

3. ANALISIS DAN DESAIN SISTEM

3.1. Analisis Permasalahan

Penggunaan *artificial intelligence* telah diketahui dapat mengurangi kemungkinan terjadinya sebuah *human error* dalam berbagai hal. Dalam kasus penentuan atau penegakan diagnosa, telah dilakukan berbagai macam penerapan guna membantu melakukan deteksi dan prediksi untuk menentukan sebuah penyakit tertentu. Meskipun demikian, banyak penelitian yang menghasilkan model dengan nilai akurasi yang berbeda. Seperti yang dilakukan oleh Shehab et al., (2022), Sridhar et al., (2023), dan juga Mahoto et al., (2022) dimana ketiga penelitian tersebut menghasilkan sebuah algoritma terbaik yang berbeda-beda. Hal ini diakibatkan dari perbedaan atribut dari data yang digunakan dan struktur dari *dataset* yang digunakan. Oleh karena itu muncul sebuah *urgency* untuk menentukan sebuah algoritma yang dapat menghasilkan hasil akurasi terbaik untuk mengolah data rekam medis untuk menghasilkan sebuah diagnosis yang relevan dan aktual terhadap keluhan dari pasiendengan berdasarkan pada data yang riil berdasarkan wilayah dari dalam Indonesia, secara spesifik, Pangkalanbun, Kalimantan Tengah.

3.2. Analisis Kebutuhan

Dalam upaya menemukan algoritma dengan hasil akurasi terbaik, penelitian skripsi akan menerapkan beberapa algoritma dengan hasil akurasi terbaik yang dihasilkan dari penelitian-penelitian sebelumnya dan juga menerapkan penggunaan *ensemble method* sebagai metode pendukung untuk meningkatkan akurasi. Pemilihan model -model tersebut meliputi :

1. Penggunaan metode Decision Tree tetap digunakan karena memiliki akurasi yang sama dengan Random Forest.
2. Penggunaan metode SVM digunakan karena memiliki hasil akurasi terbaik (Sridhar et al., 2023).
3. Penggunaan metode Naive Bayes digunakan karena memiliki hasil akurasi tertinggi berdasarkan penelitian dari Mahoto et al., (2022).
4. Penggunaan metode Extreme Gradient Boosting digunakan sebagai metode terbaru untuk membandingkan metode baru terhadap metode lama dengan akurasi yang baik.

Ensemble method yang digunakan meliputi Bagging atau Bootstrap Aggregating dan juga Voting.

3.3. Pengumpulan Data

Pengumpulan data dilakukan dengan menggunakan data rekam medis pasien yang didapat dari hasil pemeriksaan yang dilakukan oleh Dr. Rita Wey dengan diagnosis yang berjumlah 10 dan 350 kasus dari setiap diagnosis dari aplikasi berbasis web yang digunakan oleh klinik dan juga dari website PCare dari BPJS Kesehatan. Dataset sendiri terdiri dari 4 atribut penting yang berisikan karakteristik pasien (id [angka], jenis kelamin [0 = perempuan dan 1 = laki-laki], umur, tinggi badan, berat badan, tekanan darah atas, tekanan darah bawah, suhu tubuh, dan tingkat respirasi), obat yang diberikan, diagnosis yang diberikan, dan juga gejala pasien (batuk, batuk berdarah, pilek, sesak nafas, demam, luka, bernanah, pusing, lemas, bergetar, sakit dada, sakit tenggorokan, sakit badan, sakit sendi, sakit perut, sakit kulit, sakit_ulu_hati, sakit kepala, sakit leher, memar, tegang, mual, muntah, diare, kesemutan, susah bab, susah bak, bab berdarah, bak berdarah, bak bernanah, berdarah setelah berhubungan, kembung, riwayat darah tinggi, riwayat kencing manis, riwayat kolesterol, sariawan, keringat dingin, mimisan, gatal, timbilan, bintik pada badan, berasa tebal, telinga mendengung, sakit saat bak, sakit saat bab, keputihan, pembengkakan pada anus, pembengkakan pada tenggorokan, pembengkakan pada mata, penglihatan kabur, telinga kotor, sering bak, mimisan). Data yang diambil dari aplikasi klinik disimpan sebagai bentuk .json sementara data dari website bpjs berisikan data string dan angka yang diambil secara manual.

```
  "_id": {
    "$oid": "64d439f72ce92be2dbbd43fa"
  },
  "nik": "12345678910",
  "nama": "Vincentius Andy",
  "tl": "28-05-2001",
  "jk": "Laki-laki",
  "alergi": "paracetamol",
  "appointment": {
    "2023-08-10 08:14:31": {
      "age": 23,
      "height": "150",
      "weight": "60",
      "tensi": "100/70",
      "suhu": "33",
      "respirasi": "123",
      "anamnesa": "asdasdasd, lkjlkjlkj",
      "diagnosa": "malaria",
      "obat": "vometa"
    }
  }
}
```

