e-ISSN 2962-6129

Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)

Jakarta-Indonesia, 20 Agustus 2022

Deteksi Dini Penyakit Kanker Paru dengan Gabungan Algoritma *Adaboost* dan *Random Forest*

Roy Binsar Sinaga [1], Didit Widiyanto[2] , Bambang Tri Wahyono[3]
Informatika / Departemen
UPN Veteran Jakarta
Jl. RS. Fatmawati Raya, Pd. Labu, Kec. Cilandak, Kota Depok, Daerah Khusus Ibukota Jakarta 12450
roybs@upnvj.ac.id[1], didit.widiyanto@upnvj.ac.id[2], bambang.triwahyono@upnvj.ac.id[3]

Abstrak. Di Indonesia, kanker paru adalah kanker paling banyak diidap pria dan paling banyak kelima pada wanita di antara kanker lainnya. Serangkaian tes diagnostik yang kompleks dan memakan waktu dilakukan untuk mendiagnosis seseorang menderita kanker paru-paru atau tidak. Oleh karena itu, dilakukan pengujian menggunakan gabungan *Adaboost* dan *Random Forest* untuk deteksi dini kanker paru berdasarkan seperangkat parameter yang berhubungan dengan kanker paru. Penelitian ini memanfaatkan data sekunder dari *website kaggle.com*. Data berjumlah 309 set data dimana ada 10 fitur dan 1 kelas yang selanjutnya dilakukan praproses. Kemudian dibentuk dua model *Random Forest* dan model *Adaboost* yang menjadikan *Random Forest* sebagai pembelajar yang lemah. Sesudah model terbentuk, maka dilanjutkan proses pengujian menggunakan data uji. Dari performa yang dihasilkan dapat disimpulkan bahwa kombinasi *Adaboost* dan *Random Forest* mencapai *accuracy, precision, recall, dan specificity* yang lebih tinggi dari penerapan *Random Forest* tanpa *Adaboost* yaitu nilainya masing-masing 95,40%, 96%, 96,30% dan 96%.

Kata Kunci: Kanker Paru, *Random Forest, Adaboost.*

1 Pendahuluan

Kanker paru didefinisikan sebagai tipe penyakit ganas yang mengenai paru, kanker paru dibagi menjadi dua yaitu kanker paru primer dan kanker paru sekunder. Berdasarkan data dari WHO, kanker ini menempati posisi pertama di Indonesia pada jenis kanker terbanyak yang menyerang laki-laki dan paling banyak kelima yang menyerang wanita. Kebiasaan merokok adalah pemicu pokok kanker paru, baik itu pada perokok aktif maupun perokok pasif. Hal inilah yang menjadi dasar mengapa kanker paru banyak diidap laki-laki. Terdapat berbagai pemicu lain kanker paru, mulai dari lingkungan pasien tercemar oleh polusi udara dan unsur kimia berisiko lainnya, ataupun karena keluarga pasien juga pernah atau sedang mengidap kanker atau penyakit paru lainnya. Diagnosis kanker paru dilakukan melalui serangkaian langkah-langkah, pertama adalah *history taking,* lalu *physical examination, anatomical pathology examination, laboratory examination, imaging examination, special examination*, dan *examination* lainnya. Serangkaian langkah-langkah dalam mendiagnosis tersebut merupakan hal yang rumit dan diperlukan waktu yang lama untuk mendapatkan hasil akhir diagnosis. Selain itu karena proses yang rumit untuk mendapatkan hasil diagnosis yang akurat, diperlukan ketelitian dan pertimbangan yang komprehensif oleh tenaga medis [1].

Tahapan diagnosis kanker paru yang rumit tersebut menjadi dasar mencuatnya ide dalam menggunakan kombinasi algoritma klasifikasi dalam mendeteksi dini kanker paru. Berdasarkan studi literatur yang telah dilakukan terdapat beberapa penelitian yang menjadi pertimbangan dalam memilih algoritma klasifikasi. Pertama, penelitian dengan judul *AdaBoost Algorithm with Random Forests for Predicting Breast Cancer Survivability* yang ditulis oleh Thongkam, dkk dimana mereka mengkombinasikan algoritma *Adaboost* dan *Random Forest* dalam membangun model prediksi *breast cancer survival.* Didapatkan bahwa kombinasi *Adaboost* dan *Random Forest* menghasilkan nilai akurasi 88.60% dimana nilai ini paling tinggi dibanding metode lain yang coba digunakan sebagai pembanding dalam penelitian ini [2]. Kedua, penelitian dengan judul Penerapan Metode *Adaboost* Untuk Mengoptimasi Prediksi Penyakit Stroke Dengan Algoritma *Naïve Bayes* yang ditulis oleh Byna, dkk dimana mereka mengkombinasikan algoritma *Adaboost* dan *Naïve Bayes* dengan harapan setelah *Adaboost* diterapkan

dapat meningkatkan akurasi. Hasil akhir menunjukkan bahwa kombinasi *Adaboost* dan *Naïve Bayes* menghasilkan akurasi 98,10% dimana nilai ini lebih tinggi dibanding penerapan *Naïve Bayes* tanpa *Adaboost* yaitu sebesar 97,60% [3]. Berdasarkan kedua penelitian terdahulu tersebut maka dalam penelitian akan digunakan algoritma klasifikasi *Adaboost* dan *Random Forest* dimana akan dilihat bagaimana performa dari gabungan algoritma klasifikasi *Adaboost* dan *Random Forest* dibanding dengan performa dari algoritma *Random Forest* tanpa pengkombinasian dengan *Adaboost* dalam melakukan deteksi dini kanker paru.

