jaringan saraf di otak manusia, dengan *neuron* yang terhubung untuk memproses sebuah informasi (Rajendra, *Artificial Neural Network*, 2022). Tujuan utama JST adalah untuk membekali komputer dengan kemampuan kognitif yang menyerupai otak manusia, memiliki kemampuan pemecahan masalah, dan untuk melakukan proses pembelajaran. Elemen penting dari teknik ini adalah pada struktur sistem pengolahan informasi yang bersifat unik dan beragam untuk setiap aplikasi.

Jaringan saraf terdiri dari sejumlah elemen pemrosesan informasi (*neuron*) yang terhubung dan bekerja sama untuk memecahkan masalah tertentu (umumnya masalah klasifikasi atau prediksi). Fungsionalitas jaringan syaraf tiruan dapat serupa dengan pembelajaran manusia berbasis contoh atau yang disebut pembelajaran terawasi. Jaringan saraf tiruan dikonfigurasi untuk aplikasi tertentu seperti pengenalan pola dan klasifikasi data dan disempurnakan selama proses pembelajaran. Proses pembelajaran yang berlangsung dalam sistem biologis melibatkan penyesuaian koneksi sinaptik yang ada antar *neuron*. Dalam kasus jaringan syaraf, penyesuaian koneksi sinaptik antar *neuron* dilakukan dengan menyesuaikan bobot yang ada pada setiap koneksi *input*, *neuron* dan *output*. Jaringan saraf memproses informasi berdasarkan cara kerja otak manusia. Dalam hal ini, *neural network* terdiri dari sejumlah elemen pemrosesan yang bekerja secara paralel dan terhubung untuk menyelesaikan suatu masalah tertentu.

Pada komputer konvensional menggunakan pendekatan kognitif untuk memecahkan masalah, dimana cara pemecahan masalah haruslah sudah diketahui sebelumnya untuk kemudian dibuat menjadi beberapa instruksi kecil yang terstruktur. Instruksi ini diterjemahkan ke dalam program komputer dan kemudian menjadi kode mesin yang dapat dijalankan oleh komputer. Jaringan saraf memiliki kemampuan yang dapat digunakan untuk memperoleh pengetahuan dari data yang kompleks atau tidak akurat. Ini juga dapat digunakan untuk mengekstraksi dan mengidentifikasi *trend* yang merupakan pola kompleks yang tidak akan

diperhatikan oleh manusia dan teknologi komputasi yang lainnya. Jaringan saraf terlatih dapat dianggap sebagai ahli dalam kategori pemrosesan informasi yang diberikan untuk analisis. Sehingga jaringan syaraf tiruan dapat digunakan untuk membuat prediksi tentang kemungkinan situasi di masa depan dan menjawab pertanyaan "bagaimana jika" (Widiputra, Artificial Neural Network, 2016).



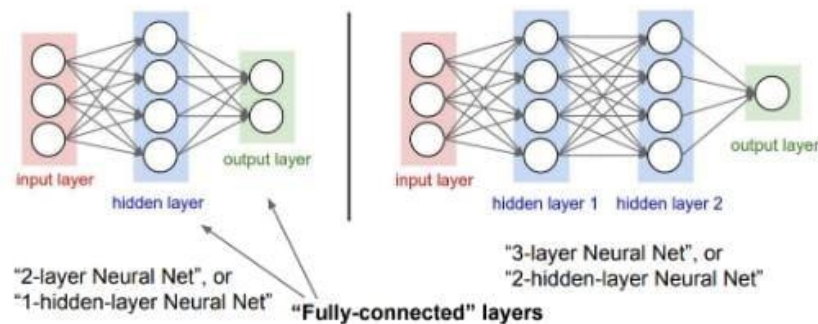
Gambar 2.3 Representasi ANN

Pada Gambar 2.3 merupakan representasi dari *neural network*, yang memiliki tiga input ke *neuron* (x_0, x_1, x_2), setiap *input* dikalikan dengan variabel bernama bobot (w_0, w_1, w_2), kemudian tiga *input* ditambahkan di bagian atas. Setiap koneksi syaraf memiliki bobot sendiri dan nilainya akan berubah selama pelatihan hingga model yang dihasilkan oleh jaringan syaraf tiruan mendekati target keluaran yang diinginkan.

Kemudian bias dapat ditambahkan ke hasil perhitungan *input* dengan bobot. Nilai bias ini tidak berasal dari lapisan *input*. Bias seperti intersep atau penghambat dalam persamaan linier ditambahkan untuk membuat hasil perhitungan lebih akurat. Ketika semua penjumlahan antara bobot dan bias di atas dilakukan, *neuron* akan memasuki fungsi yang disebut fungsi aktivasi. Fungsi aktivasi ini menentukan apakah *neuron* harus aktif atau tidak, beginilah cara kerja jaringan syaraf tiruan, semua *neuron* terhubung bersama untuk menghasilkan *output* dari *input* yang diberikan.

Architectural ANN atau Jaringan syaraf tiruan adalah kumpulan *neuron* yang disusun dalam

- *Input layer*: Lapisan yang memasukkan data ke dalam sistem untuk diproses lebih lanjut di lapisan berikutnya.
- *Hidden layer*: lapisan antara *input* dan *output layer*, di mana *neuron* buatan memiliki satu set *input* berbobot dan prosedur untuk menghasilkan *neuron output* menggunakan fungsi aktivasi.
- *Output layer*: lapisan terakhir dari *neuron* yang menghasilkan *output* atau keluaran sistem.



Gambar 2.4 arsitektur *artificial neural network*

Pada gambar 2.4 merupakan jenis-jenis neural network dan arsitekturnya:

1) *Neural Networks Feed-forward*

Neural network feed-forward adalah jaringan syaraf tiruan yang paling sederhana. Informasi masuk dari lapisan *input* ke lapisan tersembunyi ke lapisan *output* dalam arah maju, tidak berulang melakukan proses *cycle/loop* seperti pada *recurrent neural network*.

a. *Single-layer perceptron*

Jaringan saraf yang paling sederhana adalah *single-layer perceptron*, dengan hanya satu *single-layer output node*. Data *input* dimasukkan langsung menuju ke *neuron* keluaran melalui serangkaian bobot. jumlah semua produk titik(dot) antara variabel *input* dan bobot di setiap koneksi saraf. *Perceptron* dapat diuji dengan cara sederhana menggunakan aturan delta, yang menghitung kesalahan yang dihasilkan

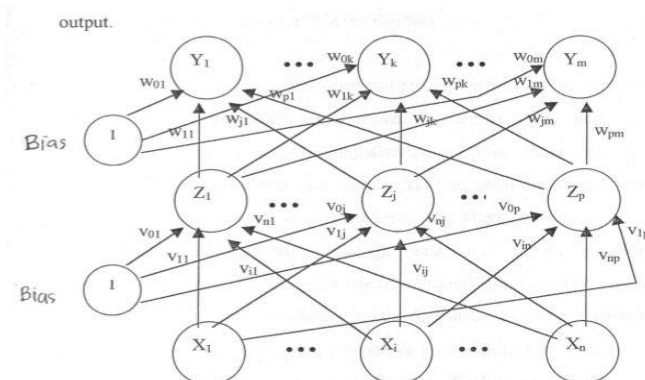
oleh lapisan keluaran terhadap nilai sebenarnya, yang kemudian digunakan untuk memperbaiki bobot atau *weight*.

Perceptron lapisan tunggal hanya dapat digunakan untuk masalah dekomposisi linier, yang berarti bahwa batas keluaran didefinisikan dengan baik dan hanya ada dua kemungkinan.

b. *Multilayer Perceptron*

Multilayer Perceptron terdiri dari beberapa lapisan yang terhubung yang saling terhubung dengan cara *feed forward*, di mana setiap *neuron* dalam satu lapisan terhubung ke semua *neuron* di lapisan berikutnya. Banyak implementasi *Multilayer Perceptron* menggunakan sigmoid sebagai fungsi aktivasi. *Multilayer Perceptron* menggunakan *backpropagation* selama pelatihan menghitung gradien dari fungsi kerugian (*error*) relatif terhadap bobot dari setiap koneksi *neuron* dan meminimalkan kerugian ketika bobot diubah (Yunus, 2020). *Backpropagation* (propagasi balik) merupakan salah satu jaringan syaraf tiruan (*Neural Network*) yang merupakan metode pelatihan terawasi (*Supervised Learning*) dengan jaringan *multilayer* dan memiliki fitur meminimalkan kesalahan pada *output* yang dihasilkan oleh jaringan tersebut. Biasanya dalam proses klasifikasi jaringan syaraf tiruan *backpropagation*.

Klasifikasi ini bekerja dengan melakukan dua langkah komputasi, yaitu komputasi maju yang menghitung nilai kesalahan antara nilai keluaran sistem dan nilai yang diharapkan dan perhitungan mundur untuk mengoreksi bobot sesuai nilai kesalahan (Prasetyo, 2019). Pada gambar 2.5 merupakan alur kerja algoritma *backpropagation*.



Gambar 2.5 *Backpropagation* dengan 1 unit *hidden layer*.

Arsitektur dari algoritma *backpropagation* terdiri dari tiga lapisan, yaitu lapisan *input*, lapisan tersembunyi dan lapisan *output*. Pada lapisan *input*, tidak ada perhitungan, tetapi sinyal pada *input* X dikirim ke lapisan tersembunyi. Pada lapisan tersembunyi dan lapisan keluaran terdapat perhitungan bobot dan bias serta besarnya keluaran lapisan tersembunyi dan lapisan keluaran dihitung berdasarkan *activation function* atau fungsi aktivasi. Pada dasarnya proses dari algoritma *backpropagation* terdiri dari komputasi maju (*forward feed*) dan propagasi balik (*backpropagation*).