Gambar 3.1. Contoh Data Dari Aplikasi Rekam Medis Sederhana

data dari aplikasi rekam medis sederhana yang digunakan oleh klinik merupakan data dengan tipe *dictionary (dict) object* dengan *keys* dan *values* dari *dictionary* mengandung unsur – unsur dari informasi pasien, seperti data diri yang mengandung nik, nama, tanggal lahir (tl), jenis kelamin (jk), dan alergi dari obat, dan juga data kunjungan yang mengandung umur, berat dan tinggi badan, suhu tubuh, tekanan darah, tingkat respirasi, anamnesa (keluhan), diagnosis, dan juga obat yang diberikan terhadap pasien.

Gambar 3.2. Contoh Gambar Identitas Pasien dari Website BPJS

Gambar 3.3. Contoh Data Hasil Pemeriksaan Pasien dari Website BPJS

Dari website BPJS, data yang digunakan disimpan kedalam bentuk *section*, dimana *section* pertama berisikan data diri dan data pendaftaran dari Pasien, dan *section* kedua berisikan anamnesa, diagnosa, obat, dan data pemeriksaan pasien.

Dalam proses pengambilan data, juga dilakukan proses *checking* dan klarifikasi dengan pihak dokter untuk menjaga konsistensi dan aktualitas data untuk menghindari *human error* dalam pemberian obat dan diagnosa yang akan dipelajari oleh model.

3.4. Pengolahan dan Analisis Data

Data yang telah diambil kemudian di *preprocess* dengan cara manual kedalam format data yang berdasarkan pada *research paper* dari Sridhar et al., (2023), dimana gejala pada data akan dikonversikan menjadi kolom masing-masing berdasarkan tiap gejala, dan diisikan oleh 0 atau 1 bila pasien mengalami gejala tersebut kemudian disimpan kedalam file dengan format .csv.

Penerapan preprocessing pada data meliputi *cleaning*, dimana data yang mengalami *mislabelled* akan dikoreksi atau dihapus sesuai dengan hasil konsultasi dengan pihak dokter, *transformation* dimana struktur dari data akan disesuaikan kedalam *file .csv*, dan *integration* dimana data yang didapat dari website BPJS dan aplikasi rekam medis dari klinik akan digabungkan kedalam format yang sama. *Preprocessing* dilanjutkan dengan mengkonversi isi data bila isi dari gejala tidak sama dengan 1 dari yang awalnya nan menjadi 0 agar dapat diproses oleh model. Setelah melakukan konversi data gejala kedalam *binary*, data kemudian di normalisasi agar memiliki *convergence rate* yang lebih baik, dan juga menghasilkan atribut data yang mirip dan tidak terlalu jauh agar mempermudah model dalam membaca data.