2 Landasan Teori

2.1 Kanker Paru

Kanker paru didefinisikan sebagai tipe penyakit ganas yang mengenai paru, kanker paru dibagi menjadi dua yaitu kanker paru primer dan kanker paru sekunder. Kanker paru primer merupakan kanker paru yang bersumber dari internal paru dimana kanker paru primer berwujud tumor ganas yang asalnya dari *bronchogenic carcinoma* [1]. Kanker paru sekunder merupakan kanker paru yang bersumber dari eksternal paru lebih tepatnya kanker ini sumbernya dari perambatan kanker jenis lainnya yang terjalin di dalam tubuh manusia [4].

2.2 *Adaptive Boosting*

Adaboost merupakan akronim dari *Adaptive Boosting* termasuk kedalam *Ensemble Methods /Boosting Methods* yang sering dipakai. Secara garis besar proses yang dilakukan dalam *Adaboost* ialah membangun sejumlah *weak learners* yang tidak memiliki korelasi satu sama lain, lalu kemudian menggabungkan prediksinya. Dalam penerapannya *Adaboost* dikombinasikan dengan algoritma lain dengan tujuan untuk mengoptimalisasi performa yang dihasilkan [5]. Adaboost $H_k(x)$ didefinisikan sebagai

$$H_k(x) = \sum_{t=1}^{T} \left(\frac{log\,1}{\beta_t}\right) h_t^k(x) \quad (1)$$

Dimana $h_t^k(x)$ merupakan *weak learners* yang memiliki nilai error terendah , sedangkan β_t merupakan bobot dari *weak learners* tersebut. Premis akhir dalam *Adaboost* dihasilkan dari kombinasi *weak learners* yang memiliki nilai suara tertinggi [6].

2.3 *Random Forest*

Random Forest merupakan pengembangan dari metode *Decision Tree* yang mana *Random Forest* merupakan gabungan dari beberapa *Decision Tree*, dalam algoritma ini pengambilan keputusan dilakukan dengan cara *voting* untuk menentukan suara yang dominan dari keseluruhan *decision trees* [7].
Metode *Random Forest* bekerja dimulai dari pembentukan *trees*, dimana setiap *decision tree* dibentuk dengan menerapkan *gini index* yang didefinisikan

$$Gini\ Index(D) = 1 - \sum_{i=1}^{m} P_i^2 \quad (2)$$

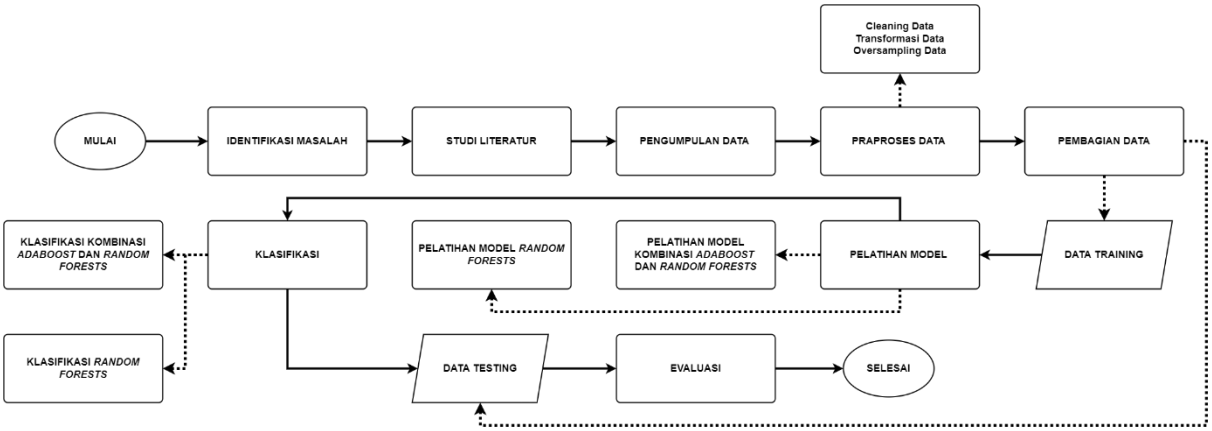
Dimana P_i adalah jumlah atribut pada tiap kelas dan m adalah jumlah dari tiap atribut. Fitur yang memiliki nilai *total gini index* terendah menjadi *root node* pada *tree*. *Total gini index* didefinisikan

$$Tot.\ Gini\ Index(K) = \frac{T_1}{T} Gini\ Index\ (D_1) + \frac{T_2}{T} Gini\ Index\ (D_2) \quad (3)$$