Algoritma untuk proses pencarian adalah:

1. Unit pada *input layer* ($X_i, i = 1, 2, \dots, n$) Menerima *input* X_i dan kemudian mengirimkannya ke semua unit di lapisan atas (*Hidden layer*). Kemudian unit pada *hidden layer* ($Z_j, j = 1, 2, \dots, N$), menghitung semua sinyal *input* dengan bobotnya:

$$z_in_j = v_{0j} + \sum X_i v_{ij} \quad \dots\dots\dots(1)$$

Selanjutnya menghitung nilai aktivasi untuk setiap unit pada *hidden layer* sebagai *output unit hidden*:

$$z_j = f(z_in_j) \quad \dots\dots\dots(2)$$

$$f(z_in_j) = \frac{1}{1 + e^{-z_in_j}} \quad \dots\dots\dots(3)$$

Lalu mengirimkan nilai aktivasi sebagai *input* untuk *unit output*. Unit pada *output layer* ($Y_k, k=1, 2, \dots, n$), menghitung semua sinyal masukannya dengan bobotnya:

$$y_in_k = w_{0k} + \sum z_j w_{jk} \quad \dots\dots\dots(4)$$

Selanjutnya menghitung nilai aktivasi pada setiap *unit output* sebagai *output jaringan*(prediksi):

$$y_k = f(y_in_k) \quad \dots\dots\dots(5)$$

$$f(y_in_k) = \frac{1}{1 + e^{-y_in_k}} \quad \dots\dots\dots(6)$$

Setelah itu dilanjutkan dengan algoritma *backpropagation*, dengan proses sebagai berikut:

Unit *output* (Y_k , $k=1,2,\dots,m$), Menerima pola *output* target yang bersesuaian dengan pola *input* kemudian menghitung informasi *error* dengan membandingkan *output* pada target dan prediksi serta mengalikan dengan turunan fungsi aktivasi yang digunakan sebagai berikut :

$$\mathbf{d}_k = (\mathbf{t}_k - \mathbf{y}_k) \mathbf{f}'(\mathbf{y}_{\text{in}k}) \quad \dots\dots\dots(7)$$

Selanjutnya menghitung besarnya koreksi untuk bobot *unit output* :

$$\Delta W_{jk} = \alpha \frac{\partial E(W_{jk})}{\partial W_{jk}} = \alpha \mathbf{d}_k z_j \quad \dots\dots\dots(8)$$

Kemudian menghitung besarnya koreksi untuk bias *output* :

$$\mathbf{Dw}_{0k} = \mathbf{a} \mathbf{d}_k \quad \dots\dots\dots(9)$$

Setelah itu mengirimkan \mathbf{d}_k ke unit-unit yang ada pada *layer* dibawahnya, yaitu ke *hidden layer*. Unit pada *hidden layer* (Z_j , $j=1,2,\dots,p$), menghitung semua koreksi *error* dengan rumus :

$$\mathbf{d}_{\text{in}j} = \sum \mathbf{d}_k w_{jk} \quad \dots\dots\dots(10)$$

Selanjutnya menghitung nilai aktivasi koreksi *error* :

$$\mathbf{d}_j = \mathbf{d}_{\text{in}j} \mathbf{f}'(\mathbf{z}_{\text{in}j}) \quad \dots\dots\dots(11)$$

Kemudian menghitung koreksi bobot pada unit *hidden layer* :

$$\mathbf{Dv}_{ij} = \mathbf{a} \mathbf{d}_j \mathbf{x}_i \quad \dots\dots\dots(12)$$

Dan selanjutnya menghitung koreksi *error* bias pada unit *hidden layer*:

$$\mathbf{Dv}_{0j} = \mathbf{a} \mathbf{d}_j \quad \dots\dots\dots(13)$$

Setelah semua koreksi dihitung dan didapatkan hasilnya, selanjutnya mengupdate bobot dan bias. Untuk unit *output* (Y_k , $k = 1, 2, \dots, m$), mengupdate bobot dan biasnya ($j = 0, \dots, p$) :

$$\begin{aligned} W_{jk}(\text{baru}) &= W_{jk}(\text{lama}) + DW_{jk} \\ W_{0k}(\text{baru}) &= W_{0k}(\text{lama}) + DW_{0k} \end{aligned} \quad \dots\dots\dots(14)$$

untuk unit *hidden layer* (Z_j , $j = 1, \dots, p$), mengupdate bobot dan biasnya ($i = 0, \dots, n$):

$$\begin{aligned} V_{ij}(\text{baru}) &= V_{ij}(\text{lama}) + DV_{ij} \\ V_{0j}(\text{baru}) &= V_{0j}(\text{lama}) + DV_{0j} \end{aligned} \quad \dots\dots\dots(15)$$

Selain terkait hasil dari update bobot dan koreksi *error*, terdapat dua kondisi berhenti pada algoritma *backpropagation* ini, yaitu : jika $Error < Error$ maksimum.

Error adalah perbedaan yang terjadi antara *output* terhadap target yang diinginkan. Proses ANN akan berhenti jika besarnya *error* yang terjadi telah bernilai lebih kecil dari nilai *error* maksimum yang telah ditetapkan. Besarnya nilai *error* dihitung dengan menggunakan fungsi *error* kuadratis.

$$E = 0.5 \sum_k (t_k - Y_k)^2 \quad \dots\dots\dots(16)$$

Dan yang kedua yaitu $Epoch > Epoch$ maksimum. *Epoch* adalah suatu langkah yang dilakukan dalam pembelajaran pada ANN. Jika besarnya *epoch* lebih besar dari besarnya *epoch* maksimum yang telah ditetapkan, maka proses pembelajaran akan berhenti.

Kedua kondisi *stopping* di atas digunakan dengan logika OR. Jadi kondisi *stopping* terjadi jika besarnya $Error < Error$ maksimum atau $Epoch > Epoch$ maksimum (Ginanto, 2012).

Beberapa faktor yang mempengaruhi keberhasilan algoritma *Backpropagation*, antara lain:

1. Inisialisasi Bobot

Bobot awal menentukan apakah jaringan akan mencapai kegagalan lokal atau *global minimal* dan menilai jaringan akan seberapa cepat mengalami konvergen.

2. *learning rate*

learning rate merupakan parameter jaringan dalam mengontrol proses penyesuaian bobot. Nilai *learning rate* yang optimal tergantung pada kasus yang sedang dipertimbangkan. *learning rate* yang terlalu rendah memperlambat konvergensi pada jaringan, sedangkan nilai *learning rate* yang terlalu tinggi dapat menyebabkan ketidakstabilan jaringan.

3. Momentum

Momentum digunakan untuk mempercepat pembentukan atau pelatihan jaringan. Metode momentum melibatkan penyesuaian bobot ditambah koefisien tertentu atas koreksi sebelumnya (Kholis, 2011). Penyesuaian ini dapat dinyatakan sebagai berikut. Bobot yang diperbarui dengan momentum dirumuskan sebagai berikut:

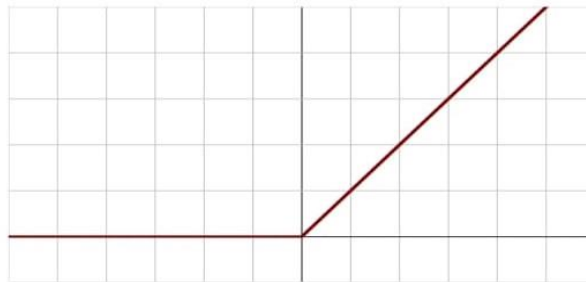
$$\mathbf{w}_{jk}(t+1) = \mathbf{w}_{jk}(t) + \mathbf{Dw}_{jk} + \mathbf{h Dw}_{jk}(t-1) \dots\dots\dots(17)$$

$$\mathbf{v}_{ij}(t+1) = \mathbf{v}_{ij}(t) + \mathbf{Dv}_{ij} + \mathbf{h Dv}_{ij}(t-1) \dots\dots\dots(18)$$

Selain lapisan atau *layer* pada algoritma *backpropagation* di dalam *Artificial Neural Network* juga terdapat fungsi aktivasi. Fungsi aktivasi merupakan fungsi yang digunakan pada jaringan syaraf untuk mengaktifkan atau tidak mengaktifkan *neuron*. Ada banyak jenis fungsi aktivasi pada jaringan syaraf tiruan, tetapi yang umum digunakan hanya 2 fungsi aktivasi yaitu fungsi aktivasi reLu dan juga sigmoid. *Rectified Linear Unit (ReLU) function* memiliki kelebihan dalam *network* yang diinisiasi secara acak, hanya 50% dari lapisan tersembunyi yang akan di aktivasi. *ReLU Function* dirumuskan sebagai berikut,

$$\mathbf{f}(\mathbf{x}) = \max(\mathbf{0}, \mathbf{x}) \dots\dots\dots(19)$$

fungsi aktivasi ReLu digambarkan dalam grafik seperti gambar 2.6 dibawah ini.

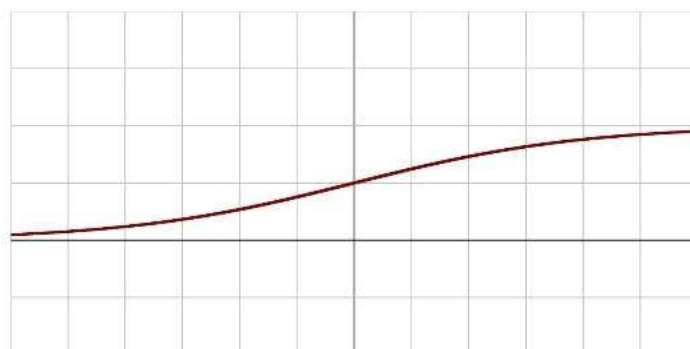


Gambar 2.6 kurva fungsi aktivasi ReLu

Sigmoid function dikenal juga dengan istilah *logistic function*, akan menghasilkan nilai pada rentang $[0, 1]$, dirumuskan sebagai berikut:

$$f(x) = \sigma(x) = \frac{1}{1 + e^{-x}} \quad \dots\dots\dots(20)$$

Pada mulanya fungsi sigmoid tumbuh secara eksponensial, setelah itu mulai terjadi saturasi dan pertumbuhan melambat sehingga menjadi linear sampai pertumbuhan berhenti (Rajendra, *Artificial Neural Network*, 2022). Kurva *Sigmoid function* digambarkan pada gambar 2.7 sebagai berikut:



Gambar 2.7 kurva fungsi aktivasi Sigmoid.