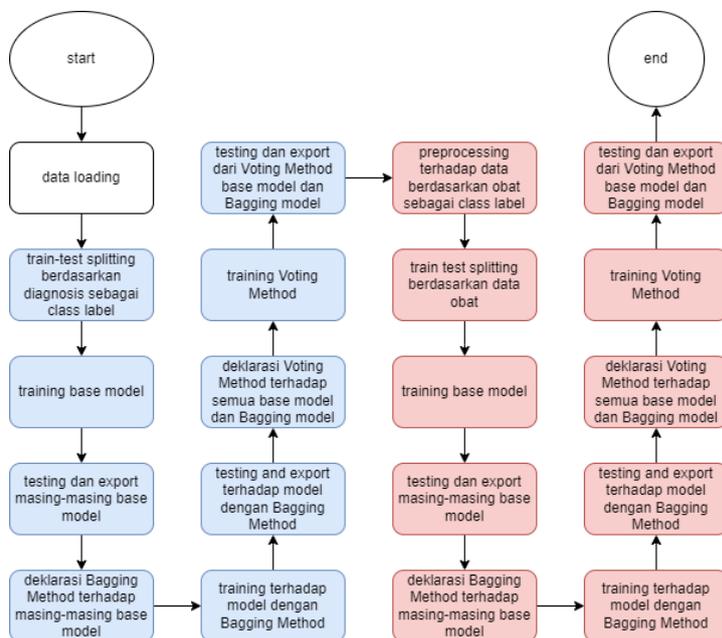
Dalam proses prediksi dari obat, data akan mengalami proses *cleaning* secara lebih mendalam, dimana data akan dikembangkan dengan menggunakan *function* *Explode* dari *library* *pandas*, dimana data baris akan diduplikasi berdasarkan obat yang ada dalam tiap baris, kemudian akan dibersihkan untuk menghasilkan nama obat yang konsisten dari setiap baris. Obat kemudian akan diklasifikasikan sesuai dengan 'replacement' yang berdasarkan kandungan dari obat sesuai dengan data dari dr. Rita Wey untuk menjaga konsistensi dan proses pembelajaran model. Proses *oversampling* kemudian akan diterapkan terhadap data obat untuk memastikan proses pembelajaran yang *balanced* terhadap semua *class label* obat.

3.5. Metode yang Diusulkan

Berdasarkan penelitian yang dilakukan dari Mahoto et al., (2022) dan Sridhar et al., (2023), dapat disimpulkan pada hasil akhir bahwa penggunaan dari *ensemble method* dapat membantu meningkatkan akurasi dari prediksi yang dilakukan. Metode yang dipilih dalam penerapan *ensemble method* adalah metode bagging bagi keduanya, dan dapat meningkatkan akurasi hingga 98%. Meskipun demikian, penggunaan metode bagging dalam penelitian keduanya dilakukan terhadap semua model, meliputi model dengan akurasi yang rendah dan tidak adanya penerapan model *machine learning* yang lebih moderen. Oleh karena itu, penelitian skripsi ini menerapkan beberapa metode untuk menggantikan metode-metode yang lama dari penelitian sebelumnya, dan juga menerapkan model dengan akurasi terbaik dari penelitian sebelumnya.

3.6. Desain Model yang Dikembangkan

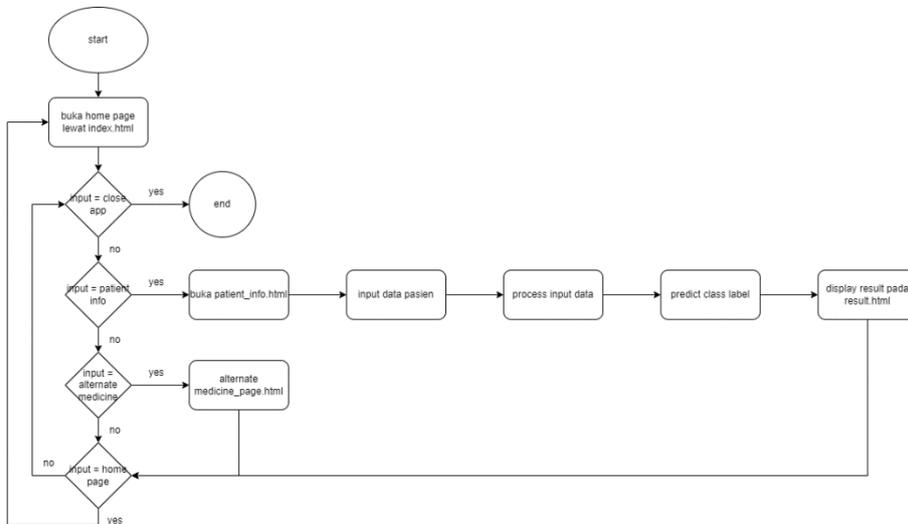
Penerapan dari model *machine learning* terhadap proses prediksi dilakukan sesuai dengan flowchart :