Dimana T_1 merupakan total *record* yang di kelas kesatu, T_2 merupakan total *record* yang di kelas kedua, dan T merupakan total *record* di semua kelas. Proses dilanjutkan dengan pembentukan *child node* hingga semua *node* pada *tree* sudah tidak dapat di *split.* Setelah seluruh pohon terbentuk dilanjutkan tahapan klasifikasi dengan menggunakan *voting* [8].

3 Metode Penelitian

Diagram alir menunjukkan hierarki penelitian yang dilakukan penulis untuk mencapai tujuan penelitian. Gambar 1 di bawah ini menunjukkan tahapan penelitian dari awal hingga akhir :



Gambar 1. Kerangka Pikir

3.1 Identifikasi Masalah

Fase identifikasi masalah adalah fase dimana penulis memutuskan masalah apa yang harus dipecahkan dan diatasi. Penulis mengidentifikasi isu-isu yang muncul selama ini terkait dengan objek penelitian yang dipilih. Dalam hal ini, pertanyaan yang diajukan oleh penulis adalah bagaimana kinerja yang dihasilkan dari penggabungan algoritma klasifikasi *Random Forest* dengan *Adaboost* untuk deteksi dini kanker paru dibandingkan dengan *Random Forest* tanpa *Adaboost*. Setelah masalah diidentifikasi, penulis dapat membuat solusi untuk memecahkan masalah yang diajukan.

3.2 Studi Literatur

Studi literatur yang diterapkan sebagai bagian dari penelitian ini bertujuan untuk menemukan beragam acuan sehubungan dengan masalah yang diidentifikasi, termasuk referensi dari jurnal, buku, dan artikel. Setelah berbagai acuan terkumpul cukup banyak, maka ditentukan penyelesaian yang bisa diterapkan untuk menyelesaikan permasalahan yang berasal dari acuan yang terkumpul. Masalah tersebut dipecahkan dengan menjalankan serangkaian proses percobaan sampai hasil yang dominan menjadi terlihat. Selain itu, saat meneliti literatur, peneliti harus membaca berbagai referensi untuk memahami konsep dan mekanisme metode yang dipilih.

3.3 Pengumpulan Data

Tahapan ini bertujuan untuk memperoleh data yang dibutuhkan untuk keperluan penelitian. Data yang digunakan pada penelitian ini merupakan data sekunder berasal dari *website 'kaggle.com'* dengan URL *https://www.kaggle.com/datasets/mysarahmadbhat/lung-cancer* dengan judul *"Lung Cancer"*. Dataset terdiri dari 16 atribut yang dibagi menjadi 15 atribut independen yaitu *Gender, Age, Smoking, Yellow Fingers, Anxiety ,Peer Pressure, Chronic Disease, Fatigue, Allergy, Wheezing, Alcohol Consuming, Coughing, Shortness Of Breath, Swallowing Difficulty,* dan *Chest Pain* . Kemudian 1 atribut dependen yaitu kelas *Lung Cancer* yang mewakili diagnosis apakah seseorang mengidap kanker paru-paru berdasarkan atribut independen. Total data dari dataset adalah 309 data. Data sekunder ini selanjutnya divalidasi oleh dokter spesialis paru untuk mengetahui validitas atribut yang digunakan dalam kumpulan data ini.

3.4 Praproses Data

Setelah data yang diperlukan dikumpulkan, perlu dimodifikasi sehingga algoritma yang telah ditetapkan bisa diimplementasikan pada data untuk mendapatkan hasil akhir yang memecahkan masalah. Langkah pertama dalam praproses data yaitu melakukan *cleaning data* dari *missing value*. Selanjutnya, perlu memeriksa tipe data dan mengonversi tipe data ke tipe data yang sesuai untuk kolom yang tipe datanya tidak sesuai. Kemudian, kolom dengan atribut kategoris diubah menjadi numerik. Terakhir, diperiksa apakah data yang digunakan sudah *balance*. Jika pada data kanker paru terdapat ketidakseimbangan kelas, maka dilakukan *oversampling* sehingga jumlah kelas minoritas sama dengan jumlah kelas mayor. Dalam penelitian ini, kami menerapkan SMOTE *(Synthetic Minority Oversampling Technique)* sebagai teknik *oversampling*. Dalam teknik ini, aturan yang digunakan adalah melipatgandakan data kelas minoritas untuk menyeimbangkan kelas mayoritas dan sistem membuat data buatan [9].

