LAPORAN AKADEMIK

TAHUN ANGGARAN 2024

Identifikasi Faktor Resiko Stunting pada Anak di Bawah Lima Tahun di Indonesia Menggunakan *Logistics Regression Ensembles* (LORENS)

Nomor DIPA	:	DIPA BLU-DIPA
Tanggal	:	23 September 2024
Satker	:	(423812) UIN Maulana Malik Ibrahim Malang
Kode Kegiatan	:	(2132) Peningkatan Akses, Mutu, Relevansi dan Daya Saing Pendidikan Tinggi Keagamaan Islam
Kode Output Kegiatan	:	(BEI) Bantuan Lembaga
Sub Output Kegiatan	:	(003) BOPTN
Kode Komponen	:	(004) Dukungan Operasional Penyelenggaraan Pendidikan
Kode Sub Komponen	:	SB Penelitian Pembinaan/Peningkatan Kapasitas **SC Penelitian Dasar Pengembangan Program Studi** SD Penelitian Dasar Interdisipliner SE Penelitian Terapan Pengembangan Nasional SH Penulisan dan Penerbitan Buku Berbasis Riset dan ebook

Oleh:
Usman Pagalay (196504142003121001)
Muhammad Khudzaifah (19900511201608011057)
Ria Dhea Layla N.K (19900709201802012228)

KEMENTERIAN AGAMA
LEMBAGA PENELITIAN DAN PENGABDIAN KEPADA MASYARAKAT (LP2M)
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG

2024

BAB I
PENDAHULUAN

1.1 Latar Belakang

Indonesia merupakan negara berkembang yang memiliki angka prevalensi stunting urutan tertinggi ke-27 dari 154 negara di dunia (Riskesdas, 2018). Dimana nilai ini mencapai 30,8% yang artinya satu dari tiga balita mengalami stunting. Sedangkan di kasus Stunting Indonesia menduduki peringkat ke-2 tertinggi stunting di Kawasan Asia Tenggara. Stunting merupakan gangguan pertumbuhan serta perkembangan pada anak yang diakibatkan kekurangan gizi kronis dan infeksi berulang (WHO, 2015). Stunting dapat ditandai dengan panjang atau tinggi badan di bawah standar menurut usia yang kurang dari -2 standar deviasi pada kurva pertumbuhan standar dari WHO. Hal ini terjadi karena kondisi irreversibel akibat asupan nutrisi yang tidak memenuhi syarat atau disebut dengat adekuat sehingga menyebabkan infeksi berulang atau kloning yang terjadi pada 1000 hari pertama kelahiran (Kemenkes, 2018). Masalah Stunting ini dapat mempengaruhi beberapa masalah dikemudian hari apabila tidak ditindaklanjuti. Misalnya seperti masalah kesehatan jangka panjang yaitu gagal tumbuh, gangguan metabolik saat dewasa (obesitas, diabetes, stroke, jantung dan lain-lain). Dampak lainya yaitu pada ekonomi karena stunting dapat menimbulkan kerugian setiap tahunnya 2-3% GDP. Pemerintah Indonesia terus berupaya melakukan penanganan stunting agar tidak menimbulkan masalah nantinya. Sehingga masalah stunting telah menjadi masalah nasional yang diperhatikan oleh banyak pihak. Penyebab utama stunting antara lain asupan kalori yang tidak adekuat. Selain itu kebutuhan yang meningkat dimana yang banyak terjadi pada anak di bawah usia 5 tahun. Stunting dapat dicegah dengan cara melakukan skrining anemia saat masa remaja serta konsumsi tablet tambah darah. Cara lainnya adalah pada saat kehamilan ibu sebaiknya memeriksakan kondisi kehamilan ke pusat kesehatan terdekat secara rutin, asupan nutrisi yang baik saat kehamilan harus diperhatikan terutama yang mengandung mineral, zat besi, asam folat, yodium. Saat balita, ibu setelah bayi lahir melakukan IMD atau Inisiasi Menyusui Dini, melakukan pemeriksaan ke Posyandu,

Puskesmas dan dokter secara berkala untuk memantau pertumbuhan dan perkembangan anak. Melakukan imunisasi juga merupakan bentuk pencegahan stunting, ASI eksklusif, serta menerapkan gaya hidup bersih dan sehat agar anak tidak mudah sakit (Ayo Sehat Kemenkes, 2022).

Meskipun angka stunting di Indonesia menurun dari 29% pada tahun 2015 menjadi 27,6% dari tahun lalu namun angka tersebut masih di atas batas yang terapkan oleh WHO yaitu 20% (Riskesdas 2018). Pemerintah menargetkan tahun 2024 angka stunting turun hingga 14% pada tahun 2024. Namun stunting di tahun 2021 mencapai 24%, artinya 1 dari 4 anak Indonesia mengalami stunting kurang lebih terdapat lima juta anak yang mengalami stunting (Studi Status Gizi Sehat, 2021).

Logistic Regression Ensemble (LORENS) merupakan salah satu metode klasifikasi pada *machine learning* dengan menggunakanakan teknik *ensemble* (penggabungan). Tujuan dari metode klasifikasi ini memperulah fungsi diskriminan yang optinal sehingga dapat memisahkan pengamatan yang berasal dari kelas berbeda. Dengan cara memperoleh aturan yang dapat digunakan untuk menentukan kategori pada setiap pengamatan yang baru. Metode ini dikembangkan dari Regresi Logistik dengan menggunakan algoritma CERP atau *Classification by Ensembles from Random Partition* (Lim, 2007).

1.2 Rumusan Masalah

Rumusan masalah yang akan dibahas pada penelitian ini yaitu

1. Bagaimana hasil klasifikasi penanganan stunting di Indonesia dengan menggunakan metode LORENS?
2. Bagaimana hasil ketepatan klasifikasi stunting dengan menggunakan LORENS?

1.3 Tujuan

Tujuan penelitian berdasarkan latar belakang serta rumusan masalah pada penelitian ini sebagai berikut

1. Mengetahui hasil klasifikasi penanganan stunting di Indonesia dengan menggunakan metode LORENS

2. Mengetahui hasil ketepatan klasifikasi stunting dengan menggunakan LORENS

1.4 Manfaat Penelitian

Manfaat peneltian sesuai dengan rumusan masalah yaitu memberikan kontribusi berupa masukan serta evaluasi kepada pihak terkait hasil klasfikasi penanganan stunting yang perlu ditangani terlebih dahulu dari model LORENS yang dihasilkan. Diharapkan dapat menekan angka stunting di Indonesia sehingga dapat memenuhi target penurunan stunting.

1.5 Batasan Penelitian

Batasan penelitian ukuran mencari model terbaik berdasarkan nilai akurasi, *sensitivity* dan *specitivity*. Banyaknya partisi dan ensembel model yaitu masing-masing yaitu 10 partisi dan 10 ensembel.

BAB II
KAJIAN TEORI

2.1. Regresi Logistik Ordinal

Model regresi logistik merupakan bagian dari model linear umum atau Generalized Linear Models (GLM). Model ini juga dikenal sebagai model logit. Model logit digunakan untuk memodelkan hubungan antara variabel respon kategori dan variabel prediktor yang bisa berupa kategori maupun kontinu. Jika variabel respon memiliki dua kategori, model ini disebut regresi logistik dikotomus atau biner (Hosmer & Lemeshow, 2000). Namun, jika variabel respon memiliki lebih dari dua kategori, model ini disebut regresi logistik politomus. Regresi logistik merupakan regresi nonlinier dan dapat digunakan untuk mengetahui hubungan antara peubah prediktor serta peubah respon yang memiliki sifat yang tidak linier (Agresti, 2013). Jika kategori pada variabel respon memiliki urutan atau tingkatan (skala ordinal), model tersebut dikenal sebagai regresi logistik ordinal.

Regresi Logistik Ordinal memiliki respon tiga kategori yang mempunyai urutan yang bermakna namun interval antar kategori tidak selalu sama. Misalnya tinggi, sedang dan rendah. Dapat dinotasikan dengan $y = 1$ untuk kategori "tinggi" sedangkan $y = 2$ adalah kategori "sedang" dan $y = 3$ adalah kategori "rendah". Model untuk regresi logistik ordinal merupakan model logit kumulatif. Sehingga model logit tersebut memiliki sifat ordinal dari respon Y dimana didefinisikan dalam peluang kumulatif sehingga model logit tersebut didapat dengan membandingkan peluang kumulatif. Dimana peluang tersbeut merupakan peluang kurang dari atau sam dengan (Hosmer & Lemeshow, 2000). Model regresi logistik ordinal atau logit kumulatif sebagai berikut

$$\text{Logit}\,[P(Y \leq j)] = \log\left[\frac{P(Y \leq j)}{1 - P(Y \leq j)}\right]$$

$$= \left(\alpha_j - \beta^T X\right)$$

Dimana $\beta^T X$ merupakan kombinasi liner pada varoabel prediktor $X = X_1, X_2, \ldots, X_p$ pada setiap model logistik ordinal adalah sama. Sedangkan $P(Y \leq j)$ merupakan peluang kumulatif pada variabel respon Y dimana merupakan kategori

j atau kelas di bawahnya. Nilai α_j merepresentasikan intersep atau *threshold* pada tiap kategori logit kumulatif ke-j dimana pada tiap-tiap kategori j.

Log odd pada peluang kumulatif yang memiki kategori model j atau kurang dapat dimodelkan sebagai kombinasi kombinasi linear variabel prediktor. Namun peluang tersebut merupakan fungsi eksponensial dari prediktor. Berdasarkan persamaan (1) dapat diperoleh fungsi regresi logistik sebagai berikut

$$\log\left[\frac{P(Y\leq j)}{1-P(Y\leq j)}\right] = \log\left[\frac{P(Y\leq j)}{P(Y>j)}\right] = \exp\alpha_j - \beta^T X \tag{2.2}$$

Pada regresi logistik ordinal kejadian tunggal dimodelkan dengan peluang kumulatif. Artinya model tersebut memiliki pusat pada peluang yang berada dalam kategori j atau lebih rendah dibanding dengan berada dalam kategori yang lebih tinggi. Peluang yang berada dalam lategori yang lebih rendah atau di bawahnya dimodelkan sebagai fungsi ekspenensial dari prediktor $\beta^T X$. Artinya peluang dapat berubah secara perkalian dengan perubahan prediktor yang serupa dengan regresi logistik biner. Pada model proporsional odds, koefesien β sama pada semua ketegori. Dimana merupakan asumsi utama pada peluang proporsional, artinya pengaruh prediktor pada peluang berada pada kategori yang lebih rendah dibanding dengan kategori yang lebih tinggi adalah konstan pada semua nilai *threshold* atau ambang batasnya. Sehingga, dari situ menunjukkan model regresi logistik menggambarkan peluang atau resiko pada suatu objek. Berikut model regresi logistik sebagai berikut

$$\pi_j(x_i) = \frac{e^{\beta_0+\beta_1 x_{i1}+\cdots+\beta_p x_{ip}}}{1+e^{\beta_0+\beta_1 x_{i1}+\cdots+\beta_p x_{ip}}} = \frac{1}{1+e^{\beta_0+\beta_1 x_{i1}+\cdots+\beta_p x_{ip}}} \tag{2.3}$$

dimana, p merupakan banyaknya variabel prediktor.

Hubungan antara variabel prediktor dan respon pada metode ini merupakan sebuah fungsi non linear. Sehingga dari persamaan (3) dapat ditransformasikan ke bentuk linier dengan kata lain pada regresi logistik disebut dengan fungsi transformasi logit dari $\pi_j(x_i)$ seperti pada persamaan (4)

$$Logit\left[\pi_j(x_i)\right] = \ln\left(\frac{\pi_j(x_i)}{1-\pi_j(x_i)}\right) = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} \tag{2.4}$$

2.2. Estimasi Parameter

Salah satu estimasi parameter pada regresi logistic yaitu metode MLE (*Maximum Likelihood*). Metode ini mengestimasi β dengan menaksimalkan fungsi *likelihood* (Hosmer dan Lemeshow, 2000). Salah satu syarat yang dipenuhi bila menggunakan MLE yaitu data yang dimiliki suatu distribusi tertentu. Liklihood merupakan produk dari peluang untuk setiap pengamatan dalam dataset. Misal sebuah observasi n dengan variabel prediktor adalah X dan kategori respon ordinal Y, maka likelihoodnya ditulis pada persamaan (2.5) berikut.

$$L(\beta, \alpha) = \prod_{i=1}^{n} P(Y_i = y_i | X_i, \beta, \alpha) \quad (2.5)$$

Dimana $P(Y_i = y_i | X_i, \beta, \alpha)$ merupakan peluang pengamtan ke- i dan menghasilkan kategori y_i. Karena seringkali fungsi likelihood sering kali sangat kecil, maka dapat menggunakan log-likelihood yang merupakan logaritma dari fungsi likelihood seperti pada persamaan (2.6)

$$\log(\beta, \alpha) = \sum_{i=1}^{n} \log\big(P(Y_i = y_i | X_i, \beta, \alpha)\big) \quad (2.6)$$

Tujuan dari estimasi MLE adalah untuk memaksimalkan fungsi log-lokelihood terhadap parameter β dan α.

2.3. LORENS

LORENS atau *Logistic Regression Ensemble* merupakan metode klasifikasi dikembangkan oleh Lim tahun 2010. Metode ini didasarkan pada model regresi logistik yang dikembangkan berdasarkan algoritma LR CERP (*Logistic Regression Classification by Ensembles from Random Partitions*). Pada algoritma LR CERP melakukan pengklasifikasian menggunakan ensemble dengan regresi logistik sebagai basis pengklasifikasian. Algoritma LR CERP dipartisi random menjadi beberapa sub ruang dimana masing-masing sub ruang memiliki sifat *mutually exclusive.* Dimana pada sebuah ruang prediktor dimisalkan Θ yang dipartisi menjadi beberapa sub ruang $((\Theta_1, \Theta_2, \dots, \Theta_n)$ dan saling *mutually exclusive* dan memiliki ukuran yang sama. (Melasasi, 2015) pada masing-masing sub ruang dapat diasumsikan tidak ada bias.

Sebelum menentukan variabel dalam partisi perlu menentukan jumlah partisi yang paling optimal. Karena performa LR CERP disesuaikan dengan banyaknya

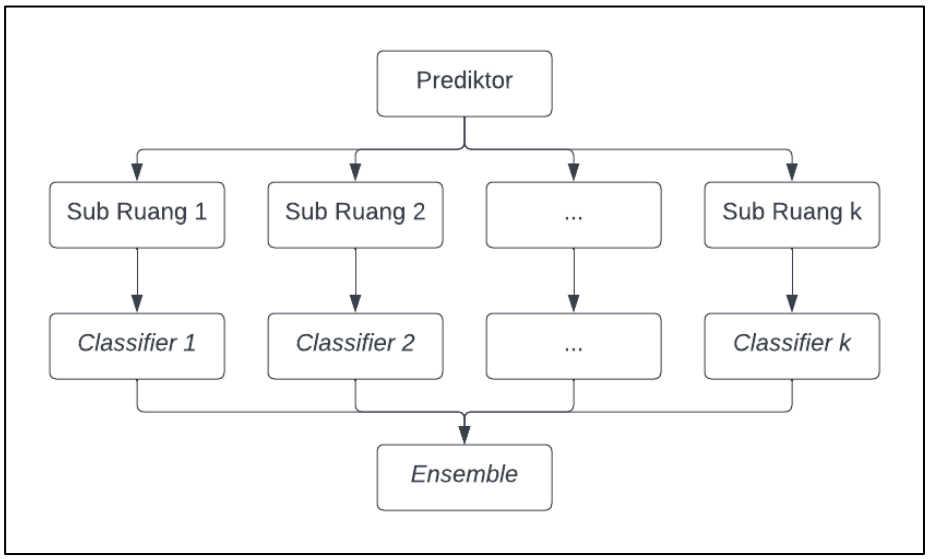
variabel prediktor pada tiap partisinya. Penentuan partisi dapat dipengaruhi oleh banyaknya data. Data yang memiliki jumlah data atau *n* lebih besar dibanding variabel prediktor *p* diperoleh menggunakan persamaan (2.7) berikut

$$K = \frac{P}{i} \tag{2.7}$$

nilai i merupakan bilanan integer yang memiliki nilai kurang dari *n*. Sedangkan data yang memiliki jumlah data *n* lebih kecil dibandung banyak variabel prediktor atau p diperoleh partisi paling optimal dengan menggunakan persamaan (2.8) sebagai berikut

$$K = \frac{6 \times p}{n} \tag{2.8}$$

Gambar 1. berikut merupakan ilustrasi algoritma pada LR CERP



Gambar 1. Ilustrasi LR CERP

Model klasifikasi dibentuk pada setiap sub ruang menggunakan regresi logistik, namun model tersebut lemah terhadap pemilihan variabel (Lim, 2007). Sehingga untuk meningkatan akurasi hasil klasifikasi pada tiap sub ruang yang dibentuk dikombinasikan dengan menggunakan LR CERP. Sehingga pada beberapa model regresi logistik yang terbentuk dikombinasikan yang dilakukan oleh LR CERP yaitu dengan cara mengambil rata-rata nilai prediksi yang dihasilkan setiap *ensemble* sehingga dapat meningkatkan akurasi. Kemudian nilai prediksi yang dihasilkan dari semua hasil klasifikasi pada sub ruang dihitung nilai rata-

ratanya kemudian di kelompokkan sebagai 1 atau 0 berdasakan *threshold* (Ahn dkk, 2007).

LORENS akan membentuk beberapa pengabungan dengan mengulangi prosedur LR CERP beberapa kali (Widhianingsih, 2018). LORENS akan mempartisi data kedalam sebuah *k* sub ruang dimana sub ruang ini ditentukan dengan secara acak dan distribusi yang sama. Sehingga dapay diasumsikan apabila pemilihan bariabel tidak akan menagalami bias pada tiap sub ruang. Model regresi logistik digunakan pada setiap sub ruang tanpa pemilihan variabel. Setelah itu, partisi variabel yang telah dilakukan secara random diharapkan memperoleh peluang kesalahan yang kecil atau hampir sama dengan nilai *k* klasifikasi. Seingga dengan begitu akan meningkatkan akurasi. Masing-masing model regresi logistik pada setiap partisi dikombinasikan nilai prediksinya sehingga diperoleh akurasi dalam satu *ensemble*. LORENS akan mengisalkan kombinasi rata-rata atau nial yang terbanyak dengan akurasi yang sama menggunakan pengulangan prosedur LR CERP. Sehingga, metode LORENS akan menggunakan hasil rata-rata dari nilai yang lebih unggul dibanding nilai terbanyak.