Sebelum melakukan pemodelan menggunakan *algoritma artificial intelligence* data yang sudah dikumpulkan harus dilakukan analisis dengan menggunakan *exploratory data analysis* (EDA) untuk mengenal kualitas dari data yang akan digunakan. *exploratory data analysis* (EDA) adalah bagian dari proses ilmu data.

EDA sangat penting sebelum melakukan rekayasa fitur dan pemodelan karena pada tahapan ini harus terlebih dahulu memahami data. Analisis data eksplorasi memungkinkan untuk memahami isi data yang digunakan, mulai dari distribusi, frekuensi, korelasi, dan lain-lain. Dalam penggunaannya, *curiosity* sangat penting dalam proses ini, pemahaman konteks data juga diperhatikan, karena akan menjawab masalah masalah dasar.

Rekayasa fitur atau *feature engineering* adalah proses memilih, memanipulasi, dan mengubah data mentah menjadi fitur yang dapat digunakan dalam pembelajaran terawasi (*supervised learning*). Ada banyak metode yang digunakan untuk melakukan rekayasa fitur, tetapi pada penelitian ini, hanya *one hot encoder* yang digunakan. *One-hot encoding* adalah metode enkripsi paling populer dalam pembelajaran mesin. Metode ini membagi nilai kolom menjadi beberapa flag kolom dan memberikan nilai 0 atau 1. Metode ini mengubah data kategorikal ke dalam format *numerical* dan memungkinkan kita untuk mengelompokkan data kategorikal tanpa kehilangan informasi apapun (Widiputra, Artificial Neural Network, 2016).

Feature scaling atau penskalaan fitur pada data yang dikodekan, menggunakan skala standar dari scikit-learning. *Scaling* adalah cara agar data *numerical* dalam suatu dataset memiliki rentang nilai (skala) yang sama. Tidak ada lagi satu variabel data yang mendominasi data lainnya (R Kukuh, 2018).

2.4 Stochastic Gradient Descent(SGD)

Gradient descent yang populer digunakan secara *default* pada optimasi pembelajaran mesin adalah *stochastic gradient descent*. Penurunan gradien stokastik adalah proses pembelajaran yang memperbarui setiap level data. Artinya setiap melakukan 1 propagasi maju (*feed forward*) dengan 1 data, maka akan ada 1 propagasi mundur (*Backpropagation*) juga. Jadi jika memiliki banyak n data, maka akan memperbarui bobot sebanyak n kali. Keuntungan menggunakan penurunan gradien stokastik adalah bisa digunakan untuk memperbarui bobot setiap kali satu data muncul. Namun hal ini dapat membuat perubahan pada pembaharuan bobot

menjadi tidak stabil, karena yang akan dioptimalkan adalah error pada keseluruhan data (Novindasari, 2020).

2.5 Random Forest

Random Forest (RF) adalah algoritma yang digunakan untuk mengklasifikasikan sejumlah besar data. Klasifikasi hutan secara acak dilakukan dengan cara menggabungkan pohon (*trees*) dengan melakukan pelatihan pada data sampel yang terorganisir. Menggunakan lebih banyak pohon akan mempengaruhi akurasi yang akan dicapai lebih baik. Penentuan klasifikasi hutan secara acak dilakukan berdasarkan hasil *voting* dari pohon yang terbentuk. Pemenang dari pohon yang terbentuk ditentukan oleh suara terbanyak atau *voting*. Pembangunan pohon di hutan acak hingga mencapai ukuran maksimum pohon data. Namun, membangun pohon hutan acak yang tidak melibatkan pemangkasan adalah metode untuk mengurangi kompleksitas ruang.

Pengembangan dilakukan dengan menerapkan metode pemilihan fitur secara acak untuk meminimalkan kesalahan. Bentuk pohon (*tree*) dengan sampel data menggunakan variabel yang dipilih secara acak dan jalankan *classifier* pada semua pohon yang dihasilkan. Hutan acak menggunakan pohon keputusan (*decision tree*) untuk melakukan proses seleksi. Pohon yang dibangun secara rekursif dipisahkan dari data yang memiliki kelas yang sama. *Split* digunakan untuk membagi data berdasarkan jenis atribut yang digunakan. Dengan menerapkan pohon keputusan saat menentukan pengklasifikasi, pohon yang tidak baik atau buruk akan membuat prediksi acak yang saling bertentangan, oleh karena itu beberapa pohon keputusan akan menghasilkan jawaban yang baik. Hutan acak adalah cara menerapkan metode acak diskriminan untuk klasifikasi. Klasifikasi akan berjalan jika semua pohon telah dilatih. Ketika klasifikasi selesai, inisialisasi dilakukan dengan data sebanyak mungkin berdasarkan nilai presisi.

Keuntungan menggunakan *Random Forest* adalah dapat mengklasifikasikan data dengan atribut yang tidak lengkap, dapat digunakan untuk klasifikasi dan regresi tetapi tidak baik untuk regresi, lebih cocok untuk klasifikasi data dan dapat

digunakan untuk mengolah sampel data yang besar. Proses klasifikasi pada *Random Forest* dimulai dengan memecah data sampel yang ada menjadi pohon keputusan acak. Setelah pohon terbentuk, *polling* atau *voting* akan dilakukan pada setiap lapisan berdasarkan data sampel. Kemudian gabungkan suara setiap kelas dan dapatkan suara terbanyak. Menggunakan *Random Forest* dalam klasifikasi data akan memberikan suara terbaik (Wikipedia, 2021). Sebuah hutan acak adalah kombinasi dari setiap pohon yang baik yang kemudian digabungkan menjadi satu sampel. Hutan Acak tergantung pada nilai vektor acak yang memiliki distribusi yang sama di semua pohon di mana setiap pohon keputusan memiliki kedalaman maksimum.

Berikut adalah karakteristik akurasi di hutan acak.

- Memusatkan *Random Forest*

Terdapat *classifier* $h_1(x)$, $h_2(x)$, . . . , $h_k(x)$ dan dengan *training set* dari distribusi *random vector* Y, X , Berikut fungsi yang terbentuk

$$mg(\mathbf{X}, Y) = av_k I(h_k(\mathbf{X}) = Y) - \max_{j \neq Y} av_k I(h_k(\mathbf{X}) = j). \quad \dots\dots\dots(21)$$

- Fungsi error yang digunakan

$$PE^* = P_{\mathbf{X}, Y}(mg(\mathbf{X}, Y) < 0) \quad \dots\dots\dots(22)$$

- Hasil dari penggabungan fungsi

$$P_{\mathbf{X}, Y}(P_{\Theta}(h(\mathbf{X}, \Theta) = Y) - \max_{j \neq Y} P_{\Theta}(h(\mathbf{X}, \Theta) = j) < 0). \quad \dots\dots\dots(23)$$

Pada hasil tersebut menjelaskan mengapa *Random Forest* tidak *overfit* saat *tree* ditambahkan, tetapi menghasilkan nilai yang terbatas pada error.

1. Kekuatan dan Korelasi

$$PE^* \leq \sum \text{var}(P_{\Theta}(h(\mathbf{X}, \Theta) = Y) - P_{\Theta}(h(\mathbf{X}, \Theta) = j))s_j^2 \dots\dots\dots(24)$$

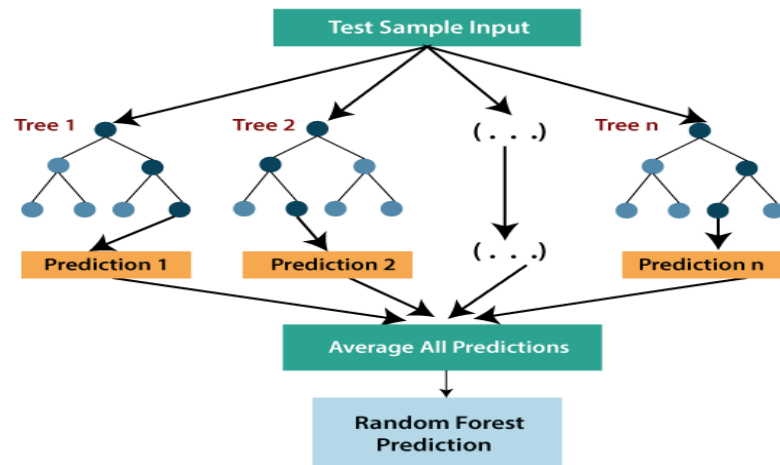
2. *Random Forest* menggunakan pemilihan input acak

Bagging digunakan untuk pemilihan fitur secara acak. Setiap set pelatihan dilakukan bersama dengan penggantinya dari set pelatihan asli. Selanjutnya, pohon ditanam di set pelatihan menggunakan fitur pilihan acak. Ada dua alasan untuk menggunakan *bagging*, yang pertama adalah menggunakan *bagging* untuk meningkatkan akurasi saat menggunakan fungsi acak. *Bagging* kedua digunakan untuk memberikan perkiraan kesalahan keseluruhan (PE^*) dari pohon gabungan, untuk memperkirakan ketahanan dan korelasi. Hutan acak paling sederhana dengan fitur acak dibentuk dengan memilih secara acak, pada setiap simpul, sekelompok kecil variabel input bersama. *training* pohon menggunakan metode CART (*Classification And Regression Tree*) dengan ukuran maksimal.

