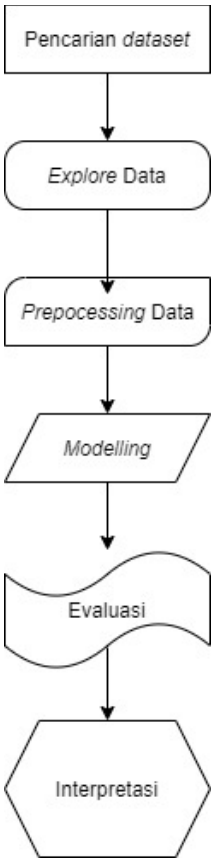
BAB III
METODE KERJA

3.1 Waktu dan Tempat

Program *Foundations of AI and Life Skills for Gen-Z* Studi Independen Bersertifikat Kampus Merdeka adalah program pelatihan *Artificial Intelligence* yang dilaksanakan secara *online* atau daring. Program ini berlangsung dari bulan Februari 2022 sampai dengan bulan Juli 2022 yang dilaksanakan setiap hari kerja (Senin sampai dengan Jumat) selama 8 jam per harinya dengan sesi pagi dimulai pukul 08.00 – 11.30 WIB dan kelas siang dilaksanakan pukul 13.00-17.30 WIB.

3.2 Metode dan Proses Kerja

Metode yang digunakan adalah *Machine Learning* dengan menggunkan model *Gradient Boost Classifier,* dimana metode ini bekerja untuk mengklasifikasikan *dataset* sehingga mampu memperoleh suatu klasifikasi.



Gambar 3. 2. 2 Flowchart proses pelaksanaan proyek akhir

3.2.1. Pencarian *Dataset*

Pengerjaan proyek akhir ini dimulai dengan mencari *dataset* yang berisi data klien bank. Pencarian dilakukan di *website* UCI *Machine Learning.*

3.2.2 *Explore Data*

Pada proses ini, dilakukan pemeriksaan dan penggalian data lebih lanjut. Pada *dataset* tersebut terdapat 45211 baris dan ada 17 kolom data. *Dataset* tersebut terdiri dari 16 kolom *feature* dan 1 kolom data *output*. Kolom *feature* terdiri dari 9 data kategori dan 7 *data numeric*.

- Data kategori :
  1. *Job* : pekerjaan
  2. *Marital* : status pernikahan
  3. *Education* : pendidikan
  4. *Default* : memiliki kredit atau tidak
  5. *Housing* : Pinjaman rumah
  6. *Loan* : Pinjaman pribadi
  7. *Contact* : Alat komunikasi
  8. *Month* : Bulan terakhir klien dihubungi
  9. *Poutcome* : Hasil dari kampanye pemasaran sebelumnya
- *Data numeric* :
  1. *Age* : umur
  2. *Balance* : saldo individu
  3. *Day* : hari terakhir klien dihubungi dalam seminggu
  4. *Duration*: durasi waktu berhubungan dengan klien dalam detik
  5. *Campaign*: jumlah kontak yang dilakukan selama kampanye ini untuk klien ini
  6. *Pdays*: jumlah hari yang dilewati setelah berhubungan terakhir kali dihitung dari kampanye sebelumnya
  7. *Previous* : jumlah kontak yang dilakukab sebelum kampanye ini untuk klien ini

3.2.3 *Preprocessing Data*

a. Data cleaning

Pada kolom "*job*" dilakukan *drop* pada klien yang memili "*job"* = "*unknown*" "*Admin*" dan "*management*" secara umumnya sama jadi keduanya diletakkan pada nilai kategori yang sama.

b. Data transformation

Metode *fit_transform*() diperoleh secara cuma-cuma dengan menambahkan *Transformer Mixin* sebagai *base class*. Jika menambahkan *Base Estimator* sebagai *base class* (dan menghindari **args dan **kwargs pada *constructor*) mendapatkan 2 tambahan dua metode *set_params*() dan menjadi *hyperparameter tuning* secara otomatis.

Untuk *transformer data* dengan *Skicit Learn* sendiri sangat berguna, namun perlu untuk menuliskan fungsi tugas tersendiri seperti operasi pembersihan khusus dan mengkombinasikannya dengan atribut spesifik.

Kolom-kolom kategori di *encoding* dengan *class* yang ada di dalam *Categorical Encoder* memiliki suatu kesulitan dalam menuliskan program tersebut, yaitu membuat isi fungsi tersebut untuk *mengencode* data kategorikal agar menjadi lebih baik. Dalam penggunaan *one-hot-encoding* terdapat keterlambatan *training* dan penurunan kinerja karena sebuah variabel kategorikal punya jumlah yang besar. Solusi yang dilakukan menggantikan *input categorical* dengan *feature numeric* yang berguna dan berhubungan ke kolom kategori.

- Transformasi *Pipelines*

  Selanjutnya konstruktor *Pipelines* untuk membantu urutan transformasi sehingga dapat dieksekusi dalam urutan yang benar.

- *Standard Scaler*

  Estimator selanjutnya adalah *Standar Scaler* yang mana sebuah *transformer* jadi seluruh transformasi diterapkan secara keseluruhan ke data secara berurutan

3.2.4 *Modelling*

a. *Stratified Sampling*

Stratified random sampling adalah suatu teknik pengambilan sampel dengan memperhatikan suatu tingkatan (*strata*) pada elemen populasi.

Elemen populasi dibagi menjadi beberapa tingkatan (stratifikasi) berdasarkan karakter yang melekat padanya. Dalam *stratified random sampling* elemen populasi dikelompokkan pada tingkatan-tingkatan tertentu dengan tujuan pengambilan sampel akan merata pada seluruh tingkatan dan sampel mewakili karakter seluruh elemen populasi yang heterogen[14].

Konsep *stratified sampling* penting ketika membangun sebuah model baik untuk klasifikasi maupun regresi. Untuk menghindari *overfitting* pada data, maka harus mengimplementasikan sebuah "*cross validation*" dimana harus dipastikan bahwa fitur-fitur yang memiliki pengaruh besar terhadap *label* terdistribusi secara merata. Melakuan *split data* menjadi *data training* dan *data test* serta implementasikan sebuah *stratified shuffle split* dari *library Python.*

b. Model Klasifikasi

Model klasifikasi yang digunakan adalah:

1. *Logistic Regression*
2. *Nearest Neighbours*
3. *Linear SVM*
4. *Gradient Boosting Classifier*
5. *Decision Tree*
6. *Random Forest*
7. *Neural Net*
8. *Naïve Bayes*

3.2.5 Evaluasi Model

Setelah diperoleh hasil model yang di*train,* didaptkanlah suatu hasil yang terbaik yaitu model *Gradient Boosting Classifier.* Untuk menghindari *overfitting* alternatif terbaik adalah menggunakan "*cross validation*". Langkah selanjutnya pengevaluasian dapat digunakan *confusion matrix* untuk menemukan *score* presisi dan *recall* f1 *score* pada *Gradient Boosting Classifier. Recall* adalah total jumlah "Ya" di label kolom-kolom pada *dataset*, sedangkan presisi berarti berapa prediksi yang tepat pada model terhadap label sebenarnya. Tujuan utama *confusion matrix* adalah untuk