Hasil *ensemble* LORENS yang dihasilkan memiliki partisi yang berbeda sesuai dengan nilai paling banyak pada tiap penggabungan. Kemudian nilai tersebut diperoleh nilai akurasi secara umum dimana akurasinya ditingkatkan. Peningkatan akurasi diperoleh apabilak jumlah *ensemble* yang dibentuk memiliki nilai lebih dari 10. Regrsi logistik menggunakan proses klasifikasi berdasarkan nilai peluang *threshold.* (Lim dkk, 2010) *threshold* umum yang digunakan pada klasifikasi yaitu 0,5. Namun akurasi klasifikasi tidak memiliki nilai yang tinggi bila proporsi kelasnya 0 dan kelas 1 tidak seimbang (*imbalanced*). Nilai *sensitivity* dan *specitivity* agar seimbang metode LORENS menggunakan *threshold* optimal. Dimana, nilai *threshold* tersebut sebagai berikut

$$Threshold = \frac{\bar{y}+0.5}{2} \tag{2.9}$$

nilai $\bar{y}$ adalah proporsi respon positif pada data.

Ketepatan prediksi dapat digunakan dengan cara membagi jumlah prediksi yang tepat diklasifikasikan dengan total jumlah prediksi. Berikut merupakan salah

satu cara *confussion matrix* yang dapat digunakan untuk mendapatkan ketepatan klasifikasi (Fawcet, 2006)

$$sensitivity = \frac{TP}{(TP+FN)} \quad (2.10)$$

$$Specificity = \frac{TN}{(FP+TN)} \quad (2.11)$$

$$Accuracy = \frac{TP+TN}{(TP+FP+TN+FN)} \quad (2.12)$$

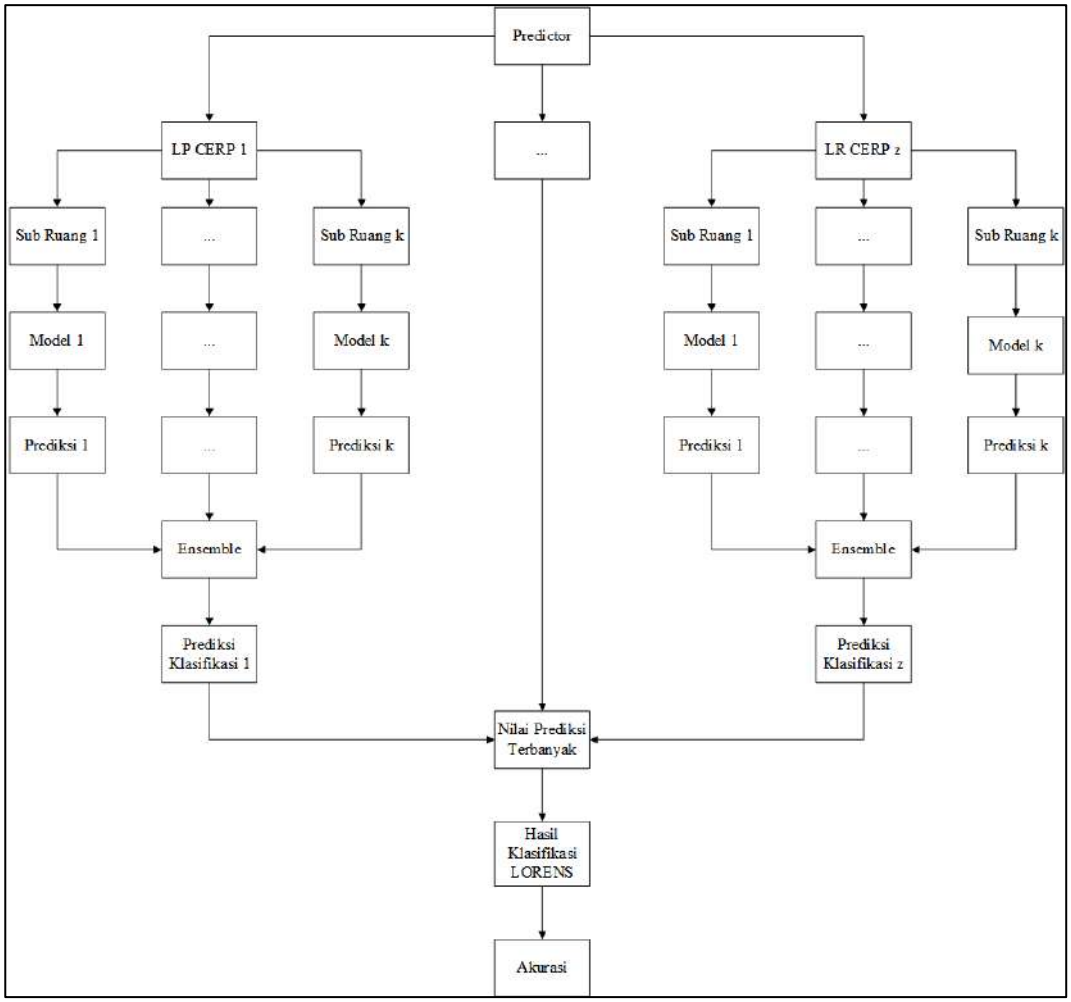
Kesalahan pada klasifikasi dapat diperoleg dengan cara mencari nilai *false positive rate, false negative rate, error.* Berikut merupakan salah satu cara uang digunakan untuk menghitung nilai kesalahan pada klasifikasi

$$false\ positive\ rate = \frac{FP}{(TP+FN)} \quad (2.13)$$

$$false\ negative\ rate = \frac{FN}{(TP+FN)} \quad (2.14)$$

$$error = \frac{(FP+FN)}{(TP+FP+TN+FN)} \quad (2.15)$$

Beberapa *ensemble* dibentuk metode LORENS dan nilai prediksi terbanyak akan terpilih diantara *ensemble.* Hasil prediksi tersebut akan diperoleh satu akurasi yang nilainya telah meningkat berdasarkan beberapa penggabungan. Kelebihan LORENS pada model klasifikasi yaitu bebas asumsi pada dimensi data. Karena LORENS akan mempartisi secara random variabel-variabel prediktornya. (Lim, dkk, 2010) LORENS memiliki keunguulan dibanding dengan LR CERP pada basis klasifikasi yang menggunakan regresi logistik. Gambar 2 merupakan ilustrasi metode LORENS.



Gambar 2. Ilustrasi Metode LORENS

2.4. Penelitian Terdahulu

Stunting merupakan kondisi di mana bayi dan balita mengalami gangguan pertumbuhan akibat kekurangan gizi kronis, yang menyebabkan anak memiliki tinggi badan yang lebih pendek dibandingkan dengan usianya. Kekurangan gizi ini terjadi sejak masa kehamilan dan berlanjut pada masa awal kehidupan anak, namun tanda-tanda stunting biasanya baru terlihat setelah anak berusia dua tahun. Anak yang mengalami stunting dikategorikan sebagai balita dengan panjang atau tinggi badan (PB/U atau TB/U) yang lebih rendah dibandingkan dengan standar WHO-MGRS (Multicentre Growth Reference Study) tahun 2006. Menurut Kementerian Kesehatan (Kemenkes), balita dengan nilai z-score <2SD dianggap mengalami stunting, sedangkan z-score <3SD dianggap mengalami stunting berat (Tobing dkk., 2021).

Stunting merupakan salah satu permasalahan gizi utama pada balita di Indonesia yang masih belum sepenuhnya teratasi. Data dari Riset Kesehatan Dasar (Riskesdes) menunjukkan bahwa prevalensi balita stunting dan sangat stunting di Indonesia sebesar 37,2% pada tahun 2013, dan menurun menjadi 30,8% pada tahun 2018. Untuk bayi di bawah dua tahun (baduta), prevalensi stunting menurun dari 32,8% pada tahun 2013 menjadi 29,2% pada tahun 2018. Meskipun terdapat penurunan, menurut standar WHO, prevalensi ini masih tergolong dalam kategori tinggi (>20%) (Kemenkes RI, 2022).

Menurut penelitian (Anggryni dkk., 2021), beberapa faktor penyebab stunting adalah kekurangan nutrisi selama kehamilan, inisiasi menyusui yang kurang dari 1 jam setelah kelahiran, atau bahkan tidak menyusui sama sekali, penghentian pemberian ASI sebelum usia 6 bulan, frekuensi menyusui yang tidak memadai, serta pemberian makanan pendamping ASI (MPASI) sebelum 6 atau 12 bulan dengan variasi, frekuensi, dan tekstur makanan yang tidak sesuai dengan usia anak. Berdasarkan kajian sistematis oleh Beal dkk. yang dianalisis oleh Kemenkes, terdapat empat faktor langsung penyebab stunting di Indonesia yang dijabarkan dalam konsep WHO.

Faktor-faktor yang memengaruhi terjadinya stunting mencakup kondisi keluarga dan rumah tangga, seperti ibu yang memiliki tubuh pendek, kelahiran prematur, panjang badan lahir bayi yang pendek, tingkat pendidikan ibu yang rendah, serta status sosioekonomi yang rendah. Penelitian tersebut juga mengungkapkan beberapa faktor penyebab stunting di Indonesia yang tidak termasuk dalam kerangka konsep WHO, yaitu ayah yang bertubuh pendek, kebiasaan merokok pada orang tua, kepadatan hunian, adanya demam, serta rendahnya cakupan imunisasi (Kemenkes RI, 2022).

Permasalahan stunting membawa dampak jangka pendek dan jangka panjang bagi negara yang dapat menghambat masa depannya. Dampak jangka pendek stunting meliputi gangguan pada perkembangan otak, kecerdasan, pertumbuhan fisik, serta gangguan metabolisme tubuh anak. Sementara itu, dampak jangka panjangnya mencakup penurunan kemampuan kognitif dan prestasi belajar, daya tahan tubuh yang lemah sehingga anak mudah sakit, risiko tinggi terkena penyakit

seperti diabetes, obesitas, penyakit jantung, kanker, stroke, dan disabilitas di usia tua. Selain itu, stunting meningkatkan risiko penyakit dan kematian perinatal-neonatal, menurunkan kualitas kerja yang berujung pada rendahnya kualitas sumber daya manusia (SDM), dan pada akhirnya mengurangi produktivitas ekonomi.

Indeks Khusus Penanganan Stunting (IKPS) adalah instrumen yang dirancang untuk menilai kinerja program percepatan penurunan stunting mulai dari tingkat nasional hingga kabupaten/kota. Tujuan dari penyusunan IKPS adalah untuk mendukung upaya percepatan penurunan stunting di Indonesia. Selain itu, IKPS juga digunakan untuk memenuhi *Disbursement Linked Indicators* (DLI) 8 dari program *Investing in Nutrition and Early Years* (INEY), yang merupakan bentuk kerja sama antara pemerintah Indonesia dan Bank Dunia. Penyusunan IKPS ini mencakup berbagai indikator dan dimensi terkait. Berikut indicator dan dimensi tersebut.

Dimensi kesehatan meliputi imunisasi, persalinan yang ditangani oleh tenaga medis di fasilitas kesehatan, peenggunaan metode Keluarga Berencana (KB) modern. Dimensi gizi mencakup pemberian ASI eksklusif, makanan pendamping ASI (MPASI). Dimensi meliputi perumahan akses terhadap air minum yang layak, ketersediaan sanitasi yang layak. Dimensi pangan meliputi pengalaman kerawanan pangan ketidakcukupan konsumsi pangan. dimensi pendidikan meliputi partisipasi dalam pendidikan anak usia dini (PAUD). Dimensi perlindungan sosial meliputi kepemilikan jaminan kesehatan nasional (JKN) atau jaminan kesehatan daerah (Jamkesda), penerimaan kartu perlindungan Sosial (KPS)/Kartu Sembako Sejahtera (KSS) atau bantuan pangan lainnya. Data untuk indikator-indikator tersebut diperoleh dari Survei Sosial Ekonomi Nasional (Susenas) yang dilaksanakan pada bulan Maret dan Juni, namun data yang digunakan dalam penyusunan IKPS adalah data Susenas Maret dari tahun yang bersangkutan.

Percepatan pengurangan kasus stunting menjadi salah satu program prioritas pemerintah di sektor kesehatan. Stunting tidak hanya memengaruhi pertumbuhan fisik, tetapi juga berdampak pada fungsi penting tubuh lainnya, seperti perkembangan otak dan sistem kekebalan. Akibatnya, balita yang mengalami stunting berpotensi memiliki kecerdasan yang kurang optimal, rentan terhadap

penyakit, serta berisiko mengalami penurunan produktivitas di masa mendatang. Untuk mempercepat penanganan stunting, pemerintah membentuk Indeks Khusus Penanganan Stunting (IKPS) sebagai instrumen evaluasi program penanganan stunting yang ada. Selain itu, IKPS digunakan untuk mengukur sejauh mana intervensi mencapai rumah tangga yang menjadi target (BPS, 2021).

IKPS merupakan alat untuk mengukur pencapaian penanganan stunting di Indonesia. IKPS disusun berdasarkan beberapa dimensi dan indikator yang mengikuti prinsip SMART (*Specific, Measurable, Achievable, Realistic, Timely*, and Simplicity). Salah satu prinsip yang mudah dianalisis dalam indikator IKPS adalah prinsip measurable (dapat diukur), yang berarti bahwa setiap indikator yang digunakan dalam IKPS harus dapat diukur (BPS, 2021).

Indikator IKPS perlu melalui proses normalisasi dengan tujuan untuk memastikan bahwa semua indikator memiliki rentang dan arah yang seragam. Dalam proses normalisasi, terdapat dua langkah utama yang perlu dilakukan, yaitu menentukan nilai minimum dan maksimum untuk setiap indikator agar rentangnya sama, serta menyelaraskan arah masing-masing indikator. Di antara indikator penyusun IKPS, terdapat sepuluh indikator yang memiliki arah positif dan satu indikator dengan arah negatif. Berikut adalah persamaan normalisasi yang digunakan untuk memastikan semua indikator memiliki arah yang seragam (BPS, 2021).

Indikator dengan arah positif

$$KX_i = \frac{X_i - X_{min}}{X_{max} - X_{min}} \times 100$$

Indikator dengan arah negatif

$$KX_i = 100 - \left(\frac{X_i - X_{min}}{X_{max} - X_{min}} \times 100 \right)$$

dimana

KX_i : nilai indicator ternormalisasi

X_i : nilai indikator (empiris)

X_{min} : nilai minimal indikator

X_{max} : nilai maksimal indikator

Selanjutnya, penentuan bobot dimensi dilakukan untuk menghitung IKPS, di mana setiap dimensi memiliki bobot yang sama, yaitu 1/6. Hal ini disebabkan karena IKPS terdiri dari 6 dimensi. Untuk menghitung IKPS, nilai indeks untuk setiap dimensi dihitung terlebih dahulu dengan mengambil rata-rata dari nilai indikator yang terdapat dalam dimensi tersebut. Rumus perhitungan IKPS dapat dijelaskan sebagai berikut:

$$IKPS = \frac{\sum_{z=1}^{p} W_z d_z}{\sum_{z=1}^{p} W_z}$$

dimana

W_z : penimbang dimensi ke- z dengan $z = 1,2, \dots, p$

d_z : dimensi ke- ke- z dengan $z = 1,2, \dots, p$

BAB III
METODOLOGI PENELITIAN

3.1 Jenis Penelitian

Penelitian ini menggunakan metode kualitatif. Penelitian kuantitatif adalah jenis penelitian yang mengumpulkan data dalam jumlah besar. Dalam penelitian kuantitatif, teori ilmiah yang telah diterima sebagai kebenaran digunakan sebagai dasar untuk mencari kebenaran lebih lanjut (Saryono, 2010).

3.2 Data dan Sumber Data

Data yang digunakan pada penelitian ini merupakan data skunder. Data sekunder adalah jenis data yang diperoleh dari sumber yang sudah tersedia. Data penelitian ini diakses melalui situs resmi Badan Pusat Statistik (BPS) Indonesia. Data yang digunakan merupakan data IKPS di 514 Kapbupaten dan Kota di Indonesia. Data yang digunakan mencakup variabel dependen (Y) dan variabel independen (X). Variabel-variabel yang akan digunakan dalam penelitian ini adalah sebagai berikut:

No	Variabel	Skala	Keterangan
1	Indeks Khusus Penanganan Stunting menurut Kabupaten/Kot di Indonesia (Y)	Ordinal	Rendah: 53,9 –65,5 Sedang: 65,6 – 77,2 Tinggi : 77,3 – 88,9
2	Imunisasi (X1)	Rasio	0 - 100
3	Penolong persalinan oleh tenaga kesehatan di fasiitas kesehatan (X2)	Rasio	0 - 100
4	KB modern (X3)	Rasio	0 - 100
5	ASI Eksklusif (X4)	Rasio	0 - 100
6	MPASI (X5)	Rasio	0 - 100
7	Air minum layak (X6)	Rasio	0 - 100
8	Sanitasi layak (X7)	Rasio	0 - 100
9	Pendidikan Anak Usia Dini (PAUD) (X8)	Rasio	0 - 100
10	Pemanfaaatan jaminan kesehatan (X9)	Rasio	0 - 100
11	Penerima KPS/KKS (X10)	Rasio	0 - 100

3.3 Tahapan Penelitian

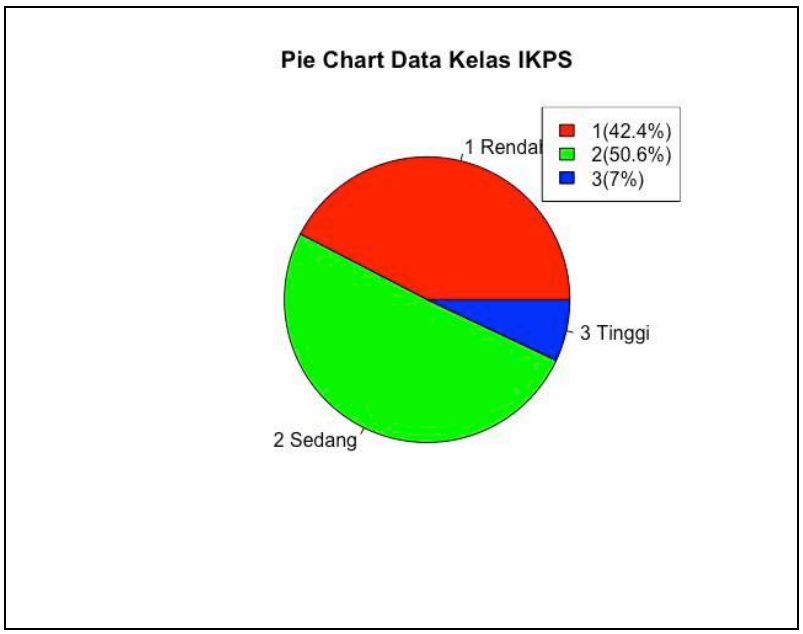
Berikut merupakan tahapan penelitian yang dilakukan pada penelitian ini menggunakan *Logistic Ensemble Regression.*

1. Melakukan statistik deskriptif untuk mengetahui karakteristik data
2. Melakukan *resempling* dengan metode SMOTE untuk mengatasi data *imbalance*
3. Melakukan sub sampling dari kumpulan data atau membagi data menjadi data *training* dan *data testing*. Dimana data *training* pada penelitian ini 85% dari jumlah data dan data *testing* 15%. Data *training* digunakan untuk membentuk model sedangkan data *testing* untuk mengukur tingkat kesalahan.
4. Membentuk jumlah *l* partisi dan *m ensemble* yang akan diterapkan
5. Mempartisi variabel prediktor ke dalam *k* sub ruang
6. Membangun model regresi logistik setiap partisi dari data training
7. Menghitung partisi dengan mensubsitusikan data testing ke model
8. Menghitung nilai rata-ratanya setelah nilai prediksi diperoleh
9. Ulangi langkah 4 sampai 8 hingga terbentuk *n ensemble*
10. Mencari nilai prediksi terbanyak dari pengamatan di semua *ensemble*
11. Menghitung *threshold* yang paling optimal
12. Mengklasifikasikan setiap nilai prediksi menjadi 0 atau 1 berdasarkan nilai *threshold*
13. Menghitung ketepatan hasil klasifikasi dengan nilai *sensitivity, specificity*, dan *accuracy*
14. Melakukan evaluasi performa model dengan menggunakan *cross validation*
15. Menarik kesimpulan

BAB IV
HASIL DAN PEMBAHASAN

4.1 Statistika Deskriptif

Pada data Indeks Khusus Penanganan Stunting (IKPS) terdapat tiga kategori, yaitu IKPS Tnggi, Sedang dan Rendah. Indeks ini merupakan ukuran penanganan kasus stunting di tiap Kabupaten dan Kota di Indonesia. Terdapat 514 data Kota dan Kabupaten yang telah terkumpul dari BPS. Berikut merupakan statistika deskriptif dari IKPS yang ditunjukkan dengan *pie chart*.