3. Hutan Acak menggunakan kombinasi input linier

Dengan asumsi ada banyak input, M , F mengambil bagian di atas M akan menyebabkan peningkatan daya tetapi korelasi tinggi. Pendekatan lain adalah mendefinisikan lebih banyak fitur dengan mengambil kombinasi acak linier dari beberapa variabel input. Karakteristik adalah variabel L , yang merupakan jumlah variabel yang digabungkan. Variabel L dipilih secara acak dan ditambahkan ke koefisien dengan bilangan acak $[-1,1]$. Kombinasi linier F dihasilkan. Prosedur ini disebut *Forest-RC* (Yanuar, 2018).

Hutan Acak sangat bagus untuk klasifikasi. Dapat digunakan untuk membuat prediksi kategorikal dengan banyak kemungkinan nilai dan dapat dikalibrasi untuk probabilitas keluaran. Satu hal yang harus diwaspadai adalah pakiaian yang berlebihan. Hutan acak rentan terhadap *overfitting*, terutama ketika bekerja dengan kumpulan data yang relatif kecil. Diragukan apakah model data dapat memberikan prediksi yang “terlalu baik” pada test set menggunakan *Random Forest*. Salah satu cara untuk melakukan *overfit* adalah dengan menggunakan fitur yang sebenarnya relevan dengan model data yang digunakan.



Gambar 2.8 diagram *Random Forest*

Berdasarkan gambar 2.8 merupakan diagram alur kerja dari algoritma *Random Forest*. *Random Forest* bekerja dengan membangun beberapa pohon keputusan dan menggabungkannya untuk mendapatkan prediksi yang lebih stabil dan akurat. "Hutan" yang dibangun oleh *Random Forest* adalah kumpulan pohon keputusan yang biasanya dilatih dengan bagging. Ide umum *bagging* adalah menggabungkan model pembelajaran untuk meningkatkan hasil secara keseluruhan. Algoritma Hutan Acak meningkatkan keacakan model saat pohon tumbuh. Alih-alih menemukan fitur yang paling penting saat memisahkan sebuah *node*, *Random Forest* mencari fitur terbaik di antara subset fitur acak. Akibatnya, metode ini menghasilkan varietas kultivar yang lebih luas dan umumnya memberikan model yang lebih baik (Algoritma Data Science Academy, 2022).

Pengoperasian hutan acak dimulai dengan pembangunan banyak pohon keputusan sesuai yang ditentukan. Banyak algoritma yang digunakan untuk membangun pohon keputusan, misalnya menggunakan algoritma C4.5. Ada banyak algoritma yang dapat digunakan dalam pelatihan pohon keputusan, antara lain: ID3, CART dan C4.5. Algoritma C4.5 merupakan evolusi dari algoritma ID3, proses pohon keputusan adalah mengubah bentuk data (tabel) menjadi model pohon, mengubah model pohon menjadi aturan, dan menyederhanakan aturan.

Secara umum, algoritma C4.5 untuk membangun pohon keputusan adalah sebagai berikut: pilih satu atribut sebagai *root*, buat cabang untuk setiap nilai, bagi kasus di

cabang, dan ulangi proses untuk setiap cabang hingga semua instance masuk atau semua kasus memiliki cabang memiliki kelas yang sama.

Proses mencari akar atau *root node* adalah sebagai berikut:

- Mencari nilai varians dari setiap kelas pada atribut dan total target.
- Kemudian hitung varians menggunakan entropi, dengan rumus

$$Entropy(S) = \sum_{i=1}^n - p_i * \log_2 p_i \quad \dots\dots\dots(25)$$

- Kemudian cari *weighting average* dari setiap atribut, lakukan perhitungan entropi total – *weighting average*, Langkah ini disebut dengan *information gain*. Dengan rumus sebagai berikut:

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} * Entropy(S_i) \quad \dots\dots\dots(26)$$

- Atribut yang memiliki *information gain* tertinggi, maka layak menjadi *root* (Saputra, 2021).

Pada *random forest*, untuk membangun *tree* atau pohon selain menggunakan *information gain* bisa menggunakan rumus *gini split indeks* untuk menentukan *root node*.

$$Gini = 1 - \sum_{i=1}^c (p_i)^2 \quad \dots\dots\dots(27)$$

Kelebihan dari algoritma ini adalah menghasilkan *error* yang lebih rendah, Memberikan hasil yang bagus dalam klasifikasi, Dapat mengatasi data *training* dalam jumlah sangat besar secara efisien, Metode yang efektif untuk mengestimasi hilangnya data, dan dapat memperkirakan variabel apa yang penting dalam klasifikasi, Menyediakan metode eksperimental untuk mendeteksi interaksi variabel. Sedangkan selain kelebihan pada algoritma ini juga ada kekurangannya yaitu, Waktu pemrosesan yang lama karena menggunakan data yang banyak dan membangun model *tree* yang banyak pula untuk membentuk *random trees* karena

menggunakan single processor, Interpretasi yang sulit dan membutuhkan mode penyetalan yang tepat untuk data, ketika digunakan untuk regresi, mereka tidak dapat memprediksi di luar kisaran dalam data percobaan, hal ini di mungkinkan data terlalu cocok dengan kumpulan data pengganggu (noisy) (Plaosan, 2020).

2.6 Confussion Matrix

Confusion matrix atau Matriks kebingungan juga dikenal sebagai matriks kesalahan. Pada dasarnya, confusion matrix memberikan informasi tentang perbandingan hasil klasifikasi yang dilakukan oleh sistem (model) dengan hasil klasifikasi yang sebenarnya. Confusion matrix dalam bentuk tabel menggambarkan kinerja model klasifikasi pada serangkaian data uji dengan nilai nyata yang diketahui. Ada empat istilah yang mewakili hasil proses klasifikasi dalam matriks konfusi. Keempat istilah yang merepresentasikan hasil proses klasifikasi tersebut adalah true positive (TP), true negative (TN), false positive (FP) dan false negative (FN). Confusion matrix dapat digunakan untuk mengukur kinerja dalam masalah klasifikasi biner dan masalah klasifikasi multiclass. Klasifikasi biner hanya menghasilkan dua kelas keluaran (label), seperti "Ya" atau "Tidak", "0" atau "1" untuk setiap data masukan yang diberikan. Lapisan utama sering disebut sebagai data positif dan lapisan lainnya disebut data negatif.

Pencarian ini dapat menggunakan matriks konfusi untuk menghitung berbagai metrik kinerja untuk mengukur kinerja model yang dihasilkan. Di bagian ini, mari kita pahami beberapa metrik kinerja yang populer dan umum digunakan: akurasi, presisi, dan recall (Nugroho, 2019). Menurut definisi, presisi adalah rasio nilai positif sejati (TP) dengan jumlah data yang diprediksi positif. Atau bisa juga ditulis dalam matematika:

$$precision = \frac{TP}{TP + FP} \dots\dots\dots(28)$$

Sedangkan untuk *Recall*, secara definisi adalah perbandingan antara *True Positive* (TP) dengan banyaknya data yang sebenarnya positif.

Dan dapat dituliskan secara matematis seperti ini:

$$recall = \frac{TP}{TP + FN} \dots\dots\dots(29)$$

jika melihat dua persamaan di atas dan membandingkannya. dapat dilihat, ada perbedaan antara keduanya, presisi memiliki variabel positif palsu (FP) sedangkan panggilan balik atau *recall* memiliki variabel negatif palsu (FN). Dan juga dari persamaan diketahui bahwa semakin kecil *false positive* (FP) semakin besar akurasi. Mengenai *recall*, semakin kecil *false negative* (FN), semakin besar *recall*. Di dalamnya, disebutkan sebelumnya bahwa *False Positive* (FP) dan *False Negative* (FN) adalah dua jenis kesalahan yang dibahas di atas. Untuk *F1-Score* adalah rata-rata yang diselaraskan antara presisi dan *recall*. Dapat ditulis secara matematis seperti ini:

$$\frac{1}{F1} = \frac{1}{2} \left(\frac{1}{precision} + \frac{1}{recall} \right) \dots\dots\dots(30)$$

Nilai terbaik F1-Score adalah 1.0 dan nilai terburuknya adalah 0. Secara representasi, jika F1-Score punya skor yang baik mengindikasikan bahwa model klasifikasi kita punya *precision* dan *recall* yang baik (Stevanus, 2020).

2.7 Metric Regression

metrik yang disajikan untuk model regresi dirancang untuk memperkirakan jumlah kesalahan. Model dianggap fit dan baik dengan data jika selisih antara nilai yang diamati dan nilai prediksi kecil. Namun, melihat model residual (perbedaan antara satu titik prediksi dan nilai sebenarnya yang sesuai) dapat memberi tahu banyak sekali terkait tentang potensi bias model. Beberapa ukuran berbeda telah dikembangkan oleh ahli statistik untuk mengevaluasi kinerja model regresi. Setiap metrik yang dijelaskan di bawah ini mengukur kesalahan terkait model menggunakan metode yang berbeda.

- Mean Absolute Error (MAE)

Mean Absolute Error (MAE) mengacu pada rata-rata nilai kesalahan absolut yang dihitung untuk setiap titik dalam kumpulan data. MAE dapat dihitung menggunakan rumus berikut:

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i| \quad \dots\dots\dots(31)$$

Kesalahan absolut dihitung untuk setiap pasangan nilai prediksi dan aktual dengan mengambil nilai absolut dari selisih antara kedua nilai. Kemudian, istilah kesalahan absolut yang dihitung untuk setiap pasangan nilai dijumlahkan dan jumlah hasil dibagi dengan jumlah total pengamatan. Nilai yang dihasilkan mewakili kesalahan absolut rata-rata atau jarak vertikal rata-rata antara setiap pasangan nilai yang diprediksi dan aktual saat dibuat grafik.