Gambar 3.4. Flowchart Tahapan Pengerjaan Penelitian Skripsi

Tahapan pengerjaan dari proses pengerjaan model meliputi *data loading* dari Google Drive kedalam *code* yang dikerjakan di Google Colab, *data splitting* berdasarkan diagnosis, training terhadap *base model* yang dilanjutkan dengan *testing* dan *export base model* kedalam *file .pkl* untuk digunakan dalam validasi oleh aplikasi berbasis web, deklarasi dari Bagging Method terhadap masing-masing *base model*, *training* model Bagging Method

yang dilanjutkan *testing* dan *export* dari model, deklarasi Voting Method terhadap *base model* maupun model yang menggunakan Bagging Method, *training* dari model Voting Method, *testing* dan *export* hasil model, kemudian preprocessing terhadap data dimana obat dijadikan sebagai *class label*, dilanjutkan dengan *train-test splitting*, dan kemudian menerapkan *training*, *testing*, dan *export* terhadap model-model yang melakukan *fitting* dengan *class label* obat.



Gambar 3.5. *Flowchart* cara kerja aplikasi berbasis web

Proses validasi dari pihak tenaga kesehatan sendiri akan dilakukan menggunakan sebuah aplikasi berbasis web yang dapat menerima input berupa data pasien yang akan diprediksi diagnosis dan obatnya. Aplikasi berbasis web memiliki 4 halaman, *home page* yang berisikan statistika akurasi dari semua model berdasarkan *evaluation matrix*, *patient page* yang berisikan *textbox* untuk *input* karakteristik pasien dan *checkbox* untuk *input* gejala pasien, *result page* yang berisikan hasil prediksi beserta *confidence score* dari model yang melakukan prediksi, dan juga *alternate medicine page* untuk menampilkan obat lain yang bisa digunakan sebagai pengganti dari obat yang diprediksi oleh model.

Model yang digunakan menerapkan 5 *machine learning algorithm* yang diambil berupa SVM, Random Forest, Decision Tree, Naïve Bayes, dan juga XGBoost, dengan penggunaan *hyperparameter tuning* :

a. *Hyperparameter Tuning*

Proses *tuning* terhadap parameter dilakukan kedalam 3 bagian, masing-masing terhadap tiap model dan metode yang digunakan. Proses *tuning* meliputi *tuning*

terhadap semua *base model* dari masing-masing prediksi, *tuning* terhadap masing-masing model dengan Bagging Method, dan *tuning* terhadap model yang digunakan dalam Voting Method. Setelah proses *tuning*, parameter terbaik akan disimpan ke dalam model untuk kemudian dibandingkan performanya dengan model yang lain. Proses ini juga memiliki perbedaan antara *tuning* dari model untuk prediksi diagnosis dan untuk prediksi obat dikarenakan perbedaan jumlah data yang berdampak pada waktu yang dibutuhkan model untuk mengolah data. *Library* yang digunakan dalam *tuning hyperparameter* merupakan GridSearchCV dari SKLearn. Pengurangan total parameter yang diujikan pada proses prediksi obat diakibatkan dari waktu yang dibutuhkan dari pengujian tiap parameter terhadap masing-masing model.

b. SVM :

Pada model SVM, beberapa parameter yang diuji meliputi : 'C', 'gamma', dan juga 'kernel', dengan nilai yang diujikan berupa

Tabel 3.1.