Gambar 4.1 *Pie Chart* Indeks Khusus Penanganan Stunting

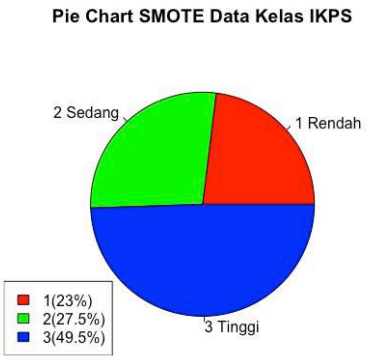
Gambar 4.1 menunjukkan bahwa IKPS Sedang paling tinggi dibanding dengan kategori lain yaitu 50,6%. IKPS Sedang artinya bahwa pemerintah telah melakukan penangan stunting sebesar 65,6% – 77,2% di seluruh Kota dan Kabupaten di Indonesia. Sedangkan yang paling rendah adalah penanganan stunting kategori Tinggi yaitu sebesar 7%. Artiya terdapat 53,9% – 65,5% Kota dan Kabupaten di Provinsi Indonesia yang masih tinggi penangannya atau yang mendapat perhatian tinggi masih paling sedikit dibanding kategori lain. Sehingga, untuk mengetahu faktor resiko yang menjadi perhatian pemerintah pusat adalah melakukan analisis lanjut.

4.1.1 SMOTE (*Synthetic Minority Over-sampling Technique*)

Analisis yang dilakukan pada penelitian ini menggunakan analisis *Logistic Regression Ensemble* (LOREN) dimana metode ini salah satu metode *Machine* Learning. Sebelum melakukan analisis tersebut dilakukan *Synthetic Minority Over-sampling Technique* atau yang dikenal dengan SMOTE karena adanya *imbalance data*. Metode *Machine Learning* menghasilkan performa yang baik pada kasus data yang memiliki keadaan yang seimbang. Sehingga untuk memaksimalkan performa tersebut maka dapat dilakukan *balance* data agar meningkatkan akurasi pada klasifikasi. Salah satu metode yang digunakan untuk mengatasi kasus *imbalance* adalah SMOTE. (Morris dan Yang, 2021) SMOTE memberikan hasil yang memuaskan untuk meningkatkan hasil klasifikasi dengan akurasi 90%.

Metode *oversampling* SMOTE menghasilkan observasi buatan baru dari sampel kelas minoritas yang sudah ada. Selain itu juga menduplikasi data yang ada, SMOTE secara aktif akan menciptakan data baru dengan nilai-nilai yang mendekati sampel minoritas melalui proses augmentasi data. Observasi buatan ini dihasilkan secara acak dengan memilih satu atau lebih tetangga terdekat (*k-nearest neighbors*) untuk setiap sampel kelas minoritas. Setelah proses oversampling selesai, ketidakseimbangan pada dataset dapat teratasi. Sehingga dataset tersebut dapat digunakan untuk menguji berbagai model klasifikasi secara lebih efektif.

Berikut hasil *oversampling* dengan menggunakan SMOTE pada data IKPS Kota dan Kabupaten di Provinsi Indonesia.



Gambar 4.2 Pie Chart Data Kelas IKPS dengan SMOTE

Gambar 4.2 menunjukkan data IKPS setelah dilakukan *oversampling* dengan metode SMOTE. Setelah dilakukan *oversampling* IKPS yang memiliki kategori tinggi menjadi paling tinggi, yaitu sebesar 49,5%. Hasil *oversampling* ini mengambil dari kelas sedang dan rendah. Selain itu, jumlah data menjadi 946 data observasi. Kategori IKPS Rendah pada hasil *oversampling* ini menjadi 218 kategori, kategori IKPS Sedang 260 kategori dan yang terakhir kategori IKPS Tinggi 468 kategori.

4.2 Analisis LORENS

Hasil data yang telah dilakukan *oversampling* dilanjutkan dianalisis dengan metode regresi logistik ensembel. Sebelum dilakukan analisis ini, data akan dibagi menjadi dua, yaitu data *training* dan data *testing*. Dimana data *training* digunakan untuk membentuk model regresi logistik ensembel. Sedangkan data *testing* digunakan untuk memeriksa kesalahan pada model. Pada penelitian ini menggunakan kombinasi data *training* dan *testing* yaitu 85% dan 15%.

Pada penelitian ini menggunakan *l* partisi sebanyak 10. Semakin banyak partisi akan meningkatkan juga untuk resiko *overfitting* apabila yang digunakan adalah data yang besar. Partisi akan memastikan bahwa setiap model regresi logistik pada data *training* beragam subset data dapat meningkatkan kemampuan ensembel untuk melakukan generalisasi. Sedangkan, *m* ensembel yang dilakukan yaitu sebanyak 10 ensembel.

Model regresi logistik yang terbentuk pada sebanyak 10 *ensemble* adalah 10 model maka model terbaik yang terbentuk sebagai berikut

$$logit(P) = \log\left(\frac{P}{1-P}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_{10} X_{10} \quad (4.1)$$

Berikut model terbaik untuk klasifikasi IKPS yang didapatkan dari LOREN dari persamaan (4.1) dimana model tyang terbentuk tanpa memperhatikan variabel yang tidak signifikan

$$P(Y \leq j) = \frac{1}{1+e^{-(\pi_j - \theta)}} \quad (4.2)$$

Dimana

π_j : intersep atau ambang batas untuk kategori *j*

θ : merupakan prediktor linear yang memiliki definisi sebagai $\theta = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_{10} X_{10}$

$\beta_0, \beta_1, \beta_2, \ldots, \beta_{10}$ merupakan koefisien regresi

$X_1, X_2, \ldots, X_{10}$: variabel prediktor

Maka dari persamaan (4.2) didapatkan,

$$\theta = 0{,}0272X_1 + 0{,}0747X_2 + 0{,}0461X_3 + 0{,}0103X_4 + +0{,}137X_5 + 0{,}085X_6 + 0{,}134X_7 + 0{,}216X_8 + 0{,}084X_9 + 0{,}137X_{10} \quad (4.3)$$

Intersep dari model terbaik LORENS didapatkan

$\pi_1 = 60{,}049$ dimana sebagai pemisah antara IKPS rendah (kategori 1) dan IKPS sedang atau tinggi (kategori 2 atau 3)

$\pi_2 = 70{,}71$ dimana sebagai pemisah anatara IKPS rendah atau sedang (kategori 1 atau 2) dan IKPS tinggi (kategori 3).

Fungsi peluang kumulatif untuk masing-masing kategori yaitu,

1. Peluang kumulatif IKPS Rendah (Kategori 1)

   $$P(Y \leq 1) = \frac{1}{1+e^{-(60{,}049-\theta)}} \quad (4.4)$$

2. Peluang kumulatif IKPS Sedang (Kategori 2)

   $$P(Y \leq 2) = \frac{1}{1+e^{-(70{,}718-\theta)}} \quad (4.5)$$

3. Peluang kumulatif IKPS Tinggi (Kategori 3)

   Persamaan (4.4) merupakan peluang IKPS Rendah dengan kategori 1 dimana $P(Y = 1) = P(Y \leq 1)$. Sedangkan persamaan (4.5) merupakan peluang IKPS Sedang dengan kategori 2 dimana $P(Y = 2) = P(Y \leq 2) - P(Y \leq 1)$. Maka untuk peluang IKPS Tinggi atau kategori 3 didapatkan:

   $$P(Y = 3) = 1 - P(Y \leq 2) = 1 - \frac{1}{1 + e^{-(70{,}718-\theta)}}$$

   Persamaan (4.3) memiliki makna sebagai berikut

Imunisasi 0,0272 memiliki koefisien yang kecil karena mendekati nol. Artinya, berarti bahwa peningkatan variabel-variabel ini memiliki pengaruh yang relatif kecil terhadap perubahan kategori respons. Sanitasi 0,134 memilikiki koefesien yangtinggi. Artinya, akses terhadap sanitasi yang lebih baik meningkatkan kemungkinan memiliki IKPS yang lebih tinggi. Daerah dengan sanitasi yang baik

cenderung memiliki kesejahteraan yang lebih baik, sehingga lebih mungkin berada pada kategori IKPS tinggi. 4Jika seseorang memiliki skor tinggi pada variabel PAUD (misalnya, banyak anak-anak yang berpartisipasi dalam program PAUD), maka kemungkinan besar mereka akan berada pada kategori respons yang lebih tinggi (misalnya, tingkat kesehatan yang lebih baik atau lebih banyak akses terhadap layanan kesehatan).

4.2.1 Nilai Prediksi

Nilai prediksi merupakan prediksi yang diperoleh dengan mensubsitusikan data testing pada model yang sebelumnya telah diperoleh. Kemudian perhitungan ini dilakukan pada masing-masing *ensemble*. Berikut hasil prediksi untuk masing-masing *ensemble.*

Tabel 4.1 Nilai Prediksi pada tiap Ensembel Data IKPS

Data ke-	Ensembel									
	1	2	3	4	5	6	7	8	9	10
578	0,388	0,339	0,360	0,421	0,441	0,373	0,306	0,405	0,396	0,361
549	0,415	0,408	0,357	0,423	0,4307	0,405	0,372	0,421	0,369	0,402
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
259	0,271	0,290	0,279	0,280	0,257	0,266	0,281	0,264	0,292	0,290

Tabel 4.1 merupakan hasil prediksi pada data testing terhadap masing-masing ensembel. Pada data testing pertama merupakan data ke-578 dan ensembel pertama dimana nilai prediksinya sebesar 0,288. Sedangkan pada ensemble ke-2 yaitu 0,339 dan seterusnya sampai esembel ke 10. Kemidian pada data ke 2 yaitu data ke-549 ensembel pertama memiliki nilai prediksi sebesar 0,415 dan dilakukan sampai ensembel ke-10 yang memiliki akurasi 0,402.

4.2.2 *Threshold* Optimal

Pada metode LORENS nilai *threshold* dapat dimodifikasi apabila terdapat kelas yang *imbalance* atau tidak seimbang. Umumnya *threshold* dapat menggunakan nilai 0,5. Namun, pada penelitian ini memiliki kelas yang tidak seimbang antara kategori rendah, sedang, dan tinggi. Berikut perhitungan *threshold optimal* yang dilakukan dengan persamaan (2. 9)

$$Threshold = \frac{(\bar{y} + 0{,}5)}{2} = \frac{\left(\frac{182}{804} + 0{,}5\right)}{2} = 0{,}363184$$

Sehingga nila *threshold* optimal yang digunakan dalam penelitian ini yaitu 0,363. Nilai ini digunakan untuk menentukan kelas klasifikasi pada masing-masing prediksi.

4.2.3 Akurasi Model Prediksi

Akurasi model prediksi dapat dilihat menggunakan data *training*. Hasil akurasi dapat diperoleh dari tabulasi silang antara *True Positive* (TP), *False Positive* (FP), *True Negative* (TN,) dan *False Negative* (FN) dengan menggunakan data training. Tabel 4.2 berikut merupakan tabel tabulasi silang dengan 10 partisi dan 10 ensembel

Tabel 4.1 Tabulasi Silang Klasifikasi Aktual dan Klasfikasi Prediksi dengan Data Training

		Kelas Aktual					
		Rendah		**Sedang**		**Tinggi**	
		p(+)	**n(-)**	**p(+)**	**n(-)**	**p(+)**	**n(-)**
Kelas Prediksi	**p(+)**	171	612	203	563	392	395
	n(-)	10	11	17	21	11	6

True Positives (TP = 171) yaitu model berhasil memprediksi 171 kejadian sebagai kelas rendah dengan benar. *False Positives* (FP = 10) yaitu model salah memprediksi 10 kejadian sebagai kelas rendah, padahal sebenarnya bukan. *True Negatives* (TN = 612): Model berhasil memprediksi 612 kejadian sebagai bukan kelas rendah dengan benar. *False Negatives* (FN = 11) yaitu model salah memprediksi 11 kejadian sebagai bukan kelas rendah, padahal sebenarnya adalah kelas rendah. Model cukup baik dalam mengenali kelas IKPS rendah, terbukti dari 171 prediksi yang benar (TP). Namun, model masih kesulitan dalam beberapa kasus, terlihat dari 11 kejadian yang sebenarnya kelas IKPS rendah tetapi diprediksi sebagai kelas lain (FN).

True Positive (TP = 203) artinya model berhasil memprediksi 203 kejadian sebagai kelas 2 dengan benar. *False Positive* (FP = 17) artinya model salah memprediksi 17 kejadian sebagai kelas 2, padahal sebenarnya bukan. *True Negative*

(TN = 563) artinya model berhasil memprediksi 563 kejadian sebagai bukan kelas 2 dengan benar. *False Negative* (FN = 21) artinya model salah memprediksi 21 kejadian sebagai bukan kelas 2, padahal sebenarnya adalah kelas 2. Model cukup baik dalam mengenali kelas 2, dengan 203 prediksi benar (TP). Namun, ada sedikit kesulitan dalam mengenali kelas ini, karena 21 instance yang seharusnya kelas 2 diprediksi sebagai kelas lain (FN), serta 17 kejadian salah diklasifikasikan sebagai kelas 2 (FP).

True Positive (TP = 392) artinya model berhasil memprediksi 392 instance sebagai kelas 3 dengan benar. *False Negative* (FP = 11) artinya model salah memprediksi 11 kejadian sebagai kelas 3, padahal sebenarnya bukan. *True Negative* (TN = 395) artinya model berhasil memprediksi 395 kejadian sebagai bukan kelas 3 dengan benar. *False Negative* (FN = 6) artinya model salah memprediksi 6 kejadian sebagai bukan kelas 3, padahal sebenarnya adalah kelas 3. Model sangat baik dalam mengenali kelas 3, dengan jumlah prediksi benar (TP) yang tinggi yaitu 392 dan kesalahan yang sangat sedikit (FN = 6). Ini menunjukkan kinerja yang sangat kuat dalam mengidentifikasi kelas 3. Kesalahan prediksi positif (FP) juga sangat rendah, hanya 11 kejadian.

Nilai akurasi, s*ensitivity* dan *specifity* dijelaskan pada Tabel 4.2 berikut ini. Nilai ini merupakan nilai metrik yang digunakan untuk mengevaluasi kinerja model klasifikasi. Sensitivitas dapat mengukur kasus positif sebenarnya, yang dapat diidentifikasikan dengan benar oleh model. Sehingga nilai ini sangat penting untuk mengidentifikasi semua kasus positif. Spesifitivitas dapat mengukur proporsi kasus negatif sebenarnya yang berhasil diidentifikasikan benar oleh model. Apabila terdapat *false positive* atau kesalahan positif dapat memiliki dampak besar. Akurasi dapat mengukur proporsi keseluruhan dari semua kasus baik itu positif maupun negatif yang diklasifikan dengan benar oleh model. Akurasi akan bermanfaat apabila jumlah kasusu positif dan negatif seimbang damun akan bermasalah apabila terdapat kasus *imbalanced data*.

Tabel 4.2 Akurasi, *Sensitivity,* dan *Specitivity* Data *Training*

IKPS	Akurasi	*Sensitivity*	*Specitivity*
Rendah	0,9653727	0,9653727	0,9789128
Sedang	0,8874541	0,8874541	0,982162
Tinggi	0,9900974	0,9900974	0,9697035

Tabel 4.2 menunjukkan bahwa nilai akurasi, *sensitivity*, dan *specitivity* mendekati 1 pada setiap kategori. Artinya tingakat akurasi, *sensitivity*, dan *specitivity* LORENS terhadapa IKPS memiliki nilai yang baik.

4.2.4 Evaluasi Performa Model

Model yang terbentuk kemudian dievaluasi dengan menggunakan data *testing*. Dimana fungsi data *testing* digunaan untuk kontrol atau evaluasi terhadap model yang didapatkan. Evaluasi performa model ini dapat dilihat dengan ketepatan klasfikasi. Ketepatan klasifikasi dapat diperoleh melalui hasil dari perhitungan tabulasi silang dimana terdiri dari kelas *True Positive* (TP), *False Positive* (FP), *True Negative* (TN,) dan *False Negative* (FN). Tabel 4.3 berikut merupakan tabel tabulasi silang dengan 10 partisi dan 10 ensembel.