Menafsirkan MAE itu mudah karena direpresentasikan dalam satuan yang sama dengan data asli kita. Model yang sempurna menghasilkan nol MAE, dan semakin dekat MAE yang diamati ke nol, semakin baik model tersebut cocok dengan data. Perhitungan MAE memperlakukan semua penalti secara sama, apakah nilai prediksi lebih rendah atau lebih tinggi dari nilai sebenarnya.

- Mean Squared Error (MSE)

Mean Squared Error (MSE) mengacu pada rata-rata nilai kesalahan kuadrat yang dihitung untuk setiap titik data.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad \dots\dots\dots(32)$$

Persamaan ini terlihat seperti yang kita lihat sebelumnya untuk MAE. Mengkuadratkan setiap istilah bias menjamin bahwa MSE akan lebih besar dari atau sama dengan nol. Karena setiap nilai deviasi dikuadratkan sebelum jumlah semua suku diambil, dampak dari satu kesalahan yang diamati secara eksponensial lebih besar pada nilai kesalahan total daripada efek dari kesalahan yang diamati lebih kecil. Semakin tinggi nilai kesalahan yang diamati, semakin besar penalti yang diterapkan saat menghitung MSE. MSE umumnya merupakan metrik

penilaian regresi yang lebih populer daripada MAE. MSE lebih menekankan pada kesalahan besar yang diinginkan, karena kami ingin membuat model yang menggeneralisasi dengan baik dan menghasilkan prediksi dengan kesalahan rendah pada kumpulan data. Seperti MAE, nilai MSE mendekati nol menunjukkan kinerja model yang lebih baik. MSE kurang umum digunakan untuk interpretasi manusia karena tidak direpresentasikan dalam satuan yang mudah dipahami, tetapi sangat populer untuk digunakan dalam pengoptimalan pembelajaran mesin.

- *Root Mean Squared Error*(RMSE)

Root Mean Squared Error (RMSE) adalah salah satu metrik evaluasi paling populer untuk masalah regresi. RMSE dihitung dengan mengambil akar kuadrat dari MSE:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \dots\dots\dots(33)$$

Nilai yang didapatkan dapat diinterpretasikan dalam sebuah unit yang sama dengan nilai yang dicoba saat melakukan prediksi, yang membuatnya lebih mudah dipahami daripada beberapa metrik lainnya. Namun, hal penting untuk diingat bahwa nilai RMSE hanya dapat dibandingkan antara model pengukuran kesalahan menggunakan unit yang sama.

Seperti parameter evaluasi lain yang sudah dijelaskan seperti MAE dan MSE, nilai RMSE yang lebih rendah menunjukkan kinerja model yang lebih baik. RMSE harus digunakan pada MAE atau ukuran lain dalam situasi di mana data observasional memiliki distribusi bias bersyarat. Jika MAE digunakan sebagai fungsi tujuan yang diminimalkan saat melatih model pembelajaran mesin, maka MAE akan menghasilkan kecocokan bias yang lebih dekat dengan rata-rata daripada yang diperoleh dengan menggunakan RMSE.

- R^2

Nilai R^2 (juga dikenal sebagai koefisien determinasi) dapat digunakan untuk mengukur seberapa dekat nilai data yang diketahui dengan garis regresi yang sesuai. Nilai R^2 biasanya antara 0,0 dan 1,0. Nilai R^2 yang mendekati 1,0

menunjukkan kecocokan model yang lebih baik. Beberapa metode dapat digunakan untuk menghitung koefisien determinasi. Metode umum untuk menghitung kuantitas ini memerlukan perhitungan dua kuantitas tambahan terlebih dahulu yaitu jumlah sisa kuadrat (RSS) dan jumlah total kuadrat (TSS).

RSS dihitung hanya dengan menambahkan kuadrat dari semua residu yang diamati antara nilai prediksi dan nilai aktual:

$$RSS = \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad \dots\dots\dots(34)$$

Sedangkan untuk TSS dihitung dengan mengambil selisih antara setiap nilai yang diamati dan rata-rata dari semua nilai yang diamati, mengkuadratkan selisih, dan menjumlahkan selisih kuadrat:

$$TSS = \sum_{i=1}^n (y_i - \bar{y})^2 \quad \dots\dots\dots(35)$$

Nilai RSS dan TSS yang dihitung kemudian dapat digunakan untuk menghitung R^2 menggunakan persamaan berikut:

$$R^2 = 1 - \frac{RSS}{TSS} \quad \dots\dots\dots(36)$$

R^2 dapat dipahami sebagai rasio varians yang dijelaskan oleh model. Nilai $R^2 = 1,0$ menunjukkan kecocokan sempurna, yang berarti bahwa model memprediksi dengan benar nilai sebenarnya tanpa kesalahan. Dalam keadaan ini, model menjelaskan 100% varians data di sekitar meannya. $R^2 = 0,0$ menunjukkan bahwa model menghasilkan garis horizontal yang sempurna. Dalam hal ini, model menjelaskan 0% dari varians data sekitarnya (OnnoWiki, 2020).

2.8 Kaggle

Menurut InfoWorld Kaggle merupakan sebuah komunitas *online* yang dibentuk oleh Anthony Goldbloom dan Ben Hamner sebagai CEO & CTO di tahun 2010. Pada komunitas *online* ini berisikan para pegiat *data science* yang ingin belajar lebih dalam tentang *machine learning* dan ilmu-ilmu data lainnya. Di dalamnya, ada berbagai kegiatan yang dilakukan, salah satunya yang paling dikenal dan populer adalah kompetisi *machine learning*. Kaggle sendiri menyatakan, bahwa selain berkompetisi anggota komunitasnya bisa bersama-sama menulis dan membagikan kode serta mempelajari berbagai hal. Bahkan, para *data scientist* bisa juga mendapatkan uang dari proyek yang ditawarkan di Kaggle. Saat ini, Kaggle sendiri telah memiliki 170.000 *post* di forum, dan juga lebih dari 1000 *dataset* serta paling tidak ada 250 *kernel*. Google bahkan telah mengakuisisinya pada tahun 2017 (Rahmalia, 2021).

2.9 Penelitian Terkait

Dalam penyusunan skripsi ini, sedikit banyak terinspirasi dan mereferensi dari penelitian-penelitian sebelumnya yang berkaitan dengan latar belakang masalah pada skripsi ini. Berikut ini beberapa penelitian terdahulu yang berhubungan dengan skripsi ini antara lain:

- Penelitian yang dilakukan oleh Indira Tantina, Budi Hartono, S.T., M.PM., Ph.D., 2019, “Pengembangan model prediksi tingkat kesuksesan *campaign* pada portal *reward-based Crowdfunding*” (Tantina & Hartono, 2019). Menggunakan pendekatan klasifikasi data mining dan algoritma pohon keputusan untuk model prediksi dengan kategori tingkat kesuksesan dibagi menjadi 3, yaitu sukses, tidak sukses, sangat sukses yang akan digunakan sebagai output dari model prediksi ini.
- Penelitian yang dilakukan oleh Nugraha Muharfandy dan Alva Edy Tontowi, 2019, “Kajian awal analisis atribut yang berpotensi mempengaruhi tingkat pendanaan pada proyek 3D printer di *platform Crowdfunding kickstarter*” (Nugraha

& Alva, 2019). Menggunakan pendekatan klasifikasi algoritma *Random Forest* dan *attribute weight* dalam penelitian ini untuk menentukan atribut yang berpotensi memiliki pengaruh besar terhadap level pendanaan. Yang mana hasilnya akan digunakan untuk menjadi acuan bagi *creator* untuk meningkatkan potensi dan peluang mendapatkan perolehan donasi pada *platform Crowdfunding*.

- Penelitian yang dilakukan oleh Gofarna Sayagiri, 2021, “ Analisis faktor-faktor yang memengaruhi keberhasilan *Crowdfunding* melalui kitabisa.com” (Sayagiri, 2021). Penelitian ini menggunakan metode kualitatif deskriptif yang berisi pemaparan fenomena atau keterkaitan sesuatu dengan detail tanpa menguji hipotesis ataupun prediksi. Yang mana hasil yang ditemukan bahwa factor-faktor tersebut berasal dari internal maupun eksternal penggalang dana, seperti jumlah share yang kemudian mengarah pada kepercayaan publik yang bersumber dari transparansi, kredibilitas kemudahan akses sehingga kampanye dapat berjalan sesuai yang dikehendaki.

- Penelitian yang dilakukan oleh Annisa Lutfi Nur Afifah, 2020, “Analisis Prediksi dan Faktor Pendukung Kesuksesan *Crowdfunding* Berbasis Donasi Menggunakan Algoritma C4.5 dan K-Nearest Neighbor (Studi Kasus: Kitabisa.com)” (Nur, 2020). Tujuan dari penelitian ini adalah untuk mengetahui faktor yang mempengaruhi donasi di Kita Bisa, mengetahui perbandingan donasi sebelum dan sesudah perbaikan, dan mengetahui kinerja algoritma yang digunakan dalam memprediksi. Metode yang digunakan adalah klasifikasi dengan algoritma C4.5 dan K-Nearest Neighbor (KNN).

- Penelitian yang dilakukan oleh Sidhart Jhaveri dkk, 2019, “*Success Prediction using Random Forest, CatBoost, XGBoost and AdaBoost for Kickstarter Campaigns*” (Jhaveri, Khedkar, Kantharia, & Jaswal, 2019). Makalah ini bertujuan untuk memprediksi keberhasilan atau kegagalan kampanye Kickstarter dan membantu pemilik kampanye merencanakan kampanye mereka dengan lebih baik dengan memberikan *trend* dan analisis berdasarkan data historis kampanye di Kickstarter, mulai dari 2014 hingga Februari 2019. Berbagai klasifikasi dan algoritma *boosting* telah diterapkan dan disimpulkan bahwa algoritma *Weighted*

Random Forest dengan *AdaBoost* untuk dataset subsampel memberikan akurasi terbaik.

