

DAFTAR PUSTAKA

- A. K. Santra, C. J. (2012). Genetic Algorithm and Confusion Matrix for Document Clustering. *IJCSI International Journal of Computer Science Issues*, 322-328.
- Afifah, L. (2023). *Algoritma K-Nearest Neighbor (KNN) untuk Klasifikasi*. Retrieved from IlmudataPy: <https://ilmudatapy.com/algoritma-k-nearest-neighbor-knn-untuk-klasifikasi>
- Ami Rahmawati, D. W. (2019). Klasifikasi Naïve Bayes untuk Mendiagnosis Penyakit Pneumonia pada Anak Balita (Studi Kasus: UPTD Puskesmas Sukaraja Sukabumi). *Kumpulan Jurnal Ilmu Komputer (KLIK)*, 241-253.
- Aytug Onan. Serdar Korukoglu, H. B. (2016). Ensemble of keyword extraction methods and classifiers in text classification. *ELSEVIER*, 232–247.
- Bramer, M. (2020). *Principles of Data Mining*. United Kingdom: Springer.
- Drajana, I. C. (2017). Metode Support Vector Machine dan Forward Selection Prediksi Pembayaran Pembelian Bahan Baku KOPRA. *ILKOM Jurnal Ilmiah*, 116-123.
- Elyza Gustri Wahyuni, A. S. (2018). Sistem diagnosis pneumonia menggunakan logika Fuzzy Tsukamoto dan Pneumonia Severity Index (PSI). *CITEE*, 24-26.
- Firdaus, I. A. (2022). Deteksi Infeksi Mycoplasma Pneumoniae Pneumonia Menggunakan Komparasi Algoritma Klasifikasi Machine Learning. *(JOINTECS) Journal of Information Technology and Computer Science*, 35-42.
- Géron, A. (2019). Hands-on Machine Learning with Scikit-Learning, Keras and Tensorflow. *O'Reilly Media*.
- Gunawan, D. (2016). Evaluasi Performa Pemecahan Database dengan Metode Klasifikasi pada Data Preprocessing Data Mining. *Khazanah Informatika*, 10-13.
- Gustavo Peixoto de Oliveira, A. F. (2022). Diabetes diagnosis based on hard and soft voting classifiers combining statistical learning models. *Brazilian Journal of Biometrics*, 415–427.
- Haijing Tang, J. Z. (2018). Explore Machine Learning for Analysis and Prediction of Lung Cancer . *Association for Computing Machinery*, 41-45.
- Imam Cholissodin, E. R. (2016). *SWARM INTELLIGENCE (Teori & Case Study)*. Indonesia: ResearchGate.

- Jiawei Han, M. K. (2012). *Data Mining Concepts and Techniques*. United States of America: Elsevier.
- Joshua P. Metlay, G. W. (2019). Diagnosis and Treatment of Adults with Community-acquired Pneumonia. *American Journal of Respiratory and Critical Care Medicine*, 45-67.
- Kementerian Kesehatan Republik Indonesia. (2021). *Profil Kesehatan Indonesia 2020*. Indonesia: Kementerian Kesehatan Republik Indonesia.
- Kuang Ming Kuo, P. C. (2019). Predicting hospital-acquired pneumonia among schizophrenic patients: a machine learning approach. *BMC Medical Informatics and Decision Making*, 1-8.
- Kumar, A. (2020, September 7). *Hard vs Soft Voting Classifier Python Example*. Retrieved from Data Analytics: <https://vitalflux.com/hard-vs-soft-voting-classifier-python-example>
- Kurnianingsih, N. F. (2020). Ensemble Learning Based on Soft Voting for Detecting Methamphetamine in Urine. *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 1-6.
- Md. Mehedi Hasan, M. O. (2021). Prediction of Pneumonia Disease of Newborn Baby Based on Statistical Analysis of Maternal Condition Using Machine Learning Approach. *IEEE Xplore*, 919-924.
- Nair, A. (2021, October 12). *Combine Your Machine Learning Models With Voting*. Retrieved from Towards Data Science: <https://towardsdatascience.com/combine-your-machine-learning-models-with-voting-fa1b42790d84>
- Nikolaos Peppes, E. D. (2021). Performance of Machine Learning-Based Multi-Model Voting Ensemble Methods for Network Threat Detection in Agriculture 4.0. *sensors*, 1-17.
- Oknalita Simbolon, M. N. (2020). Predicting the Risk of Preeclampsia using Soft Voting-based Ensemble and Its Recommendation. *IEEE*, 1-6.
- Pang-Ning Tan, M. S. (Introduction to Data Mining). 2019. United Kingdom: Pearson Education.
- Ray, S. (2017, September 13). *Learn How to Use Support Vector Machines (SVM) for Data Science*. Retrieved from Analytics Vidhya: <https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code>
- Razky Josefa, R. S. (2019). Sistem Pakar Diagnosa Penyakit Pneumonia Pada Anak Menggunakan Metode Case Based Reasoning. *SAINTEKS*, 868-872.