3.5 Pembagian Data

Pada tahapan ini dilakukan *split data* dengan mengimplementasikan teknik *hold-out validation.* Teknik ini bekerja dengan membagi data menjadi *training set* dan *testing set* secara acak. Pada penelitian ini akan diterapkan dua kali percobaan *split data* untuk mendapatkan pasangan *training set* dan *testing set* yang berkinerja terbaik. *Split data* akan diimplementasikan dalam dua skenario, skenario pertama adalah 70% *training set*: 30% *testing set*, skenario kedua adalah 80% *training set*: 20% *testing set*.

3.6 Pelatihan Model

3.6.1 *Random Forest*

Langkah-langkah proses pelatihan model pada algoritma *Random Forest* dijabarkan seperti di bawah ini:

1. Tetapkan total *trees* yang akan dibuat.
2. Lalu *subsample dataset* acak dibentuk dan *tree* dibangun dari setiap *subsample dataset*. Proses ini berlanjut hingga jumlah *trees* yang terbentuk sama jumlahnya dengan *trees* yang ditetapkan semula. Setiap *tree* yang dibuat mempunyai bobot yang sama.
3. Terakhir setelah menginput data pelatihan ke setiap *tree* dan mendapatkan hasil prediksi untuk setiap *tree*, maka dilanjutkan proses *voting* sehingga bisa diketahui mana kelas yang mendapat suara terbesar, dan hasil prediksi akhir untuk setiap baris adalah berdasarkan hasil voting tersebut.

3.6.2 Gabungan *Adaboost* dan *Random Forest*

Langkah-langkah proses pelatihan model pada gabungan algoritma *Adaboost* dan *Random Forest* dijabarkan seperti di bawah ini:

1. Tetapkan bobot awal w = 1/N, dimana N adalah total *record* baris di sampel *dataset.*
2. Tetapkan total iterasi yang akan diterapkan pada tahapan *Adaboost* ini.
3. Pada tiap iterasi proses diawali dengan normalisasi bobot pada sampel data sampai jumlahnya = 1.
4. Pada tahap berikutnya, *Random Forest* diimplementasikan pada setiap fitur dalam sampel *dataset*. Langkah pertama dalam implementasi yaitu menentukan jumlah *trees* yang akan dibentuk pada satu fitur. Hanya satu fitur yang dipakai untuk membuat *tree*, sehingga *tree* yang terbentuk berbentuk pohon satu tingkat atau disebut *stump*.
5. Sesudah total *trees* yang akan dibuat ditetapkan, *subset* baru dari *dataset* akan dibentuk secara acak dan setiap *subset* baru dari *dataset* menjadi *stump*.
6. Sesudah *stump* kesatu berhasil dibuat, bentuk *stump* selanjutnya seperti pada *step* 5.
7. Sesudah *Random Forest* terbentuk pada fitur 1, *Random Forest* diimplementasikan pada kumpulan data sampel untuk menentukan prediksi yang didapatkan. Hasil prediksi didapatkan dengan voting yang mana tiap *tree* berbobot satu.
8. Sesudah mengetahui hasil prediksi, terlihat mana *record* yang misklasifikasi. Berdasarkan *record* misklasifikasi, dapat ditemukan nilai *error* dari fitur pertama, di mana nilai *error* adalah jumlah bobot *record* yang misklasifikasi.
9. Sesudah mendapatkan nilai *error* untuk fitur 1, lakukan kembali step 4 hingga 8 guna memperoleh nilai *error* untuk fitur lainnya.
10. Sesudah seluruh fitur mempunyai nilai *error* masing-masing, pilih fitur yang memiliki nilai *error* terkecil.

11. Berikutnya hitung bobot suara untuk fitur yang memiliki nilai *error* terkecil.
12. Selanjutnya, lakukan proses *boosting* terhadap *record* misklasifikasi pada fitur yang memiliki nilai *error* terkecil supaya tidak misklasifikasi pada iterasi selanjutnya.
13. Sesudah iterasi kesatu dilakukan dan telah diperoleh bobot suara untuk iterasi kesatu, ulangi tahapan dari step 3 hingga 12 untuk iterasi kedua dan seterusnya.
14. Sesudah seluruh iterasi telah dijalankan, diperoleh hasil iterasi berbentuk *Random Forest* yang tiap *Random Forest* telah memiliki bobot suara tertentu.