Tabel 4.3 Tabulasi Silang Klasifikasi Aktual dan Klasfikasi Prediksi Evaluasi Model

		Kelas Aktual					
		Rendah		**Sedang**		**Tinggi**	
		p(+)	**n(-)**	**p(+)**	**n(-)**	**p(+)**	**n(-)**
Kelas Prediksi	p(+)	34	2	31	5	70	0
	n(-)	106	0	104	2	67	5

Tabel 4.3 tersebut kemudian dapat dihitung nilai akurasi dan *sensitivity* serta *specitivity* yang dilakukan pada data testing. Nilai tersebut ditunjukkan pada Tabel 4.4 berikut.

Tabel 4.4 Akurasi, *Sensitivity,* dan *Specitivity* Data Testing

IKPS	Akurasi	*Sensitivity*	*Specitivity*
Rendah	0,9859	0,9444	1,0000
Sedang	0,9507	0,8611	0,9811
Tinggi	0,9648	1,0000	0,9306

Tabel 4.4 menunjukkan nilai akurasi, *sensitivity* dan *specitivity* data testing. Hasil tersebut menujukkan bahwa perfoma dari testing sangat baik pada tiap kelas dan setiap partisi serta ensembel. Nilai ketiganya mendekati 1. Sehingga dari hasil tersbeut dapat diketahui juga nilai *error rate* yaitu pada IKPS Rendah 0%, IKPS Sedang adalah 6% dan kelas tinggi yaitu 6,7%. Dapat dilihat pada Tabel 4.5 berikut ini.

Tabel 4.5 Nilai *Error* pada Data Testing

IKPS	***Error rate***
Rendah	0
Sedang	0,0606
Tinggi	0,0667

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang didapat dari peneilitian ini berdasarkan hasil yang diperoleh sebagai berikut

1. Hasil klasifikasi IKPS dengan metode LOREN
   a. Model kumulatif IKPS Rendah (Kategori 1)

   $$\log(\frac{P(Y\leq 1)}{1-P(Y\leq 1)}) = 60{,}049 - (0{,}0272X_1 + 0{,}0747X_2 + 0{,}0461X_3 + 0{,}0103X_4 + 0{,}0103X_5 + 0{,}137X_6 + 0{,}085X_6 + 0{,}134X_7 + 0{,}216X_8 + 0{,}084X_9 + 0{,}137X_{10})$$

   b. Peluang kumulatif IKPS Sedang (Kategori 2)

   $$\log(\frac{P(Y\leq 2)}{1-P(Y\leq 2)}) = 70{,}718 - (0{,}0272X_1 + 0{,}0747X_2 + 0{,}0461X_3 + 0{,}0103X_4 + 0{,}0103X_5 + 0{,}137X_6 + 0{,}085X_6 + 0{,}134X_7 + 0{,}216X_8 + 0{,}084X_9 + 0{,}137X_{10})$$

   c. Peluang kumulatif IKPS Tinggi (Kategori 3)

   $$P(Y = 3) = 1 - P(Y \leq 2) = 1 - (70{,}718 - (0{,}0272X_1 + 0{,}0747X_2 + 0{,}0461X_3 + 0{,}0103X_4 + 0{,}0103X_5 + 0{,}137X_6 + 0{,}085X_6 + 0{,}134X_7 + 0{,}216X_8 + 0{,}084X_9 + 0{,}137X_{10})$$

   Peluang untuk IKPS Kategori Tinggi memiliki peluang 0 yang artinya peluang untuk berada di kategori ini sangat kecil mendekati nol dibanding dengan kategori IKPS yang lain. Koefesien model LOREN menunjukkan bahwa variabel Imunisasi, Penolong Persalinan, KB, ASI Ekslusif, Air Minum Layak dan Jaminan Kesehatan dan Jamkesda memberikan koefesien yang rendah. Artinya variabel-variabel tersebut diakukan fokus perbaikan.
2. Ketepatan hasil klasifikasi yang didapatkan dari model yaitu 98,59%, 95,07% dan 96,48%. Artinya model telah tepat terprediksi dengan baik karena mendekati nilai 100%.

5.2 Saran

Saran dari penelitian berdasarkan analisis yang dilakukan serta kesimpulan, yaitu memeriksa kembali *syntax* terutama pada bagian partisi dan ensembel. Saran untuk pihak terkait yaitu, melakukan pengawasan dan penanganan terhadap variabel yang memiliki koefesien rendah. Karena memerlukan fokus agar dapat mengatasi masalah penanganan stunting. Perlunya edukasi massif terhadap koefesien-koefesien tersebut agar tidak tertinggal di kawasan Asia.

DAFTAR PUSTAKA

Agresti, A. (2013). *Categorical Data Analysis* (3rd ed.). John Wiley & Sons.

Ahn, H., Moon, H., Fazzari, M. J., Lim, N., Chen, J. J., & Kodell, R. L. (2007). *Classification by ensembles from random partitions of high-dimensional data*. *51*, 6166–6179. https://doi.org/10.1016/j.csda.2006.12.043

Badan Penelitian dan Pengembangan Kesehatan. (2018). Laporan Nasional Riskesdas 2018. Kementerian Kesehatan RI.

Fawcett, Tom (2006). "An Introduction to ROC Analysis" (PDF). Pattern Recognition Letters. 27 (8): 861–874. doi:10.1016/j.patrec.2005.10.010. S2CID 2027090

Hosmer, D. W., & Lemeshow, S. (2000). *Applied Logistic Regression second Edition*. Wiley Series in Probability and Statistics.

Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied Logistic Regression Third Edition. In *Technometrics* (Vol. 38, Issue 2). https://doi.org/10.2307/1270433

Kemenkes. (2018). Mengenal Stunting dan Gizi Buruk. Penyebab, Gejala, Dan Mencegah https://promkes.kemkes.go.id/?p=8486

Kemenkes. (2023). https://ayosehat.kemkes.go.id/cegah-stunting-itu-penting

Kuswanto, H., Asfihani, A., Sarumaha, Y., Ohwada, H., (2015). Logistic Regression Ensemble for Predicting Customer Defection with Very Large Sample Size. Procedia Computer Science. 72. 86-93. https://doi.org/10.1016/j.procs.2015.12.108

Lim, N. (2007). *Classification by Ensembles from Random Partitions using Logistic Regression Models*. Stony Brook University.

Lim, N., Ahn, H., Moon, H., & Chen, J. J. (2010). *Classification of High-Dimensial Data With Ensemble of Logistic Regression Model*. 160–171. https://doi.org/10.1080/10543400903280639

Melasasi, J. N. (2015). *Klasifikasi Enzim Pada Database Dud-E Dengan Metode Logistic Regression Ensemble (Lorens) Dengan Metode Logistic*

Regression. Institut Teknologi Sepuluh Nopember.

Morris, C., Yang, J.J. (2021). Effectiveness of resampling methods in coping with imbalanced crash data: Crash type analysis and predictive modeling. https://doi.org/10.1016/j.aap.2021.106240

Studi Statis Gizi Sehat. (2021). https://www.badankebijakan.kemkes.go.id/buku-saku-hasil-studi-status-gizi-indonesia-ssgi-tahun-2021/

WHO. (2015). https://www.who.int/news/item/19-11-2015-stunting-in-a-nutshell#:~:text=Stunting%20is%20the%20impaired%20growth,WHO%20Child%20Growth%20Standards%20median.

Widhianingsih, T. D. A., Kuswanto, H., & Prastyo, D. D. (2020). Logistic Regression Ensemble (LORENS) Applied to Drug Discovery. *Matematika*, *36*(1), 43–49. https://doi.org/10.11113/matematika.v36.n1.1197

UNDP, 2021, Technical notes, https://hdr.undp.org/sites/default/files/2021-22_HDR/hdr2021-22_technical_notes.pdf

Utami, R. A., Setiawan, A., & Fitriyani, P. (2019). Identifying causal risk factors for stunting in children under five years of age in South Jakarta, Indonesia. Enfermeria Clinica, 29, 606-611. https://doi.org/10.1016/j.enfcli.2019.04.093

LAMPIRAN

Lampiran 1. Mengatasi *Imbalance Data* dengan SMOTE

```
install.packages("smotefamily")
library(smotefamily)

smote_datakod <- SMOTE(datakod[,-1], datakod[,1], K = 5)
class_vector <- smote_data$data
table(class_vector$class)

# Mengekstrak vektor kelas yang telah di SMOTE
new_data <- data.frame(class_vector)

# Menyimpan output smote
write.csv(class_vector, "~/Downloads/smote_datakod.csv", row.names = FALSE)

# Before SMOTE
table(datakod$Kelas)

# After SMOTE
table(smote_datakod$class)

# Membaca CSV File dengan pada direktori file
data <- read.csv("~/Downloads/smote_datakod.csv", stringsAsFactors = FALSE,
                 header = TRUE, sep = ",", quote = "\"")

#Setelah menggunakan SMOTE digambarkan dengan Pie Chart
# Menghitung kemunculan tiap kelas (kategori tiap kelas
class_counts <- table(smote_datakod$class)

# Mengekstrasi nilai dan label
slices <- as.numeric(class_counts)
# Membuat label
custom_labels <- c("1 Rendah", "2 Sedang", "3 Tinggi")

# Menghitung persentase
percentages <- round(100 * slices / sum(slices), 1)
# Mengkombinasikan label dengan persen
labels_with_percent <- paste(labels, "(", percentages, "%)", sep="")

####################################
# Membuat Pie ChartCreate the pie chart#
pie(slices, labels = custom_labels, main = "Pie Chart SMOTE Data Kelas IKPS",
    col = rainbow(length(slices)))
```

```
# Memasukkan legenda dengan mengkustom posisi legenda misalkan atas, bawah gambar
legend("bottomleft", legend = labels_with_percent, fill = rainbow(length(slices)))
```

Lampiran 2. *Syntax* LOREN pada kasus Indeks Penanganan Stunting (IKPS)

```
# Required libraries
library(MASS)        # For logistic regression
library(dplyr)       # For data manipulation
library(nnet)

# Data telah memuat tiga kategori
# Memastikan respn variabel adalah sebuah faktor
smote_datakod$class <- as.factor(smote_datakod$class)

# Set a seed for reproducibility
set.seed(2024)

# Split data into training and testing sets (85% training, 15% testing)
train_index <- sample(seq_len(nrow(smote_datakod)), size = 0.85 *
nrow(smote_datakod))
train_data <- smote_datakod[train_index, ]
test_data <- smote_datakod[-train_index, ]

# Parameternya
n <- 10  # Number of partitions
m <- 10   # Number of ensemble models to be created


# Function to train a logistic regression model
# Function to train a multinomial logistic regression model
train_logistic_model <- function(data) {
  model <- multinom(as.factor(class) ~ ., data = data, maxit = 1000)
  return(model)
}
#maxit = 1000 adalah banyaknya iterasi boleh diubah sesuai dengan peneliti

# Create partitions for the training data
set.seed(123)
partition_size <- floor(nrow(train_data) / n)
partitions <- list()

for (i in 1:n) {
  partitions[[i]] <- train_data[sample(nrow(train_data), partition_size, replace = TRUE), ]
}

# Train logistic regression models on each partition
models <- list()
for (i in 1:m) {
  models[[i]] <- train_logistic_model(partitions[[i]])
```

```
}
# Example: Extracting the logit model (coefficients) from the first model
logit_model <- models[[1]]
logit_model2 <- models[[2]]
logit_model5 <- models[[5]]
logit_model7 <- models[[7]]

# Function to make predictions with all ensemble models
ensemble_predict <- function(models, new_data) {
# Create an empty matrix to store predictions
  preds <- matrix(0, nrow = nrow(new_data), ncol = length(models))

  for (i in 1:length(models)) {
    preds[, i] <- predict(models[[i]], new_data, type = "class")
  }

  # Aggregate predictions (e.g., majority voting)
  final_preds <- apply(preds, 1, function(x) names(which.max(table(x))))

  return(final_preds)
}

# Make predictions on the test data using the ensemble
ensemble_predictions <- ensemble_predict(models, test_data)

# Evaluate the performance (Confusion matrix or accuracy)
table(Predicted = ensemble_predictions, Actual = test_data$class)

# Optionally, calculate accuracy
accuracy <- sum(ensemble_predictions == test_data$class) / nrow(test_data)
print(paste("Ensemble Model Accuracy: ", round(accuracy * 100, 2), "%", sep = ""))




##############################################################################
# The Best Logit Model#
##############################################################################

# Function to evaluate the performance of each model
evaluate_model <- function(model, test_data) {
  predictions <- predict(model, newdata = test_data, type = "class")

  # Calculate confusion matrix
  conf_matrix <- table(Predicted = predictions, Actual = test_data$class)
```

```
  # Calculate accuracy
  accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)

  # Calculate Kappa
  total <- sum(conf_matrix)
  p_observed <- sum(diag(conf_matrix)) / total
  p_expected <- sum(rowSums(conf_matrix) * colSums(conf_matrix)) / (total^2)
  kappa <- (p_observed - p_expected) / (1 - p_expected)

  return(list(accuracy = accuracy, kappa = kappa))
}

# Evaluate each model and store performance metrics
model_performances <- list()

for (i in 1:m) {
  metrics <- evaluate_model(models[[i]], test_data)
  model_performances[[i]] <- metrics
  print(paste("Model", i, "- Accuracy:", round(metrics$accuracy * 100, 2), "%", "Kappa:",
round(metrics$kappa, 2)))
}

# Find the best model based on accuracy
best_model_idx <- which.max(sapply(model_performances, function(x) x$accuracy))
best_model <- models[[best_model_idx]]
best_metrics <- model_performances[[best_model_idx]]

# Output the best model and its metrics
print(paste("Best Model Index: ", best_model_idx))
print(paste("Best Model Accuracy: ", round(best_metrics$accuracy * 100, 2), "%"))
print(paste("Best Model Kappa: ", round(best_metrics$kappa, 2)))

# Extracting the logit model (coefficients) from the BEST model
logit_model7 <- models[[7]]
logit_model7

# Extract coefficients for the category "tinggi" (assumed to be the third category)
coef_rendah <- coef(logit_model7)[1, ]
print(coef_rendah)
```

Lampiran 3. *Output* LOREN

```
polr(formula = class ~ ., data = train_data, Hess = TRUE)

Coefficients:
       Imunisasi penolong_persalinan              KB         ASI_eks           MPASI
      0.02717611        0.07467723       0.04610216       0.01034366      0.13725621
             AML          sanitasi           PAUD      jkn_jamkesda        KPS_KKS
      0.08532765        0.13392047       0.21632852       0.08383344      0.13724071

Intercepts:
    1|2     2|3
60.04872 70.71826

Residual Deviance: 331.0217
AIC: 355.0217

print(paste("Accuracy:", round(accuracy * 100, 2), "%"))
[1] "Accuracy: 95.07 %

confusion_matrix <- table(Predicted = result$decision, Actual = y_train)
print(confusion_matrix)
predictions  1  2  3
          1  0  0  0
          2  2 20  9
          3  1  9 25

result <- lr.cerp(y = y, x = x, nens = 10, search = FALSE)
[1] "Structure of y:"
 Ord.factor w/ 3 levels "1"<"2"<"3": 2 1 1 2 3 1 2 3 3 2 ...
NULL
[1] "Structure of x:"
'data.frame':    804 obs. of  10 variables:
 $ Imunisasi         : num  71.6 65.3 65.1 80.8 89.2 ...
 $ penolong_persalinan: num  95 87.5 68.6 98.4 96.9 ...
 $ KB                : num  96.5 59.3 79.9 59.9 69.2 ...
 $ ASI_eks           : num  100 89.5 100 89.9 99.8 ...
 $ MPASI             : num  85.8 86.1 92.4 63 91.5 ...
 $ AML               : num  85.8 91.4 51.3 95.7 95 ...
 $ sanitasi          : num  89.4 83 75.7 96.4 92.8 ...
 $ PAUD              : num  23.6 28.4 31.8 52.4 58.3 ...
 $ jkn_jamkesda      : num  96.5 77.9 42.8 82.2 94.9 ...
 $ KPS_KKS           : num  35.1 25.6 19.3 16.6 48.5 ...
NULL
[1] "Ensemble size: 2"
[1] "Threshold: 0.36318407960199"
```