- Saifullah. (2019). Deteksi Kelayakan Fisik Air untuk Konsumsi menggunakan Naive Bayes Classifier. *Universitas Islam Negeri Maulana Malik Ibrahim*. Indonesia.
- Sanjeevi, M. (2017, Oktober 7). *Chapter 4: Decision Trees Algorithms*. Retrieved from Medium: <https://medium.com/deep-math-machine-learning-ai/chapter-4-decision-trees-algorithms-b93975f7a1f1>
- Sasongko, T. B. (2016). Komparasi dan Analisis Kinerja Model Algoritma SVM dan PSO-SVM (Studi Kasus Klasifikasi Jalur Minat SMA). *Jurnal Teknik Informatika dan Sistem Informasi*, 244-253.
- Scikit-learn. (2023). *Support Vector Machines*. Retrieved from scikit-learn: <https://scikit-learn.org/stable/modules/svm.html>
- Sembiring, K. (2007). *Tutorial SVM Bahasa Indonesia*. Retrieved from Analytics Vidhya: <https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code>
- Syed Waseem Abbas Sherazi, J.-W. B. (2021). A soft voting ensemble classifier for early prediction and diagnosis of occurrences of major adverse cardiovascular events for STEMI and NSTEMI during 2-year follow-up in patients with acute coronary syndrome. *PLOS ONE*, 1-20.
- Tareq M. Shami, A. A.-S.-T. (2022). Particle Swarm Optimization: A Comprehensive Survey. *IEEE Access*, 10031-10061.
- Thorn, J. (2020, Maret 9). *Decision Trees Explained*. Retrieved from Towards Data Science: <https://towardsdatascience.com/decision-trees-explained-3ec41632ceb6>
- UNICEF. (2019, November 12). *Lembaga kesehatan dan anak memperingatkan satu anak meninggal akibat pneumonia setiap 39 detik*. Retrieved from UNICEF Indonesia: <https://www.unicef.org/indonesia/id/siaran-pers/lembaga-kesehatan-dan-anak-memperingatkan-satu-anak-meninggal-akibat-pneumonia-setiap-39>
- WHO. (2022, November 11). *Pneumonia in children*. Retrieved from World Health Organization: <https://www.who.int/news-room/fact-sheets/detail/pneumonia>

Lampiran 1. Hasil probabilitas model *K-Nearest Neighbor*, *Support Vector Machine*, *Decision Tree*, dan *Soft Voting*.