Tabel parameter SVM yang diujikan

<pre>param_grid_svc = { 'C': [1, 0.1, 10, 100], 'gamma': ['scale', 1, 0.1, 0.01, 0.001], 'kernel': ['linear', 'rbf'] }</pre>	Prediksi diagnosis
<pre>param_grid_svc = { 'C': [1], 'gamma': ['scale', 1, 0.1], 'kernel': ['rbf', 'linear'] }</pre>	Prediksi obat
<pre>param_grid_svc = { 'C': [1], 'gamma': ['scale'], 'kernel': ['linear'] }</pre>	<i>Default parameter</i>

c. Random Forest

Model Random Forest menerapkan *hyperparameter tuning* terhadap: 'n-estimator', 'max_depth', dan juga 'min_samples_split', dengan nilai yang diujikan

Tabel 3.2.

Tabel parameter Random Forest yang diujikan

<pre>param_grid_rf = { 'n_estimators': [100, 50, 200], 'max_depth': [None, 10, 20, 30], 'min_samples_split': [2, 5, 10] }</pre>	Prediksi diagnosis
<pre>param_grid_rf = { 'n_estimators': [100, 10, 25, 50], 'max_depth': [None, 10, 20, 30], 'min_samples_split': [2, 5, 10] }</pre>	Prediksi obat
<pre>param_grid_rf = { 'n_estimators': [100], 'max_depth': [None], 'min_samples_split': [2] }</pre>	<i>Default parameter</i>

d. Decision Tree

Penggunaan *hyperparameter tuning* pada model Decision Tree meliputi parameter : 'max_depth' dan 'min_samples_split' dengan nilai yang diujikan

Tabel 3.3.

Tabel parameter Decision Tree yang diujikan

<pre>param_grid_dtc = { 'max_depth': [None, 10, 20, 30], 'min_samples_split': [2, 5, 10] }</pre>	Prediksi diagnosis
<pre>param_grid_dtc = { 'max_depth': [None, 10, 20], </pre>	Prediksi obat

'min_samples_split': [2, 5, 10] }	
param_grid_dtc = { 'max_depth': [None], 'min_samples_split': [2] }	<i>Default parameter</i>

e. Naïve Bayes

Pada model Naïve Bayes, penerapan proses *tuning* meliputi parameter 'var_smoothing' dengan nilai yang diujikan

Tabel 3.4.

Tabel parameter Naïve Bayes yang diujikan

param_grid_nb = { 'var_smoothing': [1e-9, 1e-8, 1e-7] }

f. XGBoost

Penggunaan model XGBoost dilakukan dengan membandingkan parameter: 'n_estimator', 'max_depth', 'learning_rate', dan juga 'subsample' dengan nilai yang diujikan mencakup:

Tabel 3.5.

Tabel parameter XGBoost yang diujikan

param_grid_xgb = { 'n_estimators': [100, 50, 200], 'max_depth': [6, 3, 9], 'learning_rate': [0.3, 0.01, 0.1, 0.2], 'subsample': [1.0, 0.6, 0.8] }	Prediksi diagnosis
param_grid_xgb = { 'n_estimators': [100, 5, 25], 'max_depth': [3, 6],	Prediksi obat

<pre>'learning_rate': [0.1, 1], 'subsample': [0.5, 1.0], 'objective': ['multi:softprob', 'multi:softmax'] }</pre>	
<pre>param_grid_xgb = { 'n_estimators': [100], 'max_depth': [3], 'learning_rate': [0.1], 'subsample': [1.0], 'objective': ['multi:softmax'] }</pre>	<i>Default parameter</i>

Model yang diusulkan adalah penggunaan *ensemble method* terhadap 5 model yang dipilih berdasarkan penelitian sebelumnya. Penelitian skripsi menerapkan metode Bagging pada semua model secara individu dan juga menerapkan metode Voting terhadap semua model untuk menghasilkan *output* dengan tingkat akurasi yang lebih baik.