- Penelitian selanjutnya dilakukan oleh Pi-Fen Yu dkk, 2018, “*Prediction of Crowdfunding Project Success with Deep Learning*” (Yu, Huang, Yang, Liu, & Li, 2018). Tujuan dari penelitian ini adalah untuk mengembangkan model yang memprediksi keberhasilan proyek *Crowdfunding* dengan pembelajaran yang mendalam. Kumpulan data dikumpulkan secara retrospektif dari Kaggle dan berisi catatan sejarah kampanye Kickstarter. Model ini dapat memberikan wawasan pada tahap awal penggalangan dana. Eksperimen komprehensif dilakukan dan berbagai algoritma klasifikasi telah diuji untuk mendukung mesin prediksi ini dan disimpulkan bahwa model MLP memiliki hasil terbaik dengan tingkat kepercayaan tertinggi.

- Penelitian ini dilakukan oleh Annisyah Paradhita Sari, 2019, “*Pengaruh Brand Awareness, Kualitas Proyek Dan Kepercayaan Terhadap Keputusan Berdonasi Secara Online Pada Platform Crowdfunding KitaBisa.Com*” (Sarri, 2019). Penelitian ini bertujuan untuk mengetahui seberapa besar pengaruh brand awareness, kualitas proyek, dan kepercayaan terhadap keputusan berdonasi di Kitabisa.com. Pendekatan penelitian yang digunakan adalah pendekatan penelitian kuantitatif. Hasil penelitian ini menunjukkan bahwa secara simultan, variabel brand awareness, kualitas proyek, dan kepercayaan berpengaruh signifikan terhadap keputusan berdonasi di Kitabisa.com dan hasil uji secara parsial menunjukkan bahwa semua variabel bebas yaitu brand awareness, kualitas proyek, dan kepercayaan masing-masing berpengaruh signifikan terhadap keputusan berdonasi di Kitabisa.com.

- Penelitian ini dilakukan oleh Fahad Sarfraz Ahmad dkk, 2017, “*Predicting Crowdfunding success with optimally weighted Random Forests*” (Ahmad, Tyagi, & Kaur, 2017). Dalam penelitian ini, mengusulkan algoritma baru untuk model pembelajaran *Random Forests* dimana menetapkan bobot optimal untuk mengklasifikasi individu dan melakukan pemungutan suara mayoritas tertimbang

pada 26 ribu proyek kickstarter yang digunakan untuk memahami faktor-faktor yang paling mempengaruhi peluang memperoleh pendanaan yang sukses.

- penelitian ini ditulis oleh Wei Wang dkk, 2020, “*Prediction of fundraising outcomes for crowdfunding projects based on deep learning: a multimodel comparative study*” (Wang & Wu, 2020). Tujuan dari penelitian ini untuk mengurangi risiko partisipasi melalui metode otomatis, Pertama, berdasarkan data dari Kickstarter, dilakukan analisis preprocessing dan exploratory. Kemudian, menggunakan pembelajaran mendalam (*multilayer perceptron*) dan menerapkannya pada prediksi kinerja pembiayaan *Crowdfunding*. Dan membandingkan pembelajaran mendalam dengan algoritma pembelajaran mesin lain yang umum digunakan, termasuk pohon keputusan, hutan acak, regresi logistik, mesin vektor dukungan, dan algoritma K-nearest neighbor.

- Penelitian ini dilakukan oleh Michael J. Ryoba dkk, 2020, “*Feature subset selection for predicting the success of crowdfunding project campaigns*” (Ryoba & Zhou, 2020). Statistik dari *platform Crowdfunding* menunjukkan bahwa sebagian kecil proyek *Crowdfunding* berhasil mengamankan dana. Hal ini membuat penulis ingin mengetahui kemungkinan keberhasilan kampanye mereka dan fitur yang berkontribusi terhadap keberhasilannya sebelum meluncurkannya di *platform Crowdfunding*. Eksperimen dilakukan menggunakan WOA(*whale optimization algorithm*) dengan classifier K-Nearest Neighbor (KNN) pada dataset Kickstarter.

III. METODOLOGI PENELITIAN

3.1 Waktu dan Tempat

Penelitian ini dilaksanakan dari akhir bulan januari 2022 sampai dengan pertengahan juni 2022 bertempat di Laboratorium Komputer Jurusan Teknik Elektronika Universitas Lampung.

3.2 Jadwal Penelitian

Berikut ini merupakan jadwal kegiatan yang sudah dilakukan pada penelitian ini:

Tabel 3.1. Jadwal Penelitian:

No.	Kegiatan	Bulan																		Keterangan	
		Minggu ke-																			
		Januari				Februari				Maret				April				Mei			
4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2			
1.	Rumusan Masalah																			Perumusan masalah yang akan diteliti.	
2.	Studi Literatur																			Review jurnal literasi yang terkait.	
3.	Pengumpulan Data																			Collecting data dengan mencari referensi data dari kaggle.	
4.	Exploratory																			Melakukan pengenalan	

3.3 Alat dan Bahan

Alat dan bahan yang digunakan dalam penelitian ini adalah sebagai berikut:

Tabel 3.2. Alat Penelitian

No.	Perangkat	Spesifikasi	Deskripsi
1.	Laptop	Hp Probook intel core i5, RAM 4GB, hdd 500GB dengan sistem operasi windows 10	Perangkat keras yang digunakan untuk mengerjakan penelitian seperti membuat model dan juga menulis skripsi.
2.	Google Colab		Alat untuk menulis program model <i>machine learning</i> untuk penelitian.
3.	Kaggle		<i>Platform</i> yang berisikan kumpulan data untuk model <i>machine learning</i> .
4.	Ms. Word	Microsoft office professional plus 2016	Perangkat yang digunakan untuk pembuatan laporan.
5.	Flowdia Lite		Aplikasi yang digunakan untuk membuat diagram.

Tabel 3.3. Bahan Penelitian

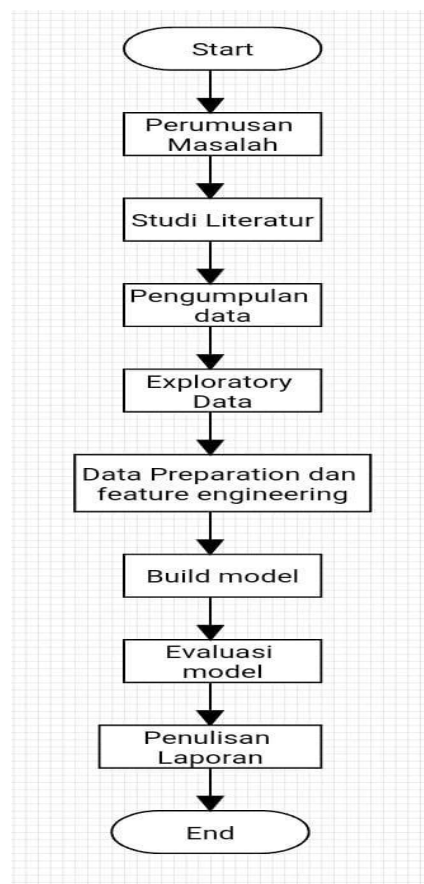
No.	Bahan	Spesifikasi	Deskripsi
1.	Dataset	Kumpulan data yang berisi tentang proyek kickstarter.	Data yang digunakan untuk menjadi bahan pelatihan untuk model prediksi ini.
2.	Jurnal	Jurnal terkait kickstarter dan <i>Crowdfunding</i> .	Digunakan untuk menjadi referensi untuk penelitian.

3.4 Metodologi Penelitian

3.4.1 *Flowchart* Penelitian

Diagram alir atau *flowchart* yang menggambarkan tahapan-tahapan yang dilakukan pada penelitian ini. Dimulai dari perumusan masalah yang menjadi landasan penelitian ini sampai mendapatkan hasil dan bisa ditulis dalam bentuk laporan.

Pada penelitian ini selain merumuskan masalah, juga melakukan riset literasi yang terkait dengan penelitian serta mengumpulkan data dan persiapan data sebelum melakukan pemodelan dan proses pelatihan dan setelah proses pemodelan dan pelatihan selesai maka bisa dilakukan evaluasi pada model yang mana pada proses ini akan dilihat apakah model sudah memiliki performa yang baik dan sesuai dengan yang diinginkan. Setelah semua penelitian selesai maka diakhiri dengan menulis laporan yang didalamnya terdapat hasil penilitan yang akan dibahas dan dijelaskan secara rinci. Gambar 3.1 dibawah berikut ini merupakan diagram alur atau flowchart tahapan penelitian.



Gambar 3.1 *flowchart* penelitian

3.4.2 Tahapan Penelitian

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

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.

2. Studi Literatur

Pada tahapan ini dilakukan dengan mengumpulkan jurnal-jurnal yang terkait, kemudian catat hal apa saja yang nantinya akan menjadi contoh atau menjadi gambaran dalam mengerjakan penelitian ini. Terdapat 10 sampai 15 jurnal terkait dan materi-materi terkait seperti *Crowdfunding*, kickstarter, *Artificial Neural Network*, algoritma *Backpropagation*, *stochastic gradient descent*, *Random Forest*, *confussion matrix*, *matrix regression*, kaggle yang didapat untuk dijadikan literasi dalam pelaksanaan penelitian ini. Materi-materi literasi ini bisa dilihat detail di bab 2 tinjauan Pustaka.