Index	Y-True	K-NN		SVM		Decision Tree		Soft Voting	
		0	1	0	1	0	1	0	1
0	1	0.800	0.200	0.096	0.904	0.000	1.000	0.424	0.576
1	1	0.200	0.800	0.012	0.988	0.000	1.000	0.105	0.895
2	0	1.000	0.000	0.968	0.032	0.987	0.013	0.992	0.008
3	1	0.800	0.200	0.000	1.000	0.000	1.000	0.416	0.584
4	1	0.200	0.800	0.012	0.988	0.000	1.000	0.105	0.895
5	0	1.000	0.000	0.995	0.005	0.987	0.013	0.994	0.006
6	0	1.000	0.000	0.991	0.009	0.987	0.013	0.994	0.006
7	1	0.000	1.000	0.000	1.000	0.000	1.000	0.000	1.000
8	1	0.000	1.000	0.000	1.000	0.000	1.000	0.000	1.000
9	0	1.000	0.000	0.976	0.024	0.987	0.013	0.993	0.007
10	1	0.200	0.800	0.000	1.000	0.028	0.972	0.115	0.885
11	0	0.600	0.400	0.991	0.009	0.987	0.013	0.786	0.214
12	1	0.200	0.800	0.013	0.987	0.000	1.000	0.105	0.895
13	0	1.000	0.000	0.986	0.014	0.987	0.013	0.994	0.006
14	1	0.200	0.800	0.020	0.980	0.028	0.972	0.117	0.883
15	1	0.000	1.000	0.945	0.055	0.987	0.013	0.470	0.530
16	0	1.000	0.000	0.971	0.029	0.987	0.013	0.992	0.008
17	0	0.600	0.400	0.984	0.016	0.987	0.013	0.785	0.215
18	0	0.800	0.200	0.079	0.921	0.625	0.375	0.668	0.332
19	1	0.000	1.000	0.008	0.992	0.000	1.000	0.001	0.999
20	0	1.000	0.000	0.982	0.018	0.987	0.013	0.993	0.007
21	1	0.400	0.600	0.000	1.000	0.000	1.000	0.208	0.792
22	0	0.600	0.400	0.991	0.009	0.987	0.013	0.786	0.214
23	1	0.200	0.800	0.029	0.971	0.028	0.972	0.117	0.883
24	0	1.000	0.000	0.972	0.028	0.987	0.013	0.992	0.008
25	1	0.200	0.800	0.106	0.894	0.000	1.000	0.112	0.888
26	0	0.800	0.200	0.907	0.093	0.625	0.375	0.740	0.260
27	1	0.000	1.000	0.010	0.990	0.000	1.000	0.001	0.999
28	1	0.800	0.200	0.023	0.977	0.000	1.000	0.418	0.582
29	0	1.000	0.000	0.979	0.021	0.987	0.013	0.993	0.007
30	0	1.000	0.000	0.979	0.021	0.987	0.013	0.993	0.007
31	0	0.200	0.800	0.995	0.005	0.987	0.013	0.578	0.422
32	0	0.600	0.400	0.291	0.709	0.625	0.375	0.581	0.419
33	0	1.000	0.000	0.991	0.009	0.987	0.013	0.994	0.006
34	0	1.000	0.000	0.985	0.015	0.987	0.013	0.993	0.007
35	0	1.000	0.000	0.976	0.024	0.987	0.013	0.993	0.007
36	0	1.000	0.000	0.941	0.059	0.987	0.013	0.989	0.011
37	0	1.000	0.000	0.984	0.016	0.987	0.013	0.993	0.007
38	0	1.000	0.000	0.993	0.007	0.987	0.013	0.994	0.006
39	0	1.000	0.000	0.956	0.044	0.987	0.013	0.991	0.009