3.7 Klasifikasi

Pada langkah ini, dilakukan penerapan model *Random Forest* dan model gabungan yang dibentuk dari *Adaboost* dan *Random Forest* untuk menguji data guna melihat bagaimana prediksi yang didapatkan dari kedua model tersebut. Dalam model *Random Forest*, klasifikasi dilakukan dengan menerapkan data uji ke setiap pohon. Setelah hasil prediksi untuk setiap pohon diperoleh, dilakukan voting untuk menentukan kelas mana yang memperoleh suara terbanyak, dan hasil voting tersebut merupakan hasil klasifikasi akhir untuk data uji. Pada kombinasi *Adaboost* dan *Random Forest* prosesnya yaitu memasukkan data uji ke setiap pohon iterasi pertama. Setelah diperoleh hasil klasifikasi dari semua pohon iterasi pertama, dilakukan voting untuk mencari hasil klasifikasi dengan suara terbanyak. Masukkan data uji pada iterasi kedua dan lakukan hal yang sama. Setelah semua iterasi diisi dengan data uji dan setiap iterasi mendapatkan hasil klasifikasi, bobot ditambahkan ke kelas dengan hasil klasifikasi yang sama di seluruh iterasi. Setelah mengetahui bobot antara kelas pertama dan kedua, prediksi akhir untuk data uji dilakukan dengan memilih kelas dengan bobot tertinggi.

3.8 Evaluasi

Pada langkah ini dilakukan penilaian performa pada tahapan klasifikasi yang telah dilakukan dengan menerapkan algoritma *Random Forest* dan gabungan algoritma *Adaboost* dan *Random Forest*. Evaluasi dilakukan dengan menghitung *accuracy, precision, recall,* dan *specificity*. Setelah didapatkan masing-masing nilai dari keempat metode evaluasi yang digunakan maka selanjutnya pembandingan dapat dilakukan untuk mengetahui manakah algoritma yang menunjukkan performa yang lebih baik apakah algoritma *Random Forest* atau gabungan algoritma *Adaboost* dan *Random Forest.*

4 Hasil dan Pembahasan

Penelitian ini memakai data dari website *kaggle.com* dengan link berikut: *https://www.kaggle.com/datasets/mysarahmadbhat/lung-cancer*. Data *lung cancer* ini di-*upload* oleh Mysar Ahmad Bhat terdiri dari 309 set data yang terdiri dari 15 variabel bebas (fitur) dan 1 variabel terikat (kelas). Setelah memeriksa validitas fitur dengan dokter spesialis paru, hanya 10 dari 15 fitur paru awal yang ditetapkan sebagai indikasi dan pemicu kanker paru-paru. Tabel 1 di bawah ini lebih lanjut menjelaskan 10 karakteristik yang dimaksud.

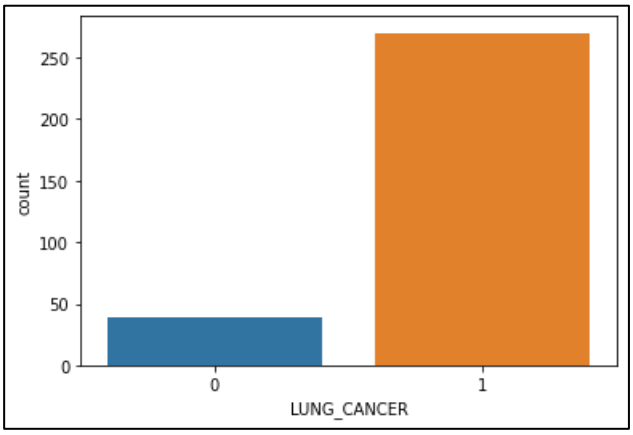
Tabel 1. Variabel Setelah Verifikasi

No.	Variabel	Keterangan	Jenis	Nilai
FITUR				
1.	*Gender*	Jenis kelamin	Kategorikal	M : Laki-laki F : Perempuan
2.	*Age*	Umur	Numerik	21 - 87
3.	*Smoking*	Merokok	Numerik	1 :Tidak 2 : Ya
4.	*Yellow Fingers*	Jari kuning	Numerik	1 :Tidak 2 : Ya
5.	*Chronic Disease*	Riwayat penyakit kronis	Numerik	1 :Tidak 2 : Ya

6.	*Wheezing*	Mengi (suara suara siulan saat bernafas)	Numerik	1 :Tidak 2 : Ya
7.	*Coughing*	Batuk	Numerik	1 :Tidak 2 : Ya
8.	*Shortness of Breath*	Sesak nafas	Numerik	1 :Tidak 2 : Ya
9.	*Swallowing Difficulty*	Kesulitan menelan	Numerik	1 :Tidak 2 : Ya
10.	*Chest Pain*	Nyeri dada	Numerik	1 :Tidak 2 : Ya
KELAS				
11.	*Lung Cancer*	Kanker paru	Kategorikal	Yes No

4.1 Praproses Data

Sebelum melanjutkan ke tahap transformasi data, langkah pertama dalam *preprocessing* data adalah membersihkan data. Dimulai dengan memeriksa nilai yang hilang dan diakhiri dengan memeriksa kompatibilitas tipe data dari setiap variabel. Setelah diperiksa, diketahui bahwa tidak ada nilai yang hilang dalam data dan tipe data yang dikenali juga konsisten dengan semua variabel. Kemudian data tersebut ditransformasikan. Artinya, variabel diubah dari kategorik ke numerik. Karena data memiliki dua variabel kategori, yaitu fitur *"Gender"* dan kelas *"Lung Cancer"*, maka transformasi yang terjadi adalah fitur *"Gender"*, di mana 'F' menjadi '0' dan 'M' menjadi '1'. Kemudian, di kelas *"Lung Cancer"*, *"NO"* menjadi "0" dan *"YES"* menjadi "1". Setelah transformasi diterapkan ke variabel di data *Lung Cancer* diketahui kelasnya tidak seimbang, oleh karena itu data yang dipakai tergolong data tidak seimbang.