Lampiran 4. Mencari Model Ensembel

```
# Definisikan jumlah ensemble model (nens) terlebih dahulu
nens <- 10  # Misalnya, ingin membuat 10 model ensemble

# Pastikan variabel lainnya juga telah didefinisikan
optsize <- 5  # Ukuran partisi optimal (misalnya)
num_pred <- ncol(x)  # Jumlah prediktor (misalnya)
num_obs <- nrow(x)  # Jumlah observasi (misalnya)
opthreshold <- 0.3644279  # Threshold untuk voting (misalnya)

# Barulah kode berikutnya dapat dieksekusi tanpa error
ptss <- floor(seq(1,optsize+.999,length.out=num_pred))
fitted <- NULL
predicted <- NULL
cname <- NULL
coef.table <- matrix(0, num_pred, nens)
partition.table <- matrix(0, num_pred, nens)
intc <- matrix(0, optsize, nens)
probability <- rep(0, num_obs)
for (i in 1:nens) {
  cname <- c(cname, paste("ens", i, sep = ""))
  rand_pred <- sample(ptss)
  partition.table[, i] <- rand_pred
  avg_fit <- rep(0, num_obs)
  for (j in 1:optsize) {
    # Create data subset
    smp_dt <- as.data.frame(cbind(class = y, x[, rand_pred == j]))
    print(head(smp_dt))  # Check subset of data

    # Fit the model
    intlr <- multinom(class ~ ., data = smp_dt)
    print(intlr)  # Check model object

    # Aggregate predictions
    avg_fit <- avg_fit + predict(intlr, type = "prob")[, 2]  # Adjust as needed

    # Debugging statement
    print(paste("i =", i, "j =", j, "avg_fit =", avg_fit[1:5]))  # Print first 5 values of
avg_fit
  }

  # Combine with fitted
  fitted <- cbind(fitted, avg_fit / optsize)  # Simplify cbind
  probability <- probability + (avg_fit / optsize) / nens
}
```

```
learning.decision <- ens.voting(fitted, opthreshold)$final.vote
colnames(fitted) <- cname
colnames(intc) <- cname
colnames(coef.table) <- cname
rownames(coef.table) <- colnames(x)
colnames(partition.table) <- cname
rownames(partition.table) <- colnames(x)
return(list(fitted = fitted, probability = probability, learning.decision = learning.decision,
        partition.table = partition.table, coef.table = coef.table, intercept = intc,
        number.ensemble = nens, optimal.size = optsize, optimal.threshold = opthreshold))

y <- as.factor(y)
print(levels(y))  # Should print three levels

# Fit the model
intlr <- multinom(class ~ ., data = smp_dt)
print(intlr)  # Check model object

# Aggregate predictions
avg_fit <- avg_fit + predict(intlr, type = "prob")[, 2]  # Adjust as needed

# Debugging statement
   print(paste("i =", i, "j =", j, "avg_fit =", avg_fit[1:5]))  # Print first 5 values of avg_fit
  }

# Combine with fitted
fitted <- cbind(fitted, avg_fit / optsize)  # Simplify cbind
print(paste("i =", i, "fitted dimensions:", dim(fitted)))  # Print dimensions of fitted
}

print(avg_fit)  # Should show numeric values
print(length(avg_fit))  # Should match the number of observations (num_obs)
print(paste("Loop i =", i, "j =", j))  # Check loop iterations
print(length(avg_fit))  # Should be equal to num_obs
print(num_obs)  # Should print the number of observations
print(optsize)  # Should be a single numeric value
fitted <- cbind(fitted, avg_fit / optsize)  # Remove deparse.level for debugging
print(dim(fitted))  # Check dimensions of fitted after cbind


learning.decision <- ens.voting(fitted, opthreshold)$final.vote
ens.voting <- function(fitted, threshold) {
```

```
  vote_count <- rowSums(fitted > threshold)
  final_vote <- ifelse(vote_count > (ncol(fitted) / 2), 1, 0)
  return(list(final.vote = final_vote))
}

# Pastikan fitted sudah berisi nilai sebelum dipanggil
if (is.null(fitted)) {
  stop("Fitted values not generated. Please check the loop that generates fitted
values.")
}

# Pastikan opthreshold didefinisikan dan merupakan angka
if (is.null(opthreshold) || !is.numeric(opthreshold)) {
  stop("Optimal threshold (opthreshold) must be defined as a numeric value.")
}

print("Generating learning.decision...")
print(dim(fitted))  # Harus memiliki dimensi yang benar (jumlah observasi x
jumlah ensemble)
print(opthreshold)  # Pastikan threshold ada

# Pastikan fitted sudah benar
  if (is.null(fitted)) {
    stop("Error: 'fitted' is null.")
  }

  # Definisikan opthreshold jika belum ada
  if (!exists("opthreshold")) {
    opthreshold <- 0.5  # Default threshold
  }

  # Generate learning.decision
  learning.decision <- ens.voting(fitted, opthreshold)$final.vote
  print("learning.decision generated successfully.")

  # Tambahkan nama kolom dan baris
  colnames(fitted) <- paste("ens", 1:nens, sep = "")
  colnames(intc) <- paste("ens", 1:nens, sep = "")
  colnames(coef.table) <- paste("ens", 1:nens, sep = "")
  rownames(coef.table) <- colnames(x)
  colnames(partition.table) <- paste("ens", 1:nens, sep = "")
  rownames(partition.table) <- colnames(x)


ensemble_logistic <- function() {
  # Kode fungsi yang sudah Anda definisikan sebelumnya
```

```
  # ...
  return(list(
    fitted = fitted,
    probability = probability,
    learning.decision = learning.decision,
    partition.table = partition.table,
    coef.table = coef.table,
    intercept = intc,
    number.ensemble = nens,
    optimal.size = optsize,
    optimal.threshold = opthreshold
  ))
}
hasil <- ensemble_logistic()
# Tampilkan matriks fitted values
print(hasil$fitted)
```

Lampiran 5. Hasil Ensemble

```
hasil <- ensemble_logistic()
> # Tampilkan matriks fitted values
> print(hasil$fitted)
        ens1      ens2      ens3      ens4      ens5      ens6      ens7      ens8      ens9
ens10
578 0.38837191 0.33903866 0.35950933 0.42564397 0.44164729 0.37299698
0.30658121 0.40479217 0.39605822 0.36088599
549 0.41470952 0.40846204 0.35749303 0.42263550 0.43073025 0.40546083
0.37172507 0.42147994 0.36933941 0.40170675
557 0.28100142 0.33133039 0.23543991 0.26022626 0.22511000 0.28103516
0.29048352 0.30795973 0.25655061 0.26276605
700 0.26805553 0.33425613 0.26455698 0.27172834 0.32805301 0.31263400
0.35739332 0.33372878 0.33672212 0.27363290
255 0.25971153 0.26241284 0.26614533 0.26980890 0.25994141 0.26218058
0.26674986 0.25370627 0.27648703 0.27448734
913 0.16026348 0.19890205 0.17797184 0.17907752 0.15970856 0.11318133
0.20902169 0.18645731 0.18532082 0.19329488
621 0.35523009 0.39602467 0.36888387 0.35350039 0.38994922 0.34350121
0.32534426 0.33319199 0.37626188 0.29751379
416 0.16754816 0.15230571 0.18194851 0.15675909 0.14895885 0.16700980
0.17839646 0.15345912 0.16521008 0.17308108
105 0.30859666 0.29286796 0.31638177 0.31373139 0.31233117 0.29955520
0.28428847 0.28601106 0.29944533 0.31191164
634 0.33086472 0.37198450 0.38111828 0.33900155 0.42406942 0.33667419
0.43794404 0.38035873 0.36690253 0.33853029
738 0.31221312 0.41972336 0.35296032 0.36958242 0.44799608 0.31204667
0.45319258 0.40020284 0.38301695 0.38170772
29  0.15676089 0.13135068 0.16230764 0.15457081 0.15652735 0.15060292
0.17933457 0.14462437 0.15501061 0.15797181
11  0.18596997 0.17395399 0.20510190 0.19463167 0.19355396 0.19174484
0.15363472 0.17276495 0.17065978 0.20032046
784 0.40145984 0.38837282 0.44026002 0.36046466 0.37304368 0.45834205
0.47632731 0.45907345 0.35936103 0.42672382
925 0.08927889 0.09304064 0.03568034 0.05852490 0.05526880 0.06773281
0.09205282 0.04871573 0.04044504 0.11107023
62  0.15938601 0.13628191 0.16173469 0.15678946 0.15499846 0.14013721
0.16963364 0.14462458 0.14616483 0.15771091
252 0.20167437 0.21448474 0.24215682 0.23227080 0.25581292 0.20026624
0.18709799 0.19964087 0.20929653 0.21082652
398 0.26163946 0.25776869 0.26179237 0.26089440 0.25546027 0.25939348
0.27745316 0.26213314 0.25141474 0.25429986
26  0.25905048 0.29331437 0.25965399 0.26580341 0.24748440 0.26001981
0.28683930 0.28204327 0.26559222 0.26792904
```

172 0.19312393 0.17598361 0.20967451 0.19327425 0.17929005 0.18255541 0.16215936 0.18246834 0.16628730 0.19152236
562 0.32568977 0.35992609 0.31063676 0.33175724 0.31304488 0.30994956 0.33617000 0.33488282 0.32629909 0.32630993
410 0.17830458 0.15629001 0.18354537 0.15946259 0.15436550 0.19028732 0.19807745 0.15766207 0.17011460 0.18397227
32 0.17695150 0.15432145 0.18248432 0.15837300 0.15342398 0.18904725 0.19668761 0.15646197 0.16926768 0.18281983
725 0.36554434 0.38225742 0.36884490 0.35786702 0.31810270 0.37859453 0.30697813 0.31795551 0.39400519 0.34776674
385 0.24869900 0.24292111 0.25529820 0.25605745 0.24477531 0.25101022 0.23871607 0.27185963 0.25410683 0.25844060
203 0.25135363 0.23143258 0.27689031 0.24853024 0.24201199 0.27706741 0.25858896 0.29022869 0.24593939 0.25399116
35 0.22747012 0.21896458 0.22083233 0.21059427 0.19846682 0.22532607 0.21155730 0.20856746 0.18450487 0.21151986
361 0.22026600 0.23089370 0.23256785 0.22131447 0.23123922 0.25030128 0.24110454 0.23945788 0.23145803 0.23427605
238 0.16159501 0.14419591 0.17508427 0.17086292 0.16202762 0.15033165 0.17996842 0.15464683 0.16247568 0.16996861
593 0.43015556 0.43197849 0.38254872 0.40940090 0.36362379 0.41907027 0.38541682 0.35965879 0.39134068 0.32899181
284 0.22245094 0.19329140 0.21676296 0.23123307 0.22555516 0.21299776 0.19474007 0.21496947 0.21383654 0.23028772
304 0.17335259 0.16953004 0.19719020 0.19838228 0.19840443 0.15895411 0.18173457 0.17013757 0.17931775 0.19171178
216 0.19734418 0.21006149 0.23722539 0.22710058 0.24833358 0.19584780 0.18141525 0.19609534 0.20189597 0.20525016
596 0.42477477 0.43484325 0.32683872 0.37578497 0.36343646 0.44622535 0.40360257 0.41858634 0.40896606 0.42093554
476 0.29950892 0.34452254 0.31365081 0.32686583 0.32225805 0.34675958 0.35136992 0.27758629 0.28437092 0.26686851
852 0.41890108 0.36424420 0.35145985 0.35076688 0.32824896 0.35021825 0.43009464 0.34878602 0.41169009 0.42937177
427 0.23669609 0.21109596 0.23269668 0.24718535 0.24338491 0.22550459 0.21118449 0.22431871 0.22789226 0.24507110
442 0.32427905 0.28586034 0.30400878 0.31311209 0.30728308 0.30721675 0.28888389 0.30883445 0.26955984 0.28706605
884 0.37926931 0.33696081 0.32675238 0.38300316 0.37607950 0.36104052 0.36349862 0.36166021 0.36071942 0.34634321
276 0.18575758 0.15612205 0.20253155 0.19218157 0.22510006 0.20326986 0.17841905 0.19950176 0.20474338 0.18921945
87 0.25761898 0.31587863 0.25215401 0.25910695 0.22244271 0.25939692 0.27181823 0.26442003 0.25916741 0.23432342
505 0.11092738 0.10232363 0.12970346 0.13246375 0.06765096 0.06893747 0.09016564 0.11840219 0.08707213 0.12431877

943 0.01947465 0.09377843 0.03081294 0.01982116 0.01880962 0.08760952 0.07462028 0.04015480 0.02705900 0.03615906
618 0.29616726 0.38057701 0.32315073 0.37200314 0.34475338 0.31360169 0.35001096 0.37258591 0.35430915 0.34706577
892 0.27525766 0.31237449 0.24604286 0.28578174 0.26992171 0.28824408 0.30303047 0.33289216 0.28852276 0.29740429
900 0.34902663 0.39171877 0.33087980 0.34532018 0.35235976 0.28637245 0.34150161 0.35347259 0.29783894 0.39402646
647 0.28736069 0.41498928 0.37118093 0.37820293 0.32345237 0.28710368 0.37506572 0.40118310 0.37934307 0.39860512
441 0.29102696 0.30253674 0.31317833 0.29529211 0.28169674 0.29626759 0.25960264 0.28604929 0.31252525 0.30711789
336 0.24147541 0.23270414 0.24044287 0.22461697 0.21274583 0.23411499 0.25039165 0.24000271 0.23302126 0.24278612
212 0.18419165 0.14822380 0.19973832 0.19002075 0.22566168 0.20419260 0.20498311 0.20016242 0.21194539 0.18215904
835 0.29085266 0.28495012 0.33144096 0.27435173 0.26858468 0.32500691 0.38010546 0.26925671 0.29216196 0.22745917
281 0.30682210 0.29146449 0.30815953 0.29993646 0.29249703 0.31141565 0.27047993 0.29560570 0.30153284 0.31642508
290 0.20002402 0.17889285 0.20490181 0.21355532 0.20840596 0.18679508 0.18270074 0.20292151 0.19253786 0.21216114
217 0.19207945 0.19070829 0.17761473 0.17882710 0.16615545 0.18683022 0.19869212 0.17059364 0.16160868 0.19797237
825 0.43355364 0.46879132 0.45090489 0.44770745 0.48968604 0.44009003 0.46129982 0.41809636 0.48755509 0.43514249
817 0.37809574 0.29754721 0.36989859 0.37353619 0.35592153 0.38460695 0.36654048 0.35502622 0.32979823 0.34738439
310 0.20646905 0.21914562 0.24505450 0.23500224 0.25747187 0.20678199 0.19117526 0.20516783 0.21397012 0.21462900
858 0.40005951 0.37420569 0.34555777 0.40207179 0.41395305 0.38426717 0.37479082 0.37688704 0.38742702 0.38856387
643 0.30299018 0.39124356 0.28793934 0.28733831 0.36110257 0.36815151 0.45967475 0.45385420 0.42536535 0.30020997
153 0.19680719 0.20642492 0.23107094 0.22548477 0.24131939 0.19235235 0.18397705 0.19252691 0.20439269 0.20420884
705 0.34785343 0.29825162 0.35784005 0.34513246 0.38627244 0.42772927 0.34937127 0.36662781 0.38092989 0.36393656
6 0.36785205 0.37346239 0.37951497 0.33975740 0.33298323 0.36973614 0.35400581 0.34251883 0.39230325 0.37517569
788 0.43767583 0.39727146 0.40780690 0.39699761 0.43354027 0.42401189 0.42240187 0.43524594 0.39548143 0.44374350
393 0.22572688 0.21201876 0.23016078 0.23225274 0.22226968 0.21645018 0.21004314 0.23679556 0.22365601 0.23422180
719 0.41850383 0.40478505 0.40562081 0.37199383 0.39350538 0.42460981 0.43754114 0.44785108 0.41849952 0.38956717

717 0.32727278 0.28898234 0.31862282 0.29918224 0.36078160 0.35727329 0.28297992 0.30922138 0.32942547 0.29291396
464 0.21039502 0.18624711 0.21040492 0.22297972 0.21809804 0.19935960 0.18875597 0.20334300 0.20336725 0.22041443
910 0.17074182 0.12303104 0.18053621 0.18368733 0.14676422 0.15217290 0.15149483 0.13883774 0.13317666 0.15330946
354 0.33597916 0.27839402 0.26773424 0.28735828 0.38003048 0.31863334 0.30214220 0.30710569 0.27590887 0.27057299
186 0.27859561 0.27311986 0.25096036 0.28523855 0.26024413 0.29954822 0.28415528 0.26360889 0.30134870 0.26322691
305 0.24194735 0.25294423 0.26463075 0.25728949 0.27792970 0.25862576 0.24752185 0.24873443 0.25955205 0.25484854
627 0.34256897 0.28869457 0.36378625 0.33956229 0.35030355 0.31704241 0.25502387 0.31689973 0.36276304 0.26526418
108 0.32033147 0.31496459 0.32781380 0.32449043 0.34538376 0.31912736 0.32348350 0.29527911 0.32792933 0.32859593
261 0.21484938 0.20847023 0.21624158 0.22558077 0.21428602 0.20382440 0.20710294 0.20270851 0.21060117 0.22854028
720 0.31833911 0.29844489 0.30421253 0.32155360 0.33437740 0.30106036 0.31409682 0.33918946 0.30104035 0.31967586
131 0.22344970 0.22144194 0.22739092 0.21181981 0.19956835 0.22492963 0.23355393 0.22195711 0.21614091 0.22416529
887 0.39704589 0.36414742 0.39096029 0.37246978 0.41055586 0.40240212 0.40605365 0.40637905 0.37229406 0.35771636
459 0.27809312 0.25685394 0.27697605 0.28598345 0.28452385 0.26598594 0.25267787 0.26042293 0.27113346 0.28335833
723 0.35639997 0.32271512 0.39853512 0.38133882 0.38022935 0.38072136 0.32092955 0.41378922 0.42804444 0.32346290
414 0.20781130 0.19649287 0.20719945 0.18538120 0.17704419 0.21208327 0.22627776 0.18861935 0.19196383 0.21003368
329 0.21505389 0.21462439 0.22159953 0.20589433 0.19339325 0.21991605 0.22508081 0.21350816 0.20820913 0.21504605
189 0.25057714 0.24322158 0.22389988 0.26256993 0.24711449 0.25113090 0.24531735 0.23768824 0.26951624 0.23629734
259 0.27108952 0.28976447 0.27895720 0.28023512 0.25656001 0.26552527 0.28096537 0.26433662 0.29181053 0.29023649

578 0.44087035 0.44087035
549 0.47328674 0.47328674
557 0.29930562 0.29930562
700 0.31685113 0.31685113
255 0.32429507 0.32429507
913 0.27152143 0.27152143
621 0.35375864 0.35375864
416 0.22952428 0.22952428
105 0.38916994 0.38916994

634 0.40947484 0.40947484
738 0.45170970 0.45170970
29 0.19673323 0.19673323
11 0.23490049 0.23490049
784 0.52163531 0.52163531
925 0.11471009 0.11471009
62 0.19184956 0.19184956
252 0.26842656 0.26842656
398 0.32855368 0.32855368
26 0.30642721 0.30642721
172 0.22750616 0.22750616
562 0.40016452 0.40016452
410 0.25003762 0.25003762
32 0.24868194 0.24868194
725 0.38817306 0.38817306
385 0.32778781 0.32778781
203 0.33953224 0.33953224
35 0.24821664 0.24821664
361 0.26893162 0.26893162
238 0.21410857 0.21410857
593 0.37821290 0.37821290
284 0.28757357 0.28757357
304 0.24248737 0.24248737
216 0.25975937 0.25975937
596 0.48193606 0.48193606
476 0.34045816 0.34045816
852 0.52031723 0.52031723
427 0.30548506 0.30548506
442 0.35869655 0.35869655
884 0.41388643 0.41388643
276 0.22506319 0.22506319
87 0.27927624 0.27927624
505 0.13035294 0.13035294
943 0.04772809 0.04772809
618 0.41677439 0.41677439
892 0.36637841 0.36637841
900 0.46987117 0.46987117
647 0.46225305 0.46225305
441 0.38470193 0.38470193
336 0.31341010 0.31341010
212 0.21843874 0.21843874
835 0.28032370 0.28032370
281 0.39506349 0.39506349
290 0.25974641 0.25974641
217 0.23925865 0.23925865
825 0.51461970 0.51461970

```
817 0.38351881 0.38351881
310 0.27258625 0.27258625
858 0.46832748 0.46832748
643 0.35363870 0.35363870
153 0.25775393 0.25775393
705 0.40998720 0.40998720
6   0.46221731 0.46221731
788 0.53330931 0.53330931
393 0.29459003 0.29459003
719 0.46413162 0.46413162
717 0.32105354 0.32105354
464 0.27400442 0.27400442
910 0.15922936 0.15922936
354 0.34048409 0.34048409
186 0.29800055 0.29800055
305 0.31718861 0.31718861
627 0.31684892 0.31684892
108 0.40223401 0.40223401
261 0.27374646 0.27374646
720 0.37243911 0.37243911
131 0.28532833 0.28532833
887 0.43164194 0.43164194
459 0.35370662 0.35370662
723 0.37576184 0.37576184
414 0.27967260 0.27967260
329 0.27140981 0.27140981
189 0.27902178 0.27902178
259 0.33102483 0.33102483
 [ reached getOption("max.print") -- omitted 721 rows ]
```