40	0	1.000	0.000	0.995	0.005	0.987	0.013	0.994	0.006
41	0	0.800	0.200	0.983	0.017	0.987	0.013	0.889	0.111
42	0	1.000	0.000	0.987	0.013	0.987	0.013	0.994	0.006
43	1	0.200	0.800	0.013	0.987	0.000	1.000	0.105	0.895
44	0	1.000	0.000	0.959	0.041	0.987	0.013	0.991	0.009
45	0	0.200	0.800	0.994	0.006	0.987	0.013	0.578	0.422
46	0	1.000	0.000	0.992	0.008	0.987	0.013	0.994	0.006
47	1	0.200	0.800	0.012	0.988	0.000	1.000	0.105	0.895
48	1	0.000	1.000	0.004	0.996	0.028	0.972	0.011	0.989
49	1	0.800	0.200	0.042	0.958	0.000	1.000	0.419	0.581
50	0	1.000	0.000	0.976	0.024	0.987	0.013	0.993	0.007
51	1	0.400	0.600	0.012	0.988	0.000	1.000	0.209	0.791
52	0	1.000	0.000	0.985	0.015	0.987	0.013	0.993	0.007
53	0	1.000	0.000	0.958	0.042	0.987	0.013	0.991	0.009
54	0	0.600	0.400	0.979	0.021	0.987	0.013	0.785	0.215
55	0	0.800	0.200	0.111	0.889	0.000	1.000	0.425	0.575
56	1	0.400	0.600	0.000	1.000	0.000	1.000	0.208	0.792
57	0	1.000	0.000	0.952	0.048	0.987	0.013	0.990	0.010
58	1	0.000	1.000	0.000	1.000	0.000	1.000	0.000	1.000
59	0	1.000	0.000	0.964	0.036	0.987	0.013	0.992	0.008
60	0	1.000	0.000	0.997	0.003	0.987	0.013	0.995	0.005
61	0	1.000	0.000	0.993	0.007	0.987	0.013	0.994	0.006
62	0	1.000	0.000	0.994	0.006	0.987	0.013	0.994	0.006
63	0	1.000	0.000	0.958	0.042	0.987	0.013	0.991	0.009
64	1	0.000	1.000	0.007	0.993	0.028	0.972	0.012	0.988
65	0	0.800	0.200	0.963	0.037	0.987	0.013	0.887	0.113
66	1	0.200	0.800	0.007	0.993	0.028	0.972	0.116	0.884
67	0	1.000	0.000	0.995	0.005	0.987	0.013	0.994	0.006
68	0	0.800	0.200	0.111	0.889	0.000	1.000	0.425	0.575
69	1	0.400	0.600	0.000	1.000	0.000	1.000	0.208	0.792
70	1	0.000	1.000	0.004	0.996	0.028	0.972	0.011	0.989
71	0	1.000	0.000	0.968	0.032	0.987	0.013	0.992	0.008
72	0	1.000	0.000	0.978	0.022	0.987	0.013	0.993	0.007
73	1	0.000	1.000	0.009	0.991	0.000	1.000	0.001	0.999
74	0	1.000	0.000	0.984	0.016	0.987	0.013	0.993	0.007
75	0	1.000	0.000	0.958	0.042	0.987	0.013	0.991	0.009
76	0	1.000	0.000	0.998	0.002	0.987	0.013	0.995	0.005
77	0	1.000	0.000	0.977	0.023	0.987	0.013	0.993	0.007
78	0	1.000	0.000	0.985	0.015	0.987	0.013	0.994	0.006
79	1	0.400	0.600	0.019	0.981	0.000	1.000	0.209	0.791
80	0	1.000	0.000	0.973	0.027	0.987	0.013	0.992	0.008
81	0	0.200	0.800	0.946	0.054	0.987	0.013	0.574	0.426
82	1	0.200	0.800	0.013	0.987	0.000	1.000	0.105	0.895
83	1	0.600	0.400	0.959	0.041	0.987	0.013	0.783	0.217
84	0	1.000	0.000	0.962	0.038	0.987	0.013	0.991	0.009