Gambar 2. Perbandingan Kelas pada Atribut *Lung Cancer*

Berdasarkan Gambar 2 di atas ada sebanyak 39 *record* kelas '0' atau *"NO"*. Sedangkan untuk kelas "1" atau "YES" ada sebanyak 270 *record*. *Oversampling* dilakukan untuk mengatasi masalah ketidakseimbangan data ini. Teknik ini bekerja dengan meningkatkan kelas minoritas (dalam hal ini kelas "0") sampai jumlah data sama dengan kelas mayoritas (dalam hal ini kelas "1"). Teknik *oversampling* yang akan diimplementasikan pada data adalah SMOTE. Setelah teknik SMOTE diimplementasikan, kelas '0' meningkat hingga proporsi *record* persis sama dengan kelas '1' yaitu menghasilkan 270 *record*. Setelah *preprocessing* data telah diterapkan dan data siap digunakan, maka selanjutnya memasuki tahap klasifikasi. Klasifikasi dilakukan dengan dua cara sesuai dengan algoritma yang telah ditentukan, yaitu *Adaboost* dan *Random Forest*, dan klasifikasi dilakukan terhadap kumpulan data 10 fitur yang dipilih oleh dokter spesialis paru.

4.2 Klasifikasi dengan *Random Forest*

Sebelum penggunaan gabungan *Adaboost* dengan *Random Forest* untuk melakukan tahap klasifikasi, terlebih dahulu kita melakukan klasifikasi dengan algoritma *Random Forest*. Hal ini dilakukan untuk membandingkan hasil komputasi pengimplementasian algoritma *Random Forest* saja dan *Adaboost* yang dikombinasikan dengan *Random Forest*. Membangun model melibatkan berbagai parameter, masing-masing dengan nilai bawaannya sendiri.

```
Clf = RandomForestClassifier(n_estimators = 100,
                             max_depth = None,
                             random_state – None,)
```

Cuplikan kode diatas merupakan kode saat membangun model *Random Forest* menggunakan nilai bawaan untuk setiap parameter. Pada penelitian ini menggunakan kurva validasi, eksperimen dilakukan dengan mengganti nilai bawaan parameter *Random Forest* yaitu *n_estimators, max_depth, dan random _state*. Tabel 2 di bawah ini menunjukkan skenario eksperimen yang diterapkan.

Tabel 2. Skenario Percobaan *Random Forest*

Skenario	***Training Set* (%)**	***Testing Set* (%)**	**Parameter**
Percobaan 1	70	30	Nilai Bawaan
	80	20	
Percobaan 2	70	30	Hasil *Validation Curve*
	80	20	

4.2.1 Skenario Percobaan 1

Percobaan 1 dijalankan dengan mengimplementasikan seluruh nilai pada parameter dengan nilai bawaan model *Random Forest* yang terbentuk. Tabel 3 di bawah ini mencantumkan hasil Percobaan 1 yang dilakukan.

Tabel 3. Hasil Evaluasi Percobaan 1 *Random Forest*

Skenario	**Hasil Pengukuran**					
	Train Size	***Test Size***	***Precision***	***Recall***	***Specificity***	***Accuracy***
Pengujian 1	70%	30%	0.897	0.946	0.945	0.926
Pengujian 2	80%	20%	0.927	0.927	0.927	0.926

4.2.2 Skenario Percobaan 2

Percobaan 2 dijalankan dengan mengimplementasikan seluruh nilai pada parameter dengan nilai hasil dari kurva validasi yang diterapkan pada empat parameter pada *Random Forest.*

```
train_score, test_score = validation_curve(RandomForestClassifier(),
                                           X = x_train,
                                           y = y_train,
                                           param_name="random_state",
                                           param_range=param_range,
                                           cv=3,
                                           scoring="accuracy")
```

Potongan kode diatas merupakan kode untuk membangun kurva validasi dimana *param_name* diisi dengan parameter yang akan dites. Prosedur tersebut akan menghasilkan *output* berupa grafik yang menunjukkan seberapa baik kemampuan model pada berbagai nilai parameter. Kombinasi nilai baru untuk parameter *Random Forest* yang ada berdasarkan proses yang dijalankan menggunakan pemisahan data 70:30 dan 80:20 dijelaskan pada Tabel 4 di bawah ini.