Lampiran 6. Akurasi Model Data Training

```
# Make predictions on the test set using the ensemble of models
# Each model makes predictions, and we will combine them
pred_matrix <- sapply(models, function(model) {
  predict(model, train_data, type = "class")
})

# Combine predictions by majority voting
ensemble_predictions <- apply(pred_matrix, 1, function(x) {
  as.character(names(sort(table(x), decreasing = TRUE)[1]))
})
# Evaluate the accuracy of the ensemble model
accuracy <- mean(ensemble_predictions == train_data$class)
print(paste("Ensemble Accuracy:", accuracy))

# Confusion matrix for performance evaluation
table(Predicted = ensemble_predictions, Actual = train_data$class)

# Menghitung prediksi ensemble pada data training
pred_matrix <- sapply(models, function(model) {
  predict(model, train_data, type = "class")
})

# Menggabungkan prediksi menggunakan majority voting
ensemble_predictions <- apply(pred_matrix, 1, function(x) {
  as.character(names(sort(table(x), decreasing = TRUE)[1]))
})

# Menghitung akurasi ensemble pada data training
train_accuracy <- mean(ensemble_predictions == train_data$class)
print(paste("Ensemble Accuracy on Training Data:", train_accuracy))

# Membuat confusion matrix untuk evaluasi performa
conf_matrix <- table(Predicted = ensemble_predictions, Actual = train_data$class)
print(conf_matrix)

# Menghitung TP, FP, TN, dan FN untuk setiap kelas
classes <- levels(train_data$class)

# Inisialisasi variabel untuk menyimpan hasil
results <- data.frame(Class = classes, TP = 0, FP = 0, TN = 0, FN = 0)

# Menghitung TP, FP, TN, dan FN untuk masing-masing kelas
for (i in 1:length(classes)) {
  class <- classes[i]

  # True Positive
  results$TP[i] <- conf_matrix[class, class]
```

```
  # False Positive: Sum of the row for the class minus True Positive
  results$FP[i] <- sum(conf_matrix[class, ]) - results$TP[i]

  # False Negative: Sum of the column for the class minus True Positive
  results$FN[i] <- sum(conf_matrix[, class]) - results$TP[i]

  # True Negative: Sum of all elements in the matrix minus sum of row and column for
the class + True Positive
  results$TN[i] <- sum(conf_matrix) - (results$TP[i] + results$FP[i] + results$FN[i])
}

# Menampilkan hasil perhitungan TP, FP, TN, dan FN
print(results)
```

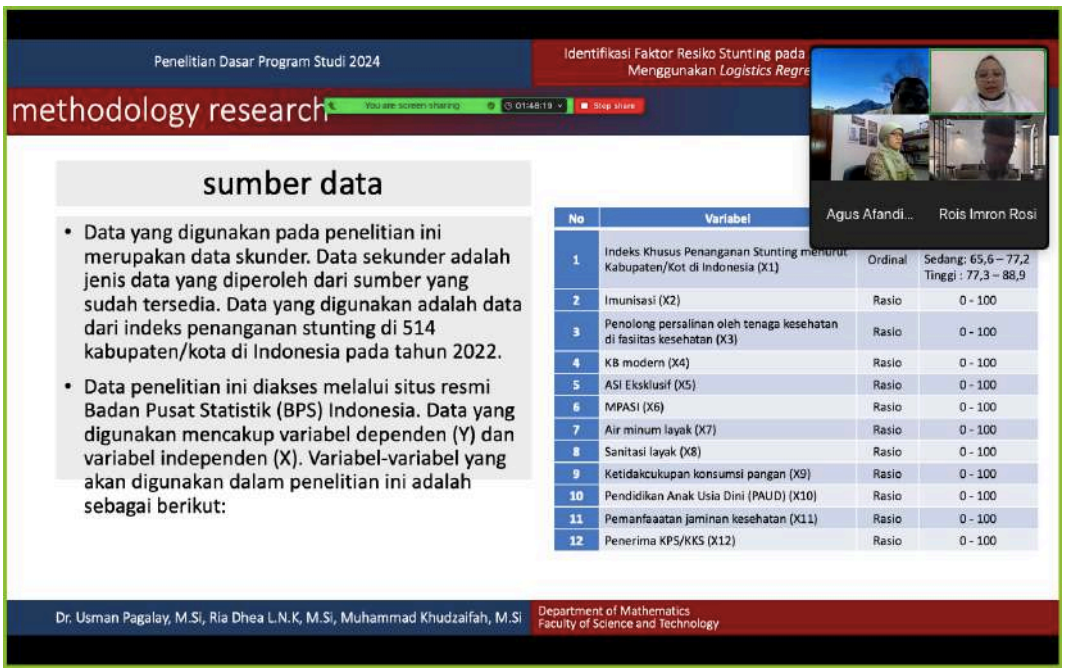
Lampiran 7. Evaluasi Model Klasifikasi

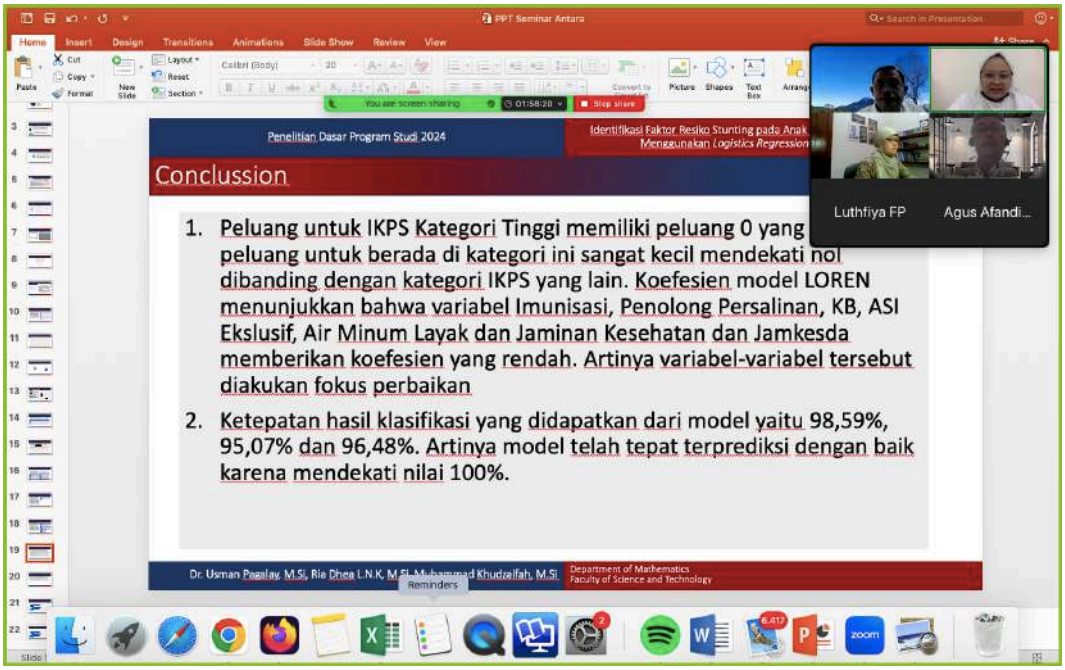
```
# Calculate sensitivity, specificity, accuracy for each class
calculate_metrics <- function(predictions, actual, class_label) {
  binary_actual <- ifelse(actual == class_label, 1, 0)
  binary_predictions <- ifelse(predictions == class_label, 1, 0)

  conf_matrix <- table(Predicted = binary_predictions, Actual = binary_actual)

  TP <- ifelse("1" %in% rownames(conf_matrix) & "1" %in% colnames(conf_matrix),
conf_matrix["1", "1"], 0)
  TN <- ifelse("0" %in% rownames(conf_matrix) & "0" %in% colnames(conf_matrix),
conf_matrix["0", "0"], 0)
  FP <- ifelse("1" %in% rownames(conf_matrix) & "0" %in% colnames(conf_matrix),
conf_matrix["1", "0"], 0)
  FN <- ifelse("0" %in% rownames(conf_matrix) & "1" %in% colnames(conf_matrix),
conf_matrix["0", "1"], 0)

  sensitivity <- TP / (TP + FN)
  specificity <- TN / (TN + FP)
  accuracy <- (TP + TN) / (TP + TN + FP + FN)

  return(list(sensitivity = sensitivity, specificity = specificity, accuracy = accuracy))
}

# Get predicted probabilities from the ensemble
probs <- ensemble_predict_probs(models, test_data)

# Make final predictions using the threshold (highest probability for each instance)
# Function to select the class with the highest probability (no threshold required)
apply_threshold <- function(probs) {
  class_labels <- colnames(probs)

  # Get the class with the highest probability for each instance
  pred_class <- apply(probs, 1, function(row) {
    class_labels[which.max(row)]
  })

  return(pred_class)
}


# Function to calculate sensitivity, specificity, accuracy for each class
calculate_metrics <- function(predictions, actual, class_label) {
  # Ensure predictions and actual are the same length
  if (length(predictions) != length(actual)) {
    stop("Error: predictions and actual values must have the same length.")
  }

  # Binary conversion: Treat class_label as positive (1), and all others as negative (0)
  binary_actual <- ifelse(actual == class_label, 1, 0)
```

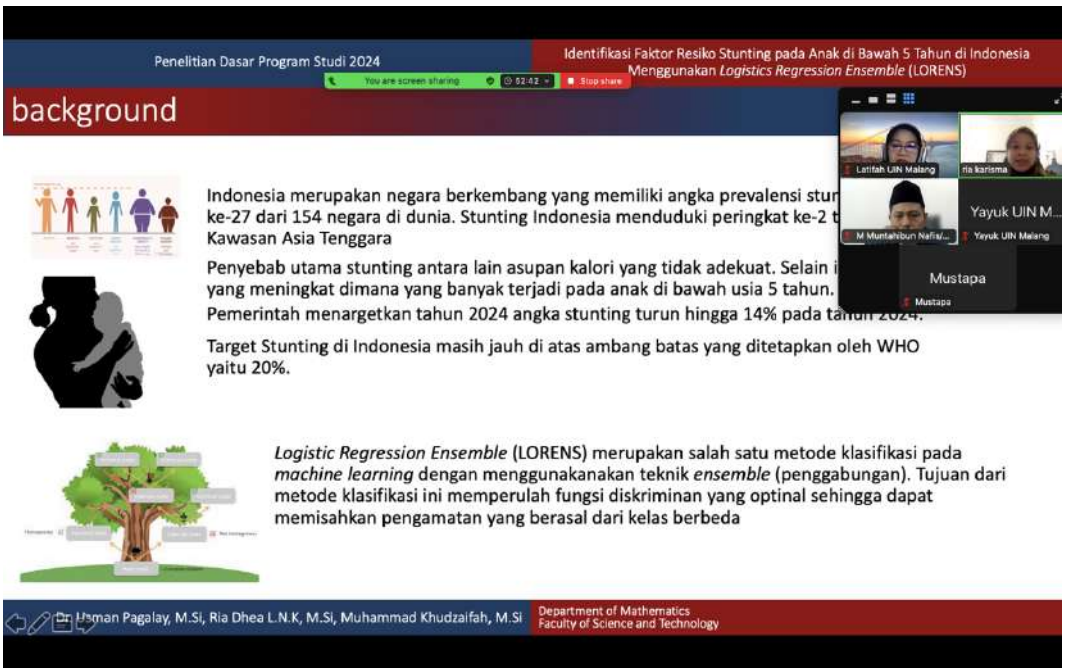
```
  binary_predictions <- ifelse(predictions == class_label, 1, 0)

  # Ensure that binary predictions and actual have the same length
  if (length(binary_predictions) != length(binary_actual)) {
    stop("Error: binary predictions and actual values must have the same length.")
  }

  # Create confusion matrix
  conf_matrix <- table(Predicted = binary_predictions, Actual = binary_actual)

  # Extract TP, TN, FP, FN from the confusion matrix
  TP <- ifelse("1" %in% rownames(conf_matrix) & "1" %in% colnames(conf_matrix),
conf_matrix["1", "1"], 0)
  TN <- ifelse("0" %in% rownames(conf_matrix) & "0" %in% colnames(conf_matrix),
conf_matrix["0", "0"], 0)
  FP <- ifelse("1" %in% rownames(conf_matrix) & "0" %in% colnames(conf_matrix),
conf_matrix["1", "0"], 0)
  FN <- ifelse("0" %in% rownames(conf_matrix) & "1" %in% colnames(conf_matrix),
conf_matrix["0", "1"], 0)

  # Calculate metrics
  sensitivity <- TP / (TP + FN)
  specificity <- TN / (TN + FP)
  accuracy <- (TP + TN) / (TP + TN + FP + FN)

  return(list(sensitivity = sensitivity, specificity = specificity, accuracy = accuracy))
}

# Example usage: predictions and actual labels
predictions <- c("rendah", "sedang", "tinggi", "rendah", "rendah", "sedang", "tinggi",
"rendah")
actual <- c("rendah", "sedang", "sedang", "rendah", "rendah", "tinggi", "tinggi",
"sedang")

# Ensure both predictions and actual values are of the same length
if (length(predictions) != length(actual)) {
  stop("Error: Length of predictions and actual values must match.")
}

# Calculate metrics for each class
class_labels <- unique(actual)
results <- list()

for (class_label in class_labels) {
  metrics <- calculate_metrics(predictions, actual, class_label)
  results[[class_label]] <- metrics
  print(paste("Class", class_label, "- Sensitivity:", round(metrics$sensitivity, 2),
              "Specificity:", round(metrics$specificity, 2),
              "Accuracy:", round(metrics$accuracy * 100, 2), "%"))
}
```

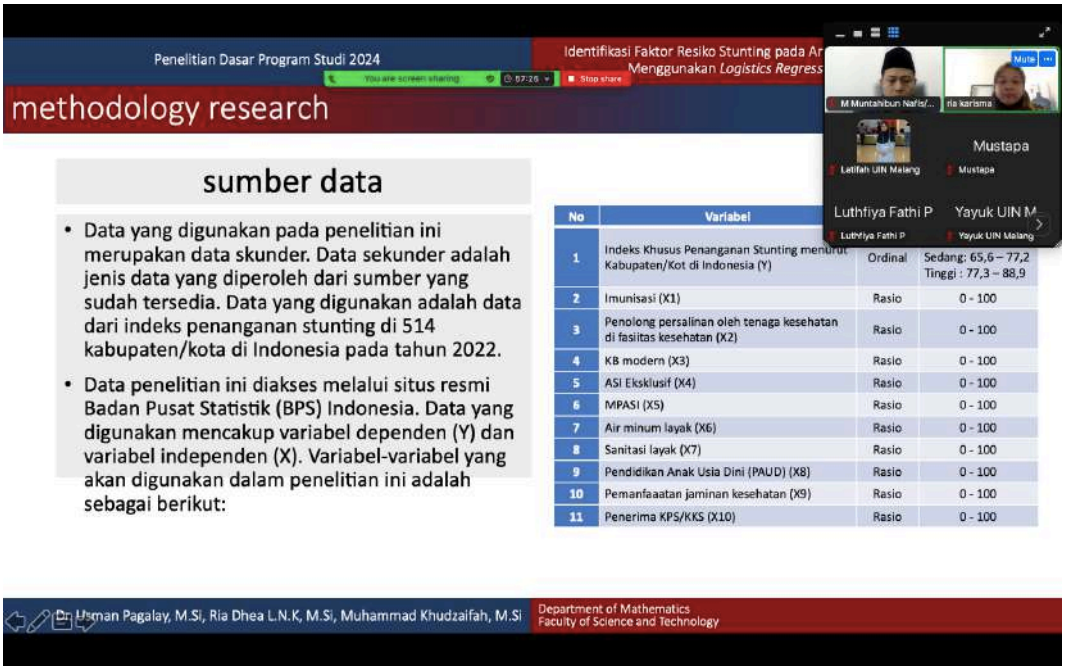
```
# View the results for each class
print(results)

$rendah
$rendah$sensitivity
[1] 1

$rendah$specificity
[1] 0.8

$rendah$accuracy
[1] 0.875


$sedang
$sedang$sensitivity
[1] 0.3333333

$sedang$specificity
[1] 0.8

$sedang$accuracy
[1] 0.625


$tinggi
$tinggi$sensitivity
[1] 0.5

$tinggi$specificity
[1] 0.8333333

$tinggi$accuracy
[1] 0.75
```

Lampiran 8. Dokumentasi Kegiatan Seminar Antara

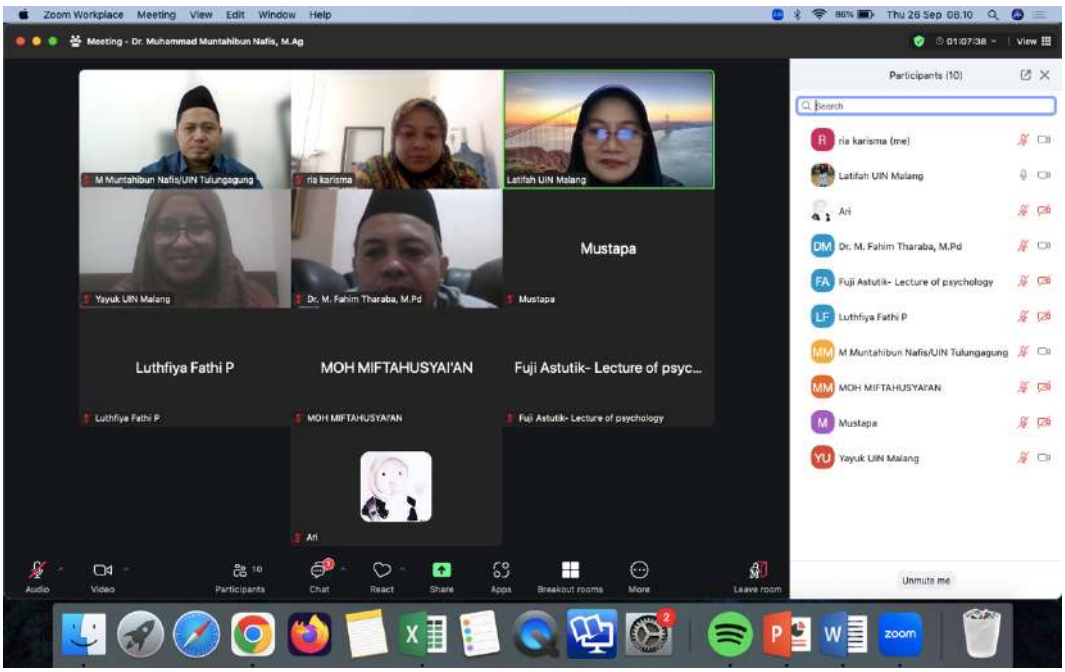








No	Variabel		
1	Indeks Khusus Penanganan Stunting menurut Kabupaten/Kot di Indonesia (Y)	Ordinal	Sedang: 65,6 – 77,2 Tinggi : 77,3 – 88,9
2	Imunisasi (X1)	Rasio	0 - 100
3	Penolong persalinan oleh tenaga kesehatan di fasiitas kesehatan (X2)	Rasio	0 - 100
4	KB modern (X3)	Rasio	0 - 100
5	ASI Eksklusif (X4)	Rasio	0 - 100
6	MPASI (X5)	Rasio	0 - 100
7	Air minum layak (X6)	Rasio	0 - 100
8	Sanitasi layak (X7)	Rasio	0 - 100
9	Pendidikan Anak Usia Dini (PAUD) (X8)	Rasio	0 - 100
10	Pemanfaaatan jaminan kesehatan (X9)	Rasio	0 - 100
11	Penerima KPS/KKS (X10)	Rasio	0 - 100