85	0	1.000	0.000	0.989	0.011	0.987	0.013	0.994	0.006
86	1	0.200	0.800	0.006	0.994	0.000	1.000	0.104	0.896
87	0	1.000	0.000	0.970	0.030	0.987	0.013	0.992	0.008
88	1	0.200	0.800	0.100	0.900	0.000	1.000	0.112	0.888
89	1	0.000	1.000	0.000	1.000	0.000	1.000	0.000	1.000
90	1	0.000	1.000	0.004	0.996	0.028	0.972	0.011	0.989
91	1	0.600	0.400	0.012	0.988	0.000	1.000	0.313	0.687
92	0	0.800	0.200	0.967	0.033	0.987	0.013	0.888	0.112
93	0	1.000	0.000	0.979	0.021	0.987	0.013	0.993	0.007
94	0	1.000	0.000	0.977	0.023	0.987	0.013	0.993	0.007
95	0	1.000	0.000	0.996	0.004	0.987	0.013	0.995	0.005
96	1	0.400	0.600	0.019	0.981	0.000	1.000	0.209	0.791
97	1	0.200	0.800	0.037	0.963	0.000	1.000	0.107	0.893
98	0	1.000	0.000	0.971	0.029	0.987	0.013	0.992	0.008
99	0	1.000	0.000	0.981	0.019	0.987	0.013	0.993	0.007
100	0	1.000	0.000	0.972	0.028	0.987	0.013	0.992	0.008
101	1	0.200	0.800	0.106	0.894	0.000	1.000	0.112	0.888
102	0	1.000	0.000	0.983	0.017	0.987	0.013	0.993	0.007
103	0	1.000	0.000	0.980	0.020	0.987	0.013	0.993	0.007
104	1	0.000	1.000	0.000	1.000	0.000	1.000	0.000	1.000
105	1	0.400	0.600	0.017	0.983	0.000	1.000	0.209	0.791
106	0	1.000	0.000	0.481	0.519	0.625	0.375	0.805	0.195
107	0	1.000	0.000	0.991	0.009	0.987	0.013	0.994	0.006
108	0	1.000	0.000	0.942	0.058	0.987	0.013	0.990	0.010
109	0	1.000	0.000	0.984	0.016	0.987	0.013	0.993	0.007
110	0	1.000	0.000	0.968	0.032	0.987	0.013	0.992	0.008
111	0	1.000	0.000	0.993	0.007	0.987	0.013	0.994	0.006
112	0	1.000	0.000	0.995	0.005	0.987	0.013	0.994	0.006
113	0	1.000	0.000	0.956	0.044	0.987	0.013	0.991	0.009
114	0	0.600	0.400	0.952	0.048	0.987	0.013	0.782	0.218
115	1	0.600	0.400	0.000	1.000	0.000	1.000	0.312	0.688
116	1	0.400	0.600	0.023	0.977	0.028	0.972	0.221	0.779
117	0	1.000	0.000	0.998	0.002	0.987	0.013	0.995	0.005
118	0	1.000	0.000	0.971	0.029	0.987	0.013	0.992	0.008
119	0	1.000	0.000	0.973	0.027	0.987	0.013	0.992	0.008
120	0	1.000	0.000	0.983	0.017	0.987	0.013	0.993	0.007
121	0	0.800	0.200	0.991	0.009	0.987	0.013	0.890	0.110
122	0	0.600	0.400	0.940	0.060	0.987	0.013	0.781	0.219
123	0	0.800	0.200	0.988	0.012	0.987	0.013	0.890	0.110
124	1	0.800	0.200	0.054	0.946	0.000	1.000	0.420	0.580
125	0	0.600	0.400	0.937	0.063	0.987	0.013	0.781	0.219
126	0	1.000	0.000	0.964	0.036	0.987	0.013	0.991	0.009
127	1	0.000	1.000	0.010	0.990	0.143	0.857	0.057	0.943
128	0	1.000	0.000	0.981	0.019	0.987	0.013	0.993	0.007
129	0	1.000	0.000	0.995	0.005	0.987	0.013	0.995	0.005