Tabel 4. Kombinasi Nilai Baru Parameter *Random Forest*

Skenario	*Train Size*	*Test Size*	Nilai Parameter		
			N_estimators	***Max_depth***	***Random_state***
Pengujian 1	70%	30%	118	20	227
Pengujian 2	80%	20%	147	5	95

Setelah nilai parameter baru ditentukan maka selanjutnya adalah proses membangun model klasifikasi *Random Forest* termasuk menginisialisasi setiap nilai dengan parameternya masing-masing. Performa yang dihasilkan dari model *Random Forest* yang menggunakan nilai parameter baru dideskripsikan pada Tabel 5.

Tabel 5. Hasil Evaluasi Percobaan 2 *Random Forest*

Skenario	Hasil Pengukuran					
	Train Size	***Test Size***	***Precision***	***Recall***	***Specificity***	***Accuracy***
Pengujian 1	70%	30%	0.897	0.96	0.959	0.932
Pengujian 2	80%	20%	0.927	0.927	0.927	0.926

4.3 Klasifikasi dengan *Adaboost dan Random Forest*

Tahapan klasifikasi algoritma *Adaboost* diimplementasikan dengan *Random Forest* menjadi *base estimator. Base estimator* adalah pembelajaran lemah yang berfungsi dalam melatih model.

```
ada = AdaBoostClassifier(base_estimator=RandomForestClassifier(),
                         n_estimators= 50)
```

Cuplikan kode diatas merupakan kode saat membangun model *Adaboost* menggunakan nilai bawaan untuk setiap parameter. Pada penelitian ini menggunakan kurva validasi, eksperimen dilakukan dengan mengganti nilai bawaan parameter *Adaboost* yaitu *n_estimators*.

4.3.1 Skenario Percobaan 1

Percobaan 1 dijalankan dengan mengimplementasikan nilai pada parameter dengan nilai bawaan model *Adaboost* yang terbentuk dalam hal ini nilai bawaan parameter *n_estimators* ialah 50. Tabel 6 di bawah ini mencantumkan hasil Percobaan 1 yang dilakukan.

Tabel 6. Hasil Evaluasi Percobaan 1 *Adaboost dan Random Forest*

Skenario	Hasil Pengukuran					
	Train Size	***Test Size***	***Precision***	***Recall***	***Specificity***	***Accuracy***
Pengujian 1	70%	30%	0.960	0.923	0.960	0.944
Pengujian 2	80%	20%	0.945	0.945	0.945	0.944

4.3.2 Skenario Percobaan 2

Percobaan 2 dijalankan dengan mengimplementasikan nilai pada parameter dengan nilai hasil dari kurva validasi yang diterapkan pada parameter *n_estimators* di *Adaboost.*

```
train_score, test_score = validation_curve(AdaBoostClassifier(),
                                           X = x_train,
                                           y = y_train,
                                           param_name="n_estimators",
```

```
param_range=param_range,
cv=3,
scoring="accuracy")
```

Potongan kode diatas merupakan kode untuk membangun kurva validasi dimana *param_name* diisi dengan parameter yang akan dites. Prosedur tersebut akan menghasilkan *output* berupa grafik yang menunjukkan seberapa baik kemampuan model pada berbagai nilai parameter. Sesudah nilai *n_estimators* baru ditentukan, maka langkah selanjutnya ialah setup model klasifikasi *Adaboost* dan tetapkan nilai baru pada parameter *n_estimators* dan *base estimator* yang dipakai pada pengujian ini akan mengimplementasikan model *Random Forest* yang memiliki nilai parameter sudah dimodifikasi sesuai dengan kurva validasi percobaan *Random Forest* yang sudah diterapkan sebelumnya. Tabel 7 di bawah ini menggambarkan parameter yang dipakai pada percobaan 2.

Tabel 7. Kombinasi Nilai Baru Parameter *Adaboost* dan *Random Forest*

Skenario	***Train Size***	***Test Size***	**Nilai Parameter**			
			Random Forest			***Adaboost***
			N_estimators	***Max_depth***	***Random_state***	***N_estimators***
Pengujian 1	70%	30 %	118	20	227	135
Pengujian 2	80%	20 %	147	5	95	154

Performa yang dihasilkan dari model *Adaboost* yang menggunakan nilai parameter baru dideskripsikan pada Tabel 8.

Tabel 8. Hasil Evaluasi Percobaan 2 *Adaboost* dan *Random Forest*

Skenario	**Hasil Pengukuran**					
	Train Size	***Test Size***	***Precision***	***Recall***	***Specificity***	***Accuracy***
Pengujian 1	70%	30%	0.956	0.91	0.959	0.938
Pengujian 2	80%	20%	0.946	0.963	0.946	0.954

4.4 Pembahasan

Dalam penelitian ini, penulis menerapkan algoritma *Random Forest* dan gabungan algoritma *Adaboost* dengan *Random Forest* dari 10 fitur yang dipilih oleh dokter spesialis paru. Dalam percobaan 1, klasifikasi diterapkan menggunakan model di mana seluruh parameter menerapkan nilai defaultnya. Di sisi lain pada percobaan 2, kedua model yaitu *Random Forest* dan Adaboost melakukan klasifikasi dengan model yang mengandung nilai parameter yang berubah. Setiap percobaan menjalankan dua pengujian, pengujian 1 membagi data menjadi 70% data latih dan 30% data uji, dan pengujian 2 membagi data menjadi 80% data latih dan 20% data uji . Setelah delapan percobaan, Tabel 9 di bawah ini menunjukkan nilai kinerja tertinggi pada setiap metode penilaian di kedua algoritma.