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:

Tabel 3.4. Variabel yang digunakan

No	Nama Variabel	Deskripsi
1.	<i>ID</i>	ID dari masing-masing proyek kickstarter
2.	<i>Name</i>	Nama dari sebuah proyek
3.	<i>Category</i>	Kategori yang lebih detail

4.	<i>Main_category</i>	Kategori untuk kampanyenya secara umum
5.	<i>Currency</i>	Mata uang yang digunakan untuk memberikan dukungan finansial
6.	<i>Deadline</i>	Batas waktu untuk pengumpulan dana
7.	<i>Goal</i>	Jumlah dana yang ingin dikumpulkan, yang dibutuhkan oleh para pembuat untuk menyelesaikan proyeknya
8.	<i>Launched</i>	Tanggal proyek tersebut diluncurkan
9.	<i>Pledged</i>	Jumlah dana yang sudah dikumpulkan
10.	<i>State</i>	Status atau kondisi proyek pada saat ini
11.	<i>Backers</i>	Jumlah orang yang mendukung proyek ini.
12.	<i>Usd_goal_real</i>	Jumlah uang yang ingin dikumpulkan dalam satuan dolar AS yang dikonversi menggunakan API fixer.io
13.	<i>Usd_pledged</i>	Jumlah uang yang telah dikumpulkan dalam satuan dolar AS yang dikonversi dalam situs KS.
14.	<i>Usd_pledged_real</i>	Jumlah uang yang telah dikumpulkan dalam satuan dolar AS yang dikonversi menggunakan API fixer.io
15.	<i>Country</i>	Negara Asal pembuat proyek.

4. *Exploratory Data*

Tahapan selanjutnya adalah *exploratory data analysis*. Pada tahapan ini 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. Adapun grafik dan diagramnya dapat dilihat di lampiran.

5. *Data Preparation & Feature Engineering*

Tahapan ini merupakan lanjutan dari pengumpulan data dan EDA, 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 adalah menghapus varibel yang tidak diperlukan dalam penelitian ini yaitu variable *id*, *name* karena variabel tersebut tidak diperlukan untuk melakukan penelitian ini, lalu selain itu ada variabel *usd_pledge*, *usd_pledge_real* dan *usd_goal_real* yang dihapus karena

memiliki nilai *null* atau kosong yang bisa menyebabkan model memiliki performa tidak maksimal.

Kemudian melakukan *feature engineering* dengan teknik *one-hot encoder* pada variabel yang akan digunakan yaitu proses membuat kolom baru dari variabel kategorikal dimana setiap kategori menjadi kolom baru dengan nilai 0 atau 1 alasannya dikarenakan model prediksi yang akan dilakukan hanya menerima nilai *numerical* bukan nilai kategorik, selanjutnya preparasi data dengan menerapkan *feature scaling* pada variabel yang sudah diubah menjadi data *numerical* menggunakan standar *scaler* dari *scikit-learn* untuk membuat *numerical* data yang ada pada dataset kita memiliki rentang nilai yang sama dan tidak terlalu jauh sehingga tidak ada satu variabel data yang mendominasi variabel data lainnya. Setelah semua tahapan persiapan data selesai maka data siap untuk dilakukan pelatihan model. Tahapan data *Preparation* dan *feature engineering* dapat dilihat pada lampiran.

6. Build Model

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

- a. menggunakan teknik jaringan syaraf tiruan atau ANN.

```

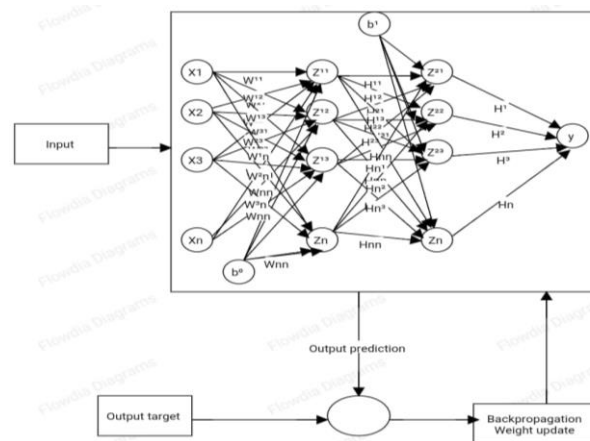
outputs = tf.keras.layers.Dense(1, activation='sigmoid')(x)
model = tf.keras.Model(inputs, outputs)
model.summary()

Model: "model_21"
-----
Layer (type)                 Output Shape         Param #
-----
input_22 (InputLayer)        [(None, 218)]        0
dense_50 (Dense)              (None, 128)          28032
dense_51 (Dense)              (None, 4)             516
dense_52 (Dense)              (None, 1)             5
-----
Total params: 28,553
Trainable params: 28,553
Non-trainable params: 0

```

Gambar 3.2 struktur model multi layer pada Algoritma ANN

Berdasarkan gambar 3.2 diatas yang merupakan struktur model dengan algoritma *Artificial Neural Network* yang didalamnya terdapat beberapa *layer*(lapisan), pada *layer input* ada 9 data dari 10 fitur yang sudah di *encoding* sehingga menjadi 218 data yang digunakan untuk menjadi *input* model dan juga menggunakan fungsi aktivasi relu, kemudian untuk lapisan *output* 1 data yang menjadi label atau target prediksi. Pada model ini menggunakan *multilayer* perceptron yaitu *input*, *hidden* dan *output layer* yang masing-masing *hidden* dan *output layer* juga terdapat fungsi aktivasi didalamnya yaitu fungsi aktivasi relu pada *hidden layer* dan aktivasi sigmoid pada *output layer*.



Gambar 3.3 blok diagram alur algoritma ANN

Berdasarkan gambar 3.3 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 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

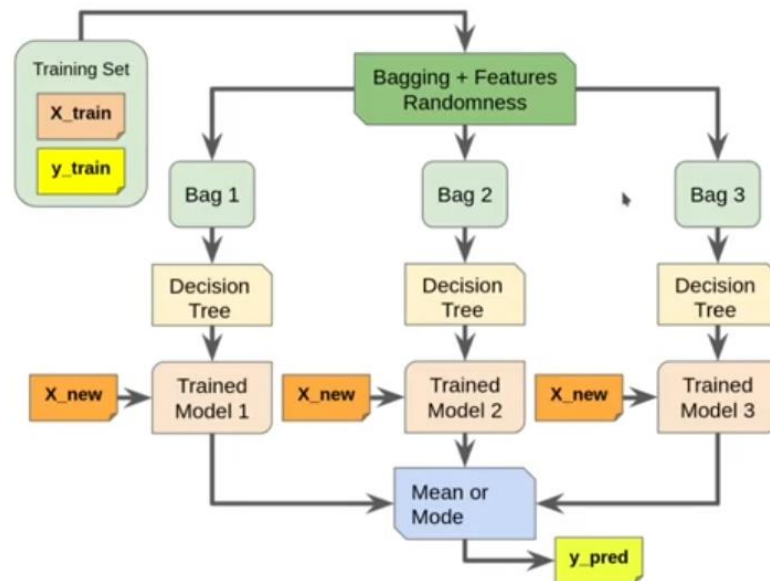
layer ini terdapat 1 *neuron* sebagai hasil dengan fungsi aktivasi sigmoid.

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 metrik penilaian yang akan digunakan untuk melihat hasil model yang sudah di *training*. 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 mengupdate 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 metrik disini menggunakan *accuracy* yang nantinya hasil *training* yang akan dilihat adalah nilai *accuracy* dan nilai *loss*.

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.

b. algoritma *Random Forest*

pada gambar 3.4 merupakan alur kerja algoritma *Random Forest*. Setelah data dipecah menjadi dua yaitu *training* dan *testing* dan 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 *training model* yang sudah ditentukan akan dilakukan *testing* menggunakan data *testing* dan setelah hasilnya didapat kemudian akan dilakukan *majority voting* untuk mendapatkan model prediksi terbaik.