Zoom Workplace Meeting View Edit Window Help
Meeting - Dr. Muhammad Muntahibun Nafis, M.Ag
M Muntahibun Nafis/UIN Tulungagung
ria karisma
Latifah UIN Malang
Yayuk UIN Malang
Dr. M. Fahim Tharaba, M.Pd
Mustapa
Luthfiya Fathi P
MOH MIFTAHUSYAI'AN
Fuji Astutik- Lecture of psyc...
Ari
Participants (10)
ria karisma (me)
Latifah UIN Malang
Ari
Dr. M. Fahim Tharaba, M.Pd
Fuji Astutik- Lecture of psychology
Luthfiya Fathi P
M Muntahibun Nafis/UIN Tulungagung
MOH MIFTAHUSYAI'AN
Mustapa
Yayuk UIN Malang
Unmute me
Audio
Video
Participants
Chat
React
Share
Apps
Breakout rooms
More
Leave room

IDENTIFIKASI FAKTOR RESIKO STUNTING PADA ANAK DI BAWAH LIMA TAHUN DI INDONESIA MENGGUNAKAN LOGISTICS REGRESSION ENSEMBLES (LORENS)

Penelitian Dasar Program Studi 2024

Malang, 23 September 2024

Dr. Usman Pagalay, M.Si, Ria Dhea L.N.K, M.Si, Muhammad Khudzaifah, M.Si

Department of Mathematics
Faculty of Science and Technology

outline

1. Pendahuluan
2. Rumusan Masalah dan Tujuan Penelitian
3. Manfaat
4. Kajian Teori
5. Konsep
6. Metodologi Penelitian
7. Waktu Pelaksanaan
8. Anggaran Penelitian

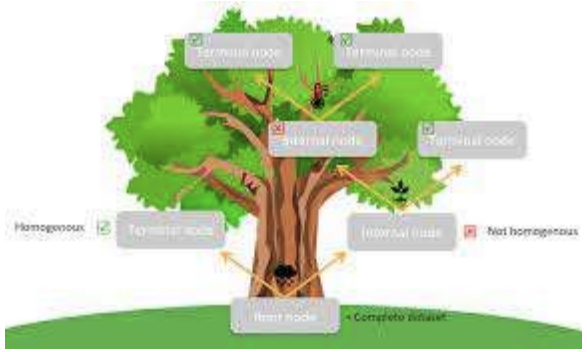
background

Indonesia merupakan negara berkembang yang memiliki angka prevalensi stunting urutan tertinggi ke-27 dari 154 negara di dunia. Stunting Indonesia menduduki peringkat ke-2 tertinggi stunting di Kawasan Asia Tenggara

Penyebab utama stunting antara lain asupan kalori yang tidak adekuat. Selain itu kebutuhan yang meningkat dimana yang banyak terjadi pada anak di bawah usia 5 tahun.

Pemerintah menargetkan tahun 2024 angka stunting turun hingga 14% pada tahun 2024.

Target Stunting di Indonesia masih jauh di atas ambang batas yang ditetapkan oleh WHO yaitu 20%.



Logistic Regression Ensemble (LORENS) merupakan salah satu metode klasifikasi pada *machine learning* dengan menggunakanakan teknik *ensemble* (penggabungan). Tujuan dari metode klasifikasi ini memperulah fungsi diskriminan yang optinal sehingga dapat memisahkan pengamatan yang berasal dari kelas berbeda

problem and aim

Bagaimana faktor resiko stunting di Indonesia dengan menggunakan metode LORENS?

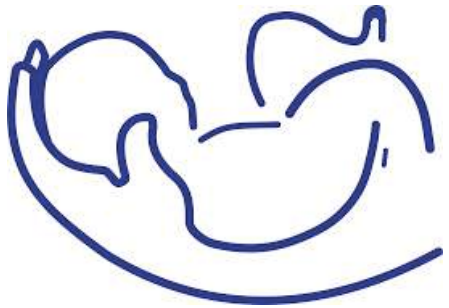
Bagaimana hasil ketepatan klasifikasi stunting dengan menggunakan LORENS

Memberikan kontribusi berupa masukan serta evaluasi kepada pihak terkait untuk menekan Angka Kematian Bayi berdasarkan model yang dihasilkan dengan menggunakan model *Random Forest*

Dr. Usman Pagalay, M.Si, Ria Dhea L.N.K, M.Si, Muhammad Khudzaifah, M.Si

Department of Mathematics
Faculty of Science and Technology

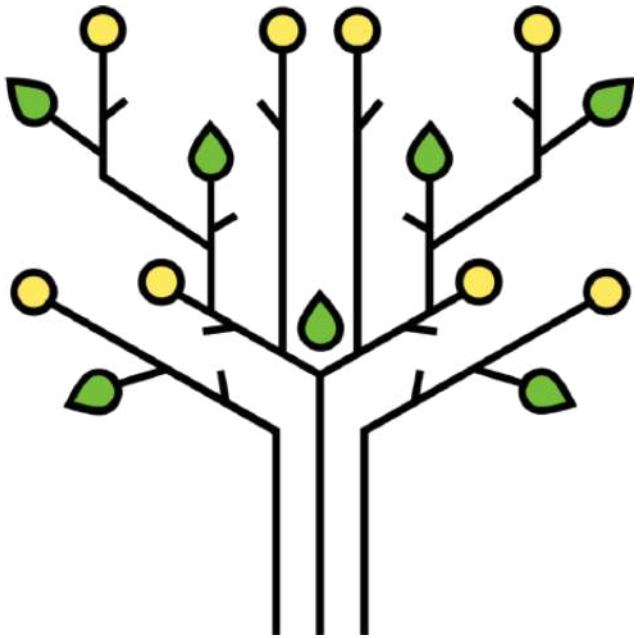
previous research



Stunting merupakan kondisi gagal tumbuh yang terjadi pada anak-anak akibat kurangnya asupan nutrisi yang memadai dalam jangka waktu yang lama, terutama pada periode 1.000 hari pertama kehidupan, yaitu dari kehamilan hingga usia dua tahun (WHO, 2015). Tingkat stunting yang tinggi dapat memberikan kontribusi pada rendahnya peringkat Indeks Pembangunan Manusia (*Human Development Index*/ HDI) sebuah negara

- 2019 oleh Utami, dkk
  - Metode regresi logistik multivariate digunakan untuk mengetahui faktor resiko stunting pada anak di Jakarta Selatan, yaitu pendapatan keluarga, tingkat pendidikan, dan kemampuan keluarga merupakan faktor resiko terjadinya stunting di wilayah tersebut (Utami dkk, 2019).
- 2019 oleh Widhianingsih dkk
  - LORENS untuk mempredikasi aplikasi obat untuk kanker dan memiliki performa yang lebih baik dibanding dengan regresi logistik berdasarkan kurva AUC, yaitu sebesar 0,7306
- 2017 Kuswanto dkk, 2015
  - LORENS memiliki hasil akurasi yang cukup tinggi yaitu 66% sampai 77%.

literature review



LOGISTIC REGRESSION

- Logistic Regression (LR) merupakan suatu metode analisis statistika untuk menganalisis model terbaik berdasarkan hubungan yang terjadi antara variabel prediktor (*independent*) dan variabel respon (*dependent*). Variabel peubah respon memiliki dua kategori atau lebih, sedangkan variabel prediktor berskala kategori maupun interval yang memiliki satu atau lebih variabel.
- Variabel respon dalam *binary logistic regression* memiliki dua kategori yaitu “gagal” dan “sukses”. Variabel respon kategori “gagal” dinotasikan dengan y=0 dan kategori “sukses” dinotasikan dengan y=1

LR CERP

Logistic Regression Classification by Ensemble from Random Partition (LR CERP) merupakan pengklasifikasian menggunakan *ensemble* di mana LR sebagai basis klasifikasi. LR CERP mempartisi data secara acak menjadi beberapa sub ruang yang *mutually exclusive* dengan ukuran yang sama.

Partisi optimal jika n (banyak data) $> p$ (banyak variabel prediktor) adalah

$$\frac{p}{i}$$

Partisi jika n (banyak data) $< p$ (banyak variabel prediktor) adalah

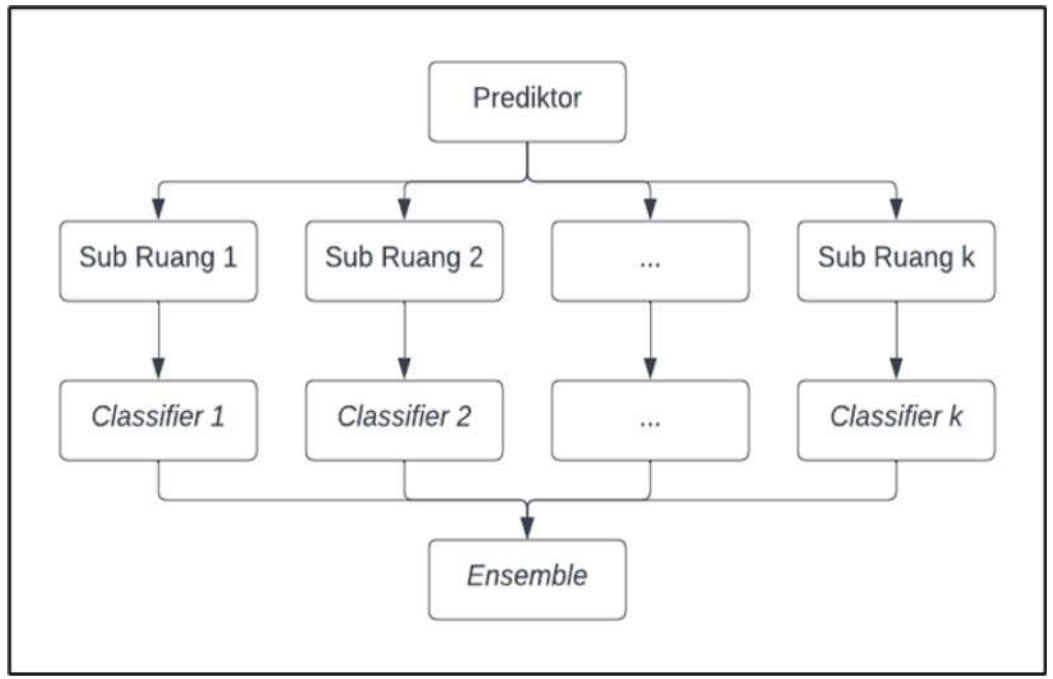
$$\frac{6 \times p}{n}$$

LR CERP digunakan untuk meningkatkan akurasi dengan mengkombinasikan hasil klasifikasi dari setiap sub ruang

Dr. Usman Pagalay, M.Si, Ria Dhea L.N.K, M.Si, Muhammad Khudzaifah, M.Si

Department of Mathematics
Faculty of Science and Technology

literature review

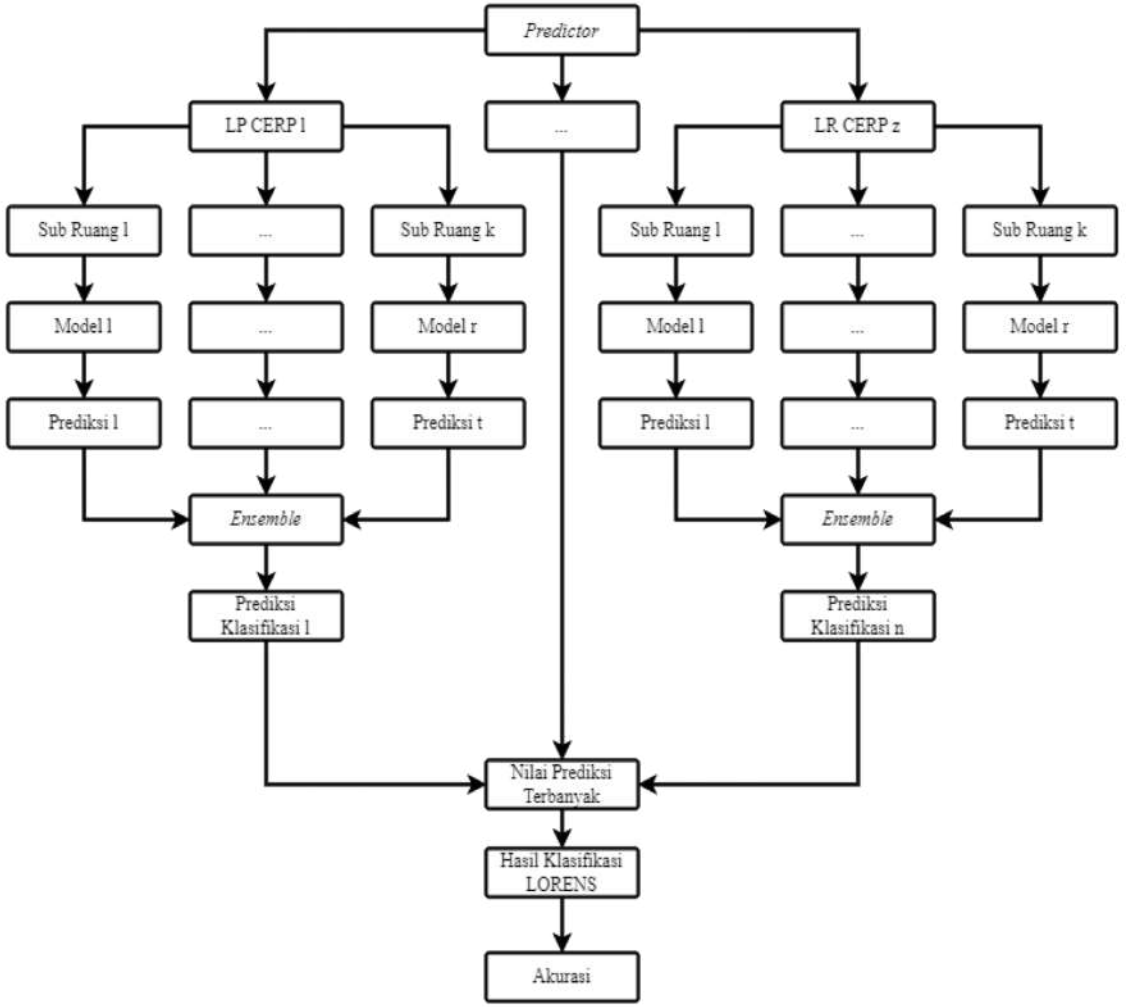


LORENS

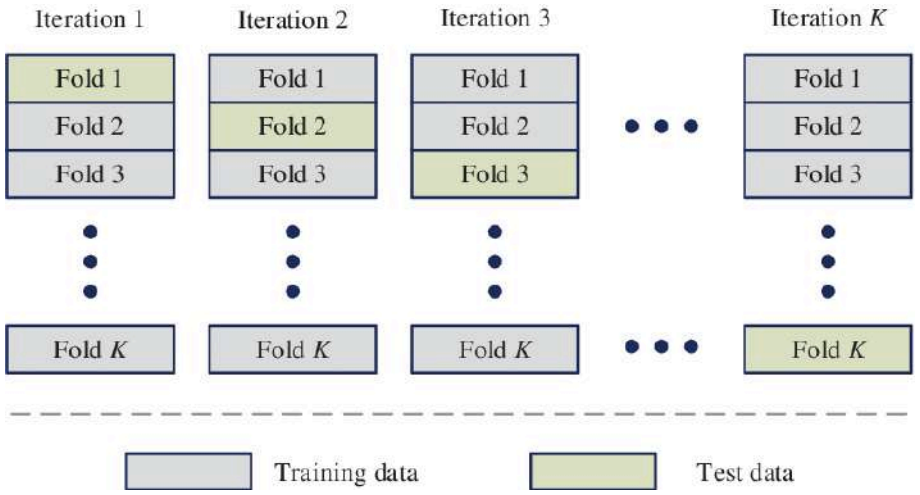
Dasar klasifikasi ini adalah LR dan dikembangkan berdasarkan LR CERP. LORENS membentuk beberapa *ensemble* dengan mengulangi prosedur LR CERP.