Tabel 9. Hasil Performa Terbaik

Performa	***Random Forest* (%)**	***Adaboost* + *Random Forest* (%)**
Accuracy tertinggi	93.20%	95.4%

Precision tertinggi	92.70%	96.00%
Recall tertinggi	96.00%	96.30%
Specificity tertinggi	95.90%	96.00%

Tabel 9 menunjukkan bahwa gabungan teknik *Adaboost* dan *Random Forest* mencapai hasil terbaik. Hal ini dikarenakan nilai performansi terbaik untuk *accuracy*, *precision*, *recall*, dan *specificity* ditunjukkan pada gabungan *Adaboost* dan *Random Forest*.

5 Kesimpulan dan Saran

Setelah berbagai macam langkah penelitian sudah diimplementasikan dan dijelaskan, penulis dapat menyimpulkan bahwa implementasi gabungan algoritma klasifikasi *Adaboost* dan *Random Forest* telah terbukti berkinerja lebih baik daripada algoritma klasifikasi *Random Forest* saja. Hal ini dibuktikan dengan gabungan teknik *Adaboost* dan *Random Forest* memberikan performa terbaik dalam hal *accuracy*, *precision*, *recall*, dan *specificity*. *Accuracy* tertinggi 95,4%, *precision* tertinggi 96,00%, *recall* tertinggi 96,30%, dan *specificity* tertinggi 96,00%. Selanjutnya diharapkan penelitian ini dapat dikembangkan lebih baik lagi di masa mendatang dengan mengimplementasikan gabungan algoritma *Adaboost* dan *Random Forest* pada *dataset* lain yang memiliki *record* lebih banyak, untuk melihat apakah kinerja yang didapatkan lebih baik atau sama.

Referensi

[1] Indonesia, K. K. R., Indonesia, P. D. S. O. R., Indonesia, I. A. P. A., Fisik, P. D. S. K., & Indonesia, R. (2016). Pedoman Nasional Pelayanan Kedokteran Kanker Paru. *Jakarta: Kementerian Kesehatan Republik Indonesia*, 1-3.

[2] Thongkam, J., Xu, G., & Zhang, Y. (2008, June). AdaBoost algorithm with random forests for predicting breast cancer survivability. In 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence) (pp. 3062-3069). IEEE.

[3] Byna, A., & Basit, M. (2020). Penerapan Metode Adaboost Untuk Mengoptimasi Prediksi Penyakit Stroke Dengan Algoritma Naïve Bayes. *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, *9*(3), 407-411.

[4] Puskesmaskutautara. 2016. "Kanker Paru", https://dikes.badungkab.go.id/puskesmaskutautara/artikel/read/127/KANKER-PARU-PARU.html, diakses pada 19 November 2021.

[5] Bakti, I. S., & Ivandari, I. (2019). MODEL PREDIKSI PENYAKIT DIABETES MENGGUNAKAN BAYESIAN CLASSIFICATION DAN INFORMATION GAIN UNTUK SELEKSI FITUR DAN ADAPTIVE BOOSTING UNTUK PEMBOBOTAN DATA. *IC-Tech*, *14*(1).

[6] Gan, J. Y., Cao, X. H., & Zeng, J. Y. (2010, October). Combining heritance adaboost and random forests for face detection. In *IEEE 10th INTERNATIONAL CONFERENCE ON SIGNAL PROCESSING PROCEEDINGS* (pp. 666-669). IEEE.

[7] Qalbi Fajar Islami, A. (2020). *Implementasi Algoritma Random Forest Menggunakan TF-IDF untuk Analisis Sentimen dengan Penerapan Transfer Learning* (Doctoral dissertation, Universitas Multimedia Nusantara).

[8] Amiarrahman, M. R., & Handhika, T. (2018). Analisis dan implementasi algoritma klasifikasi Random Forest dalam pengenalan Bahasa Isyarat Indonesia (BISINDO). In *Prosiding SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi)* (Vol. 2, No. 1, pp. 083-088).

[9] Sofyan, S., & Prasetyo, A. (2021, November). Penerapan Synthetic Minority Oversampling Technique (SMOTE) Terhadap Data Tidak Seimbang Pada Tingkat Pendapatan Pekerja Informal Di Provinsi DI Yogyakarta Tahun 2019. In *Seminar Nasional Official Statistics* (Vol. 2021, No. 1, pp. 868-877).