Gambar 3.4 diagram blok alur algoritma *Random Forest*

ALGORITMA RANDOM FOREST

```

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 3.5 algoritma *Random Forest*

Berdasarkan gambar 3.5 yaitu Algoritma *Random Forest* dimana pada model algoritma ini setelah melakukan *splitting* data *training* 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* 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 akurasinya.

Selain itu menggunakan *confussion matrix* sebagai metrik penilaian kualitas model. Dengan *confussion matrix* ini maka akan menampilkan *accuracy*, *precision*, *recall* dan *f-score*. Dimana terdapat dua kategori *output* pada model ini yaitu 0 dan 1. Model yang termasuk model baik pastinya yang memiliki nilai *f1-score* karena jika *f1-score* nya baik maka nilai lainnya seperti nilai *accuracy*, *precision* dan *recall* juga bagus.

7. Evaluasi Model

Tahapan ini merupakan lanjutan dari tahapan sebelumnya yaitu *build* model, dimana 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 memberikan hasil yang sesuai maka model tersebut yang akan digunakan. Tetapi jika belum sesuai yang diharapkan maka akan mengulang tahapan preparasi data untuk diolah lagi supaya model yang dilatih bisa memberikan hasil yang bagus.

8. Penulisan Laporan

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

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang diperoleh dari penelitian prediksi kesuksesan kickstarter menggunakan algoritma *Backpropagation ANN* dan *Random Forest* adalah sebagai berikut :

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.

5.2 Saran

Saran untuk penelitian selanjutnya terkait prediksi kesuksesan kickstarter adalah sebagai berikut :

1. Penelitian selanjutnya agar memakai data yang berbeda supaya hasilnya bisa menjadi perbandingan.
2. Penelitian selanjutnya agar dilakukan dengan menggunakan metode ataupun algoritma lainnya supaya bisa menjadi perbandingan.
3. Penelitian selanjutnya agar membuat model tidak hanya untuk kickstarter tetapi jenis *Crowdfunding* lainnya.
4. Pengembangan penelitian prediksi kickstarter selanjutnya bisa menggunakan aplikasi android supaya lebih mudah dijangkau user.

DAFTAR PUSTAKA

P

DAFTAR PUSTAKA

- Ahmad, F. S., Tyagi, D., & Kaur, S. (2017). *Predicting Crowdfunding success with optimally weighted Random Forests*. Dubai, United Arab Emirates: IEEE.
- Algoritma Data Science Academy. (2022). *Cara Kerja Algoritma Random Forest*. Retrieved 02 12, 2022, from <https://algoritma.blog/cara-kerja-algoritma-random-forest-2022/#:~:text=Algoritma%20Random%20Forest%20meningkatkan%20keacakan,antara%20subset%20fitur%20yang%20acak>
- Chairunisa. (2018). *Pengetahuan, Kepercayaan, Informasi dan Teknologi yang Mempengaruhi Motivasi Masyarakat dalam Investasi Menggunakan Crowdfunding Syariah*. Retrieved 02 10, 2022, from <https://repository.uinjkt.ac.id/dspace/bitstream/123456789/41199/1/CHAIRUNISA-FEB.pdf>
- Chalimah, C. V. (2018). *Model Penjelas Keputusan Berdonasi Melalui Layanan Pendanaan Berbasis Financial Technology(Crowdfunding)*". Retrieved 02 10, 2022, from <https://dspace.uin.ac.id/bitstream/handle/123456789/11100/05.%20BAB%20I.pdf?sequence=5&isAllowed=y>
- Charman, A. (2018). *The FTC is investigating a Crowdfunding campaign that disappeared*. Retrieved 02 10, 2022, from <https://www.theverge.com/circuitbreaker/2018/8/29/17793720/ftc-ibackpack-investigation-indiegogo-kickstarter>
- Ginanto, N. (2012). *Backpropagation*. Retrieved 02 10, 2022, from <https://novikaginanto.wordpress.com/2012/11/14/backpropogation/>,
- Indonesia, I. (2018). *Crowdfunding: Resiko Kecurangan pada dana Patungan Online*. Retrieved 02 10, 2022, from <https://www.integrity-indonesia.com/id/blog/2018/09/13/Crowdfunding-risiko-kecurangan-pada-dana-patungan-online/>,
- Jhaveri, S., Khedkar, I., Kantharia, Y., & Jaswal, S. (2019, 03 29). *Success Prediction using Random Forest, CatBoost, XGBoost and AdaBoost for Kickstarter Campaigns*. Retrieved 02 16, 2022, from <https://ieeexplore.ieee.org/abstract/document/8819828>

- Kholis, I. (2011, 07 07). *Feed Forward dan Backpropagation*. Retrieved 02 10, 2022, from <https://ikkkholis27.wordpress.com/2011/07/07/feed-forward-dan-Backpropagation/>
- Li, B., Buck, A., Lu, P., & v-chmccl. (2022, 02). *Komponen Model Evaluasi*. Retrieved 02 12, 2022, from <https://docs.microsoft.com/id-id/azure/machine-learning/component-reference/evaluate-model>
- Novindasari, I. (2020). *Pengaruh Ukuran Batch dan learning rate dalam Konvergensi Gradient Descent*. Retrieved 02 10, 2022, from <https://idanovinda.medium.com/pengaruh-ukuran-batch-dan-learning-rate-dalam-konvergensi-gradient-descent-ebfe6f4cae75>
- Nugraha, M., & Alva, T. E. (2019). Kajian awal analisis atribut yang berpotensi mempengaruhi tingkat pendanaan pada proyek 3D printer di platform *Crowdfunding* kickstarter. *01*, 26-34.
- Nugroho, K. S. (2019). *Confussion Matrix Untuk Evaluasi Model pada Supervised learning*. Retrieved 02 12, 2022, from <https://ksnugroho.medium.com/confusion-matrix-untuk-evaluasi-model-pada-unsupervised-machine-learning-bc4b1ae9ae3f>
- Nur, A. L. (2020). *Analisis Prediksi dan Faktor Pendukung Kesuksesan Crowdfunding Berbasis Donasi Menggunakan Algoritma C4.5 dan K-Nearest Neighbor (Studi Kasus: Kitabisa.com)*. Retrieved 02 15, 2022, from <https://dspace.uui.ac.id/handle/123456789/31436>
- OnnoWiki. (2020). *Metric Evaluation Model*. Retrieved 02 13, 2022, from https://lms.onnocenter.or.id/wiki/index.php/Orange:_Metric_Evaluation_Model
- Plaosan, S. V. (2020). *Algoritma Random Forest*. Retrieved 02 10, 2022, from http://learningbox.coffeecup.com/05_2_randomforest.html
- Prasetyo, H. (2019). *Backpropagation Neural Network(Jaringan Syaraf Tiruan)*. Retrieved 02 10, 2022, from <https://hendroprasetyo.com/Backpropagation-neural-network-jaringan-saraf-tiruan/#.YmAOWShBzIU>
- R Kukul. (2018). *Melakukan Feature Scaling Pada Dataset*. Retrieved 02 10, 2022, from <https://medium.com/machine-learning-id/melakukan-feature-scaling-pada-dataset-229531bb08de>
- Rahmalia, N. (2021). *Kaggle, komunitas belajar data science yang bisa menghasilkan uang*. Retrieved 02 13, 2022, from <https://glints.com/id/lowongan/kaggle-adalah/#.YmAjuihBzIU>, <https://www.kaggle.com/datasets/kemical/kickstarter-projects>

- Rajendra, L. (2022, 01 03). *Artificial Neural Network*. Retrieved 02 10, 2022, from <http://sistem-informasi-s1.stekom.ac.id/informasi/baca/Artificial-Neural-Network/b1c26e9347ef547ff06845ca38cc443aedc4fa86>
- Rajendra, L. (2022). *Artificial Neural Network*. Retrieved 02 10, 2022, from <http://sistem-informasi-s1.stekom.ac.id/informasi/baca/Artificial-Neural-Network/b1c26e9347ef547ff06845ca38cc443aedc4fa86>
- Ryoba, M. J., & Zhou, S. Q. (2020, 01 22). *Feature subset selection for predicting the success of Crowdfunding project campaigns*. Retrieved 02 16, 2022, from <https://link.springer.com/article/10.1007/s12525-020-00398-4>
- Saputra, I. (2021). *Data Mining 5, klasifikasi Decission Tree dan Random Forest*. Retrieved 02 12, 2022, from https://www.slideshare.net/IrwansyahSaputra1/data-mining-5-klasifikasi-decision-tree-dan-random-forest?from_action=save
- Sarri, A. P. (2019, 10 21). *Pengaruh Brand Awareness, Kualitas Proyek Dan Kepercayaan Terhadap Keputusan Berdonasi Secara Online Pada Platform Crowdfunding KitaBisa.Com*. Retrieved 02 16, 2022, from <http://repository.uinsu.ac.id/id/eprint/6846>
- Sayagiri, G. (2021, 01 21). *Analisis faktor-faktor yang memengaruhi keberhasilan Crowdfunding melalui kitabisa.com*. Retrieved 02 14, 2022, from <https://osf.io/85frg/>
- Setiawan, B. J., Naomi, P., & Wijanarko, A. A. (2020). *Cowdfunding: Aaspek Kemitraan pada Penyelenggaraan (Studi pada platform gandengtangan)*. *Jurnal Manajemen & Bisnis Madani*, 02, 62-77.
- Stevanus, S. (2020). *Membicarakan Precission, Recall dan F1-score*. Retrieved 02 12, 2022, from <https://stevkarta.medium.com/membicarakan-precision-recall-dan-f1-score-e96d81910354>
- Tantina, I., & Hartono, B. (2019). *Pengembangan prediksi tingkat kesuksesan campaign pada portal reward-based Crowdfunding*. Retrieved 02 14, 2022, from https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiP18GnzJn5AhUkRmwGHeJKDI4QFnoECACQAQ&url=http%3A%2F%2Fetd.repository.ugm.ac.id%2Fpenelitian%2Fdetail%2F176005&usg=AOvVaw0FhBHZ1qDhi3HC_yrfAfYz
- Travel, G. (2022). *Apa Itu Kickstarter*. Retrieved 02 10, 2022, from <https://id.go-travels.com/17574-what-is-kickstarter-3486258-9020894>

- Wang, W., & Wu, H. Z. (2020, 03 07). *Prediction of fundraising outcomes for Crowdfunding projects based on deep learning: a multimodel comparative study*. Retrieved 02 16, 2022, from <https://link.springer.com/article/10.1007/s00500-020-04822-x>
- Widiputra, H. D. (2016, 10 12). *Artificial Nerual Network*. Retrieved 02 10, 2022, from <https://dosen.perbanas.id/artificial-neural-network/>
- Widiputra, H. D. (2016). *Artificial Nerual Network*. Retrieved 02 10, 2022, from <https://kotakode.com/blogs/3967/Mengenal-Feature-Engineering---Part-10>
- Wikipedia. (2021). *Random Forest*. Retrieved 02 10, 2022, from https://id.wikipedia.org/wiki/Random_forest
- Yanuar, A. (2018). *Random Forest*. Retrieved 02 12, 2022, from <https://machinelearning.mipa.ugm.ac.id/2018/07/28/random-forest/>
- Yu, P.-F., Huang, F.-M., Yang, C., Liu, Y.-H., & Li, Z.-Y. (2018). *Prediction of Crowdfunding Project Success with Deep Learning*. Xi'an, China: IEEE.
- Yunus, M. (2020). *Artificial Neural Network-Part1(Pengenalan)*. Retrieved 02 10, 2022, from <https://yunusmuhammad007.medium.com/6-artificial-neural-network-ann-part-1-pengenalan-db487b8f8d85>