LORENS dapat meningkatkan akurasi dalam satu *ensemble* diperoleh dari hasil kombinasi dalam setiap *ensemble* yang dihasilkan. Umumnya ambang (*threshold*) probabilitas yang digunakan adalah 0,5, namun hal tersebut dirasa kurang baik apabila proporsi kelas 0 dan 1 tidak seimbang. LORENS menggunakan *threshold* optimal dalam klasifikasinya yaitu menggunakan $\frac{\bar{y}+0,5}{2}$

literature review



literature review



Cross validation

- Cross Validation merupakan metode untuk mengevaluasi model dengan membagi data menjadi k folds atau partisi yang memiliki jumlah seimbang. Berdasarkan partisi tersebut, cross validation juga membagi data testing dan training
- Semua partIsi digunakan sebagai data testing sekaligus data training. Penggunaan data sebagai data testing maupun training dilakukan pada saat gilirannya masing-masing.
- Misalkan dalam cross validation digunakan data testing sebanyak satu partisi, maka k-1 sisanya digunakan sebagai data training. Selanjutnya, dilakukan secara berulang hingga semua partisi telah dijadikan sebagai data testing. Prosedur ini disebut sebagai k fold cross validation

methodology research

sumber data

- Data yang digunakan pada penelitian ini merupakan data skunder. Data sekunder adalah jenis data yang diperoleh dari sumber yang sudah tersedia. Data yang digunakan adalah data dari indeks penanganan stunting di 514 kabupaten/kota di Indonesia pada tahun 2022.
- Data penelitian ini diakses melalui situs resmi Badan Pusat Statistik (BPS) Indonesia. Data yang digunakan mencakup variabel dependen (Y) dan variabel independen (X). Variabel-variabel yang akan digunakan dalam penelitian ini adalah sebagai berikut:

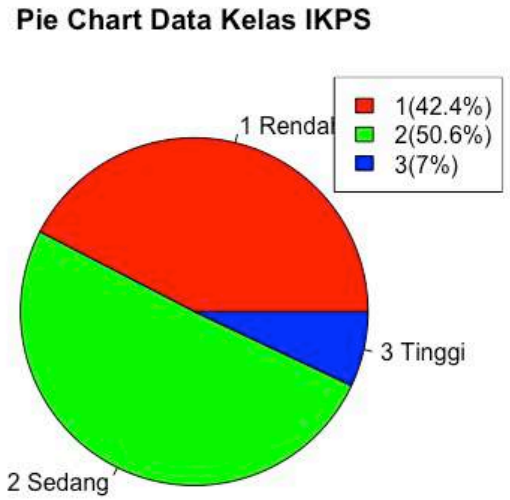
No	Variabel	Skala	Keterangan
1	Indeks Khusus Penanganan Stunting menurut Kabupaten/Kot di Indonesia (Y)	Ordinal	Rendah: 53,9 –65,5 Sedang: 65,6 – 77,2 Tinggi : 77,3 – 88,9
2	Imunisasi (X1)	Rasio	0 - 100
3	Penolong persalinan oleh tenaga kesehatan di fasiitas kesehatan (X2)	Rasio	0 - 100
4	KB modern (X3)	Rasio	0 - 100
5	ASI Eksklusif (X4)	Rasio	0 - 100
6	MPASI (X5)	Rasio	0 - 100
7	Air minum layak (X6)	Rasio	0 - 100
8	Sanitasi layak (X7)	Rasio	0 - 100
9	Pendidikan Anak Usia Dini (PAUD) (X8)	Rasio	0 - 100
10	Pemanfaaatan jaminan kesehatan (X9)	Rasio	0 - 100
11	Penerima KPS/KKS (X10)	Rasio	0 - 100

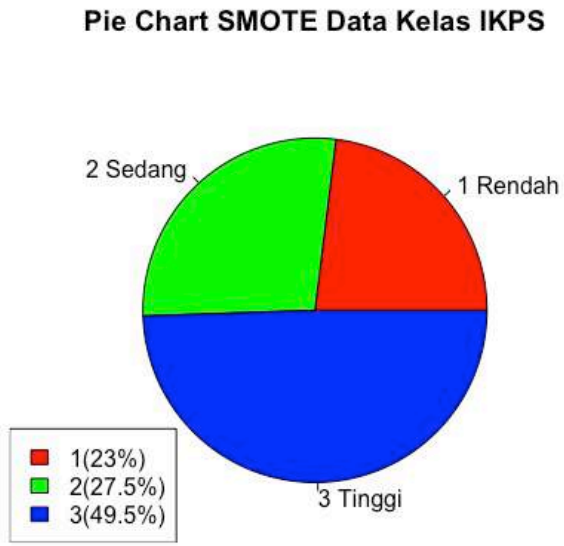
Research Steps

1. Melakukan statistik deskriptif untuk mengetahui karakteristik data
2. Melakukan *resempling* dengan metode SMOTE untuk mengatasi data *imbalance*
3. Melakukan sub sampling dari kumpulan data atau membagi data menjadi data *training* dan *data testing*. Dimana data *training* pada penelitian ini 85% dari jumlah data dan data *testing* 15%. Data *training* digunakan untuk membentuk model sedangkan data *testing* untuk mengukur tingkat kesalahan.
4. Membentuk jumlah *l* partisi dan *m ensemble* yang akan diterapkan
5. Mempartisi variabel prediktor ke dalam *k* sub ruang
6. Membangun model regresi logistik setiap partisi dari data training
7. Menghitung partisi dengan mensubsitusikan data testing ke model
8. Menghitung nilai rata-ratanya setelah nilai prediksi diperoleh
9. Ulangi langkah 4 sampai 8 hingga terbentuk *n ensemble*
10. Mencari nilai prediksi terbanyak dari pengamatan di semua *ensemble*
11. Menghitung *threshold* yang paling optimal
12. Mengklasifikasikan setiap nilai prediksi menjadi 0 atau 1 berdasarkan nilai *threshold*
13. Menghitung ketepatan hasil klasifikasi dengan nilai *sensitivity, specificity*, dan *accuracy*
14. Melakukan evaluasi performa model dengan menggunakan *cross validation*
15. Menarik kesimpulan

Discussion

- Descriptive Statistics





Discussion

$$logit(P) = \log\left(\frac{P}{1-P}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_{10} X_{10} \quad (4.1)$$

Berikut model terbaik untuk klasifikasi IKPS yang didapatkan dari LOREN dari persamaan (4.1) dimana model tyang terbentuk tanpa memperhatikan variabel yang tidak signifikan

$$P(Y \leq j) = \frac{1}{1+e^{-(\pi_j - \theta)}} \quad (4.2)$$

Dimana

π_j : intersep atau ambang batas untuk kategori j

θ : merupakan prediktor linear yang memiliki definisi sebagai $\theta = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_{10} X_{10}$

$\beta_0, \beta_1, \beta_2, \ldots, \beta_{10}$ merupakan koefisien regresi

$X_1, X_2, \ldots, X_{10}$: variabel prediktor

Maka dari persamaan (4.2) didapatkan,

$$\theta = 0{,}0272X_1 + 0{,}0747X_2 + 0{,}0461X_3 + 0{,}0103X_4 + + 0{,}137X_5 + 0{,}085X_6 + 0{,}134X_7 + 0{,}216X_8 + 0{,}084X_9 + 0{,}137X_{10}$$

Intersep dari model terbaik LORENS didapatkan

$\pi_1 = 60{,}049$ dimana sebagai pemisah antara IKPS rendah (kategori 1) dan IKPS sedang atau tinggi (kategori 2 atau 3)

$\pi_2 = 70{,}71$ dimana sebagai pemisah anatara IKPS rendah atau sedang (kategori 1 atau 2) dan IKPS tinggi (kategori 3).

Fungsi peluang kumulatif untuk masing-masing kategori yaitu,

Peluang kumulatif IKPS Rendah (Kategori 1)

$$P(Y \leq 1) = \frac{1}{1+e^{-(60{,}049-\theta)}}$$

Peluang kumulatif IKPS Sedang (Kategori 2)

$$P(Y \leq 2) = \frac{1}{1+e^{-(70{,}718-\theta)}}$$

Peluang kumulatif IKPS Tinggi (Kategori 3)

Persamaan (4.4) merupakan peluang IKPS Rendah dengan kategori 1 dimana $P(Y = 1) = P(Y \leq 1)$. Sedangkan persamaan (4.5) merupakan peluang IKPS Sedang dengan kategori 2 dimana $P(Y = 2) = P(Y \leq 2) - P(Y \leq 1)$. Maka untuk peluang IKPS Tinggi atau kategori 3 didapatkan:

$$P(Y = 3) = 1 - P(Y \leq 2) = 1 - \frac{1}{1 + e^{-(70{,}718-\theta)}}$$

Discussion

Nilai Prediksi

Nilai prediksi merupakan prediksi yang diperoleh dengan mensubsitusikan data testing pada model yang sebelumnya telah diperoleh. Kemudian perhitungan ini dilakukan pada masing-masing *ensemble*. Berikut hasil prediksi untuk masing-masing *ensemble*

Data ke-	Ensembel									
	1	2	3	4	5	6	7	8	9	10
578	0,388	0,339	0,360	0,421	0,441	0,373	0,306	0,405	0,396	0,361
549	0,415	0,408	0,357	0,423	0,4307	0,405	0,372	0,421	0,369	0,402
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
259	0,271	0,290	0,279	0,280	0,257	0,266	0,281	0,264	0,292	0,290

Discussion

- ***Threshold* Optimal**

$$Threshold = \frac{(\bar{y} + 0{,}5)}{2} = \frac{\left(\frac{182}{804} + 0{,}5\right)}{2} = 0{,}363184$$

Sehingga nila *threshold* optimal yang digunakan dalam penelitian ini yaitu 0,363. Nilai ini digunakan untuk menentukan kelas klasifikasi pada masing-masing prediksi.

Discussion

○ Akurasi Model Prediksi

Akurasi model prediksi dapat dilihat menggunakan data *training*. Hasil akurasi dapat diperoleh dari tabulasi silang antara *True Positive* (TP), *False Positive* (FP), *True Negative* (TN,) dan *False Negative* (FN) dengan menggunakan data training.

		Kelas Aktual					
		Rendah		Sedang		Tinggi	
		p(+)	n(-)	p(+)	n(-)	p(+)	n(-)
Kelas Prediksi	p(+)	171	612	203	563	392	395
	n(-)	10	11	17	21	11	6

IKPS	Akurasi	Sensitivity	Specitivity
Rendah	0,9653727	0,9653727	0,9789128
Sedang	0,8874541	0,8874541	0,982162
Tinggi	0,9900974	0,9900974	0,9697035

Discussion

○ Evaluasi Performa Model

Model yang terbentuk kemudian dievaluasi dengan menggunakan data *testing*. Dimana fungsi data *testing* digunaan untuk kontrol atau evaluasi terhadap model yang didapatkan. Evaluasi performa model ini dapat dilihat dengan ketepatan klasfikasi. Ketepatan klasifikasi dapat diperoleh melalui hasil dari perhitungan tabulasi silang dimana terdiri dari kelas *True Positive* (TP), *False Positive* (FP), *True Negative* (TN,) dan *False Negative* (FN). Tabel 4.3 berikut merupakan tabel tabulasi silang dengan 10 partisi dan 10 ensembel.

		Kelas Aktual					
		Rendah		Sedang		Tinggi	
		p(+)	n(-)	p(+)	n(-)	p(+)	n(-)
Kelas Prediksi	p(+)	34	2	31	5	70	0
	n(-)	106	0	104	2	67	5

Discussion

○ Evaluasi Performa Model

Model yang terbentuk kemudian dievaluasi dengan menggunakan data *testing*. Dimana fungsi data *testing* digunaan untuk kontrol atau evaluasi terhadap model yang didapatkan. Evaluasi performa model ini dapat dilihat dengan ketepatan klasfikasi. Ketepatan klasifikasi dapat diperoleh melalui hasil dari perhitungan tabulasi silang dimana terdiri dari kelas *True Positive* (TP), *False Positive* (FP), *True Negative* (TN,) dan *False Negative* (FN). Tabel 4.3 berikut merupakan tabel tabulasi silang dengan 10 partisi dan 10 ensembel.

		Kelas Aktual					
		Rendah		Sedang		Tinggi	
		p(+)	n(-)	p(+)	n(-)	p(+)	n(-)
Kelas Prediksi	p(+)	34	2	31	5	70	0
	n(-)	106	0	104	2	67	5

IKPS	Akurasi	Sensitivity	Specitivity
Rendah	0,9859	0,9444	1,0000
Sedang	0,9507	0,8611	0,9811
Tinggi	0,9648	1,0000	0,9306

IKPS	Error rate
Rendah	0
Sedang	0,0606
Tinggi	0,0667

Conclussion

1. Peluang untuk IKPS Kategori Tinggi memiliki peluang 0 yang artinya peluang untuk berada di kategori ini sangat kecil mendekati nol dibanding dengan kategori IKPS yang lain. Koefesien model LOREN menunjukkan bahwa variabel Imunisasi, Penolong Persalinan, KB, ASI Ekslusif, Air Minum Layak dan Jaminan Kesehatan dan Jamkesda memberikan koefesien yang rendah. Artinya variabel-variabel tersebut diakukan fokus perbaikan
2. Ketepatan hasil klasifikasi yang didapatkan dari model yaitu 98,59%, 95,07% dan 96,48%. Artinya model telah tepat terprediksi dengan baik karena mendekati nilai 100%.

Suggestion

Saran dari penelitian berdasarkan analisis yang dilakukan serta kesimpulan, yaitu memeriksa kembali *syntax* terutama pada bagian partisi dan ensembel. Saran untuk pihak terkait yaitu, melakukan pengawasan dan penanganan terhadap variabel yang memiliki koefesien rendah. Karena memerlukan fokus agar dapat mengatasi masalah penanganan stunting. Perlunya edukasi massif terhadap koefesien-koefesien tersebut agar tidak tertinggal di kawasan Asia

references

Agresti, A. (2013). *Categorical Data Analysis* (3rd ed.). John Wiley & Sons.

Ahn, H., Moon, H., Fazzari, M. J., Lim, N., Chen, J. J., & Kodell, R. L. (2007). *Classification by ensembles from random partitions of high-dimensional data*. *51*, 6166–6179. https://doi.org/10.1016/j.csda.2006.12.043

Badan Penelitian dan Pengembangan Kesehatan. (2018). Laporan Nasional Riskesdas 2018. Kementerian Kesehatan RI.

Fawcett, Tom (2006). "An Introduction to ROC Analysis" (PDF). Pattern Recognition Letters. 27 (8): 861–874. doi:10.1016/j.patrec.2005.10.010. S2CID 2027090

Hosmer, D. W., & Lemeshow, S. (2000). *Applied Logistic Regression second Edition*. Wiley Series in Probability and Statistics.

Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied Logistic Regression Third Edition. In *Technometrics* (Vol. 38, Issue 2). https://doi.org/10.2307/1270433

Kemenkes. (2018). Mengenal Stunting dan Gizi Buruk. Penyebab, Gejala, Dan Mencegah https://promkes.kemkes.go.id/?p=8486

Kemenkes. (2023). https://ayosehat.kemkes.go.id/cegah-stunting-itu-penting

Kuswanto, H., Asfihani, A., Sarumaha, Y., Ohwada, H., (2015). Logistic Regression Ensemble for Predicting Customer Defection with Very Large Sample Size. Procedia Computer Science. 72. 86-93. https://doi.org/10.1016/j.procs.2015.12.108

Lim, N. (2007). *Classification by Ensembles from Random Partitions using Logistic Regression Models*. Stony Brook University.

Lim, N., Ahn, H., Moon, H., & Chen, J. J. (2010). *Classification of High-Dimensial Data With Ensemble of Logistic Regression Model*. 160–171. https://doi.org/10.1080/10543400903280639

Melasasi, J. N. (2015). *Klasifikasi Enzim Pada Database Dud-E Dengan Metode Logistic Regression Ensemble (Lorens) Dengan Metode Logistic Regression*. Institut Teknologi Sepuluh Nopember.

Studi Statis Gizi Sehat. (2021). https://www.badankebijakan.kemkes.go.id/buku-saku-hasil-studi-status-gizi-indonesia-ssgi-tahun-2021/

WHO. (2015). https://www.who.int/news/item/19-11-2015-stunting-in-a-nutshell#:~:text=Stunting%20is%20the%20impaired%20growth,WHO%20Child%20Growth%20Standards%20median.

Widhianingsih, T. D. A., Kuswanto, H., & Prastyo, D. D. (2020). Logistic Regression Ensemble (LORENS) Applied to Drug Discovery. *Matematika*, *36*(1), 43–49. https://doi.org/10.11113/matematika.v36.n1.1197

UNDP, 2021, Technical notes, https://hdr.undp.org/sites/default/files/2021-22_HDR/hdr2021-22_technical_notes.pdf

Utami, R. A., Setiawan, A., & Fitriyani, P. (2019). Identifying causal risk factors for stunting in children under five years of age in South Jakarta, Indonesia. Enfermeria Clinica, 29, 606-611. https://doi.org/10.1016/j.enfcli.2019.04.093

references

Agresti, A. (2013). *Categorical Data Analysis* (3rd ed.). John Wiley & Sons.

Ahn, H., Moon, H., Fazzari, M. J., Lim, N., Chen, J. J., & Kodell, R. L. (2007). *Classification by ensembles from random partitions of high-dimensional data*. *51*, 6166–6179. https://doi.org/10.1016/j.csda.2006.12.043

Badan Penelitian dan Pengembangan Kesehatan. (2018). Laporan Nasional Riskesdas 2018. Kementerian Kesehatan RI.

Fawcett, Tom (2006). "An Introduction to ROC Analysis" (PDF). Pattern Recognition Letters. 27 (8): 861–874. doi:10.1016/j.patrec.2005.10.010. S2CID 2027090

Hosmer, D. W., & Lemeshow, S. (2000). *Applied Logistic Regression second Edition*. Wiley Series in Probability and Statistics.

Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied Logistic Regression Third Edition. In *Technometrics* (Vol. 38, Issue 2). https://doi.org/10.2307/1270433

Kemenkes. (2018). Mengenal Stunting dan Gizi Buruk. Penyebab, Gejala, Dan Mencegah https://promkes.kemkes.go.id/?p=8486

Kemenkes. (2023). https://ayosehat.kemkes.go.id/cegah-stunting-itu-penting

Kuswanto, H., Asfihani, A., Sarumaha, Y., Ohwada, H., (2015). Logistic Regression Ensemble for Predicting Customer Defection with Very Large Sample Size. Procedia Computer Science. 72. 86-93. https://doi.org/10.1016/j.procs.2015.12.108

Lim, N. (2007). *Classification by Ensembles from Random Partitions using Logistic Regression Models*. Stony Brook University.

Lim, N., Ahn, H., Moon, H., & Chen, J. J. (2010). *Classification of High-Dimensial Data With Ensemble of Logistic Regression Model*. 160–171. https://doi.org/10.1080/10543400903280639

Melasasi, J. N. (2015). *Klasifikasi Enzim Pada Database Dud-E Dengan Metode Logistic Regression Ensemble (Lorens) Dengan Metode Logistic Regression*. Institut Teknologi Sepuluh Nopember.

Studi Statis Gizi Sehat. (2021). https://www.badankebijakan.kemkes.go.id/buku-saku-hasil-studi-status-gizi-indonesia-ssgi-tahun-2021/

WHO. (2015). https://www.who.int/news/item/19-11-2015-stunting-in-a-nutshell#:~:text=Stunting%20is%20the%20impaired%20growth,WHO%20Child%20Growth%20Standards%20median.

Widhianingsih, T. D. A., Kuswanto, H., & Prastyo, D. D. (2020). Logistic Regression Ensemble (LORENS) Applied to Drug Discovery. *Matematika*, *36*(1), 43–49. https://doi.org/10.11113/matematika.v36.n1.1197

UNDP, 2021, Technical notes, https://hdr.undp.org/sites/default/files/2021-22_HDR/hdr2021-22_technical_notes.pdf

Utami, R. A., Setiawan, A., & Fitriyani, P. (2019). Identifying causal risk factors for stunting in children under five years of age in South Jakarta, Indonesia. Enfermeria Clinica, 29, 606-611. https://doi.org/10.1016/j.enfcli.2019.04.093