Penerapan dari *ensemble method* menggunakan library SK-Learn dengan *function* masing-masing VotingClassifier dan BaggingClassifier.

g. Bagging :

Penerapan dari metode bagging diterapkan terhadap semua model secara individu dengan masing-masing model menerapkan parameter yang didapatkan dari proses *hyperparameter tuning*.

Penggunaan Bagging Method juga menerapkan adanya proses *tuning* terhadap parameter yang digunakan saat deklarasi, dengan parameter tersebut meliputi : 'n_estimator', 'max_samples', dan juga 'random_state' dengan nilai yang diujikan meliputi :

Tabel 3.6.

Tabel parameter Bagging Method yang diujikan

<pre>Param_grids: { 'n_estimators': [10, 20, 50], 'max_samples': [0.5, 0.7, 1.0],</pre>	Prediksi diagnosis
-----------------------------------------------------------------------------------------	--------------------

'random_state': [42] },	
Param_grids: { 'n_estimators': [1, 5], 'max_samples': [0.5, 0.7, 1.0], 'random_state': [42] }	Prediksi obat

h. Voting :

Metode Voting diterapkan terhadap semua model kedalam sebuah *voting classifier* masing-masing terhadap semua base model dan model dengan Bagging method, dimana semua model yang digunakan telah melalui proses *hyperparameter tuning*.

Penerapan Voting Method juga melalui proses *tuning* pada parameter 'weights' dimana penerapan parameter ini akan memberikan bobot lebih terhadap masing-masing model yang digunakan untuk dibandingkan dengan nilai :

Tabel 3.7.

Tabel parameter Voting Method yang diujikan

<pre> param_grid = { 'voting': ['hard'], 'weights': [[1, 1, 1, 1, 1], [1, 2, 1, 1, 1], [1, 1, 2, 1, 1], [2, 1, 1, 1, 1], [1, 1, 1, 2, 1], [1, 1, 1, 1, 2],] } </pre>

3.7. Proses validasi

Pemberlakuan validasi terhadap *performance* dari penggunaan metode *machine learning* dilakukan dengan menggunakan 2 (dua) cara, yakni dengan menggunakan *confusion* dan *evaluation matrix* untuk menilai performa dari algoritma yang digunakan dan validasi dengan pihak tenaga kesehatan untuk menilai kelayakan dalam penggunaan algoritma dalam melakukan tugasnya untuk menjadi salah satu alat bantu dalam pemberian diagnosis.

3.7.1. Confusion Matrix

Confusion matrix merupakan tolak ukur dari performa dari sebuah algoritma klasifikasi machine learning yang sangat berguna dalam penghitungan *recall*, *precision*, *accuracy*, *specificity*, dan juga AUC-ROC curves (S. Narkhede, June 15, 2021). Dalam melakukan proses validasi dari hasil algoritma yang digunakan, penelitian skripsi menghasilkan *confusion matrix* dengan ukuran 10 x 10 untuk membandingkan hasil prediksi yang dihasilkan oleh algoritma *machine learning* dengan 10 mewakili 10 diagnosa yang akan diprediksi. Dalam prediksi obat, *confusion matrix* akan menghasilkan matrix dengan ukuran 97 x 97 yang mewakili masing masing obat yang digunakan.

3.7.2. Evaluation Matrix

proses validasi dengan menerapkan *evaluation matrix* berisikan akurasi, presisi, dan *recall*, dimana validasi akan dilakukan dengan menalpilkan *evaluation matrix* dari semua model yang digunakan guna membandingkan performa tiap model.

3.7.3. Validasi dengan Tenaga Kesehatan

Proses validasi dengan tenaga kesehatan akan dilakukan dengan memberikan hasil *output* dari data *test* kepada dr. Rita Wey dan juga dr. Mahadi. Proses validasi dilakukan dengan menginputkan karakteristik dan gejala pasien kedalam *UI* sederhana yang kemudian akan menampilkan *output* berupa hasil prediksi diagnosa dengan 5 model algoritma *machine learning* berserta *ensemble method* yang digunakan dan juga hasil prediksi dari obat yang juga berdasarkan 5 model algoritma *machine learning* dan *ensemble method*.