130	0	1.000	0.000	0.989	0.011	0.987	0.013	0.994	0.006
131	1	0.400	0.600	0.032	0.968	0.028	0.972	0.222	0.778
132	0	1.000	0.000	0.983	0.017	0.987	0.013	0.993	0.007
133	0	1.000	0.000	0.987	0.013	0.987	0.013	0.994	0.006
134	1	0.000	1.000	0.068	0.932	0.000	1.000	0.005	0.995
135	0	0.600	0.400	0.291	0.709	0.625	0.375	0.581	0.419
136	0	0.800	0.200	0.945	0.055	0.987	0.013	0.886	0.114
137	0	1.000	0.000	0.941	0.059	0.987	0.013	0.989	0.011
138	1	0.600	0.400	0.026	0.974	0.000	1.000	0.314	0.686
139	1	0.400	0.600	0.000	1.000	0.000	1.000	0.208	0.792
140	0	1.000	0.000	0.949	0.051	0.987	0.013	0.990	0.010
141	0	0.800	0.200	0.992	0.008	0.987	0.013	0.890	0.110
142	0	1.000	0.000	0.998	0.002	0.987	0.013	0.995	0.005
143	1	0.400	0.600	0.007	0.993	0.000	1.000	0.209	0.791
144	0	1.000	0.000	0.984	0.016	0.987	0.013	0.993	0.007
145	0	1.000	0.000	0.976	0.024	0.987	0.013	0.993	0.007
146	0	1.000	0.000	0.998	0.002	0.987	0.013	0.995	0.005
147	1	0.600	0.400	0.000	1.000	0.000	1.000	0.312	0.688
148	0	0.800	0.200	0.973	0.027	0.987	0.013	0.888	0.112
149	0	0.800	0.200	0.992	0.008	0.987	0.013	0.890	0.110
150	0	1.000	0.000	0.973	0.027	0.987	0.013	0.992	0.008
151	0	1.000	0.000	0.928	0.072	0.987	0.013	0.988	0.012
152	1	0.000	1.000	0.815	0.185	0.000	1.000	0.070	0.930
153	0	1.000	0.000	0.968	0.032	0.987	0.013	0.992	0.008
154	s0	1.000	0.000	0.985	0.015	0.987	0.013	0.994	0.006
155	0	1.000	0.000	0.976	0.024	0.987	0.013	0.993	0.007
156	1	0.000	1.000	0.874	0.126	0.987	0.013	0.463	0.537
157	0	0.800	0.200	0.998	0.002	0.987	0.013	0.891	0.109
158	1	0.000	1.000	0.008	0.992	0.000	1.000	0.001	0.999
159	0	1.000	0.000	0.993	0.007	0.987	0.013	0.994	0.006
160	0	0.800	0.200	0.963	0.037	0.987	0.013	0.887	0.113
161	0	0.600	0.400	0.942	0.058	0.987	0.013	0.782	0.218
162	0	1.000	0.000	0.996	0.004	0.987	0.013	0.995	0.005
163	0	1.000	0.000	0.996	0.004	0.987	0.013	0.995	0.005
164	0	1.000	0.000	0.967	0.033	0.987	0.013	0.992	0.008
165	0	1.000	0.000	0.956	0.044	0.987	0.013	0.991	0.009
166	0	0.800	0.200	0.989	0.011	0.987	0.013	0.890	0.110
167	0	1.000	0.000	0.995	0.005	0.987	0.013	0.995	0.005
168	0	1.000	0.000	0.980	0.020	0.987	0.013	0.993	0.007
169	0	1.000	0.000	0.973	0.027	0.987	0.013	0.992	0.008
170	0	0.600	0.400	0.975	0.025	0.987	0.013	0.785	0.215
171	1	0.200	0.800	0.073	0.927	0.000	1.000	0.110	0.890
172	0	1.000	0.000	0.992	0.008	0.987	0.013	0.994	0.006
173	1	0.000	1.000	0.009	0.991	0.000	1.000	0.001	0.999
174	0	1.000	0.000	0.972	0.028	0.987	0.013	0.992	0.008

Lampiran 2. Hasil prediksi model *K-Nearest Neighbor*, *Support Vector Machine*, *Decision Tree*, dan *Soft Voting*.

Index	Aktual	Prediksi			
		K-NN	SVM	Decision Tree	Soft Voting
0	Berat	Ringan	Berat	Berat	Berat
1	Berat	Berat	Berat	Berat	Berat
2	Ringan	Ringan	Ringan	Ringan	Ringan
3	Berat	Ringan	Berat	Berat	Berat
4	Berat	Berat	Berat	Berat	Berat
5	Ringan	Ringan	Ringan	Ringan	Ringan
6	Ringan	Ringan	Ringan	Ringan	Ringan
7	Berat	Berat	Berat	Berat	Berat
8	Berat	Berat	Berat	Berat	Berat
9	Ringan	Ringan	Ringan	Ringan	Ringan
10	Berat	Berat	Berat	Berat	Berat
11	Ringan	Ringan	Ringan	Ringan	Ringan
12	Berat	Berat	Berat	Berat	Berat
13	Ringan	Ringan	Ringan	Ringan	Ringan
14	Berat	Berat	Berat	Berat	Berat
15	Berat	Berat	Ringan	Ringan	Berat
16	Ringan	Ringan	Ringan	Ringan	Ringan
17	Ringan	Ringan	Ringan	Ringan	Ringan
18	Ringan	Ringan	Berat	Ringan	Ringan
19	Berat	Berat	Berat	Berat	Berat
20	Ringan	Ringan	Ringan	Ringan	Ringan
21	Berat	Berat	Berat	Berat	Berat
22	Ringan	Ringan	Ringan	Ringan	Ringan
23	Berat	Berat	Berat	Berat	Berat
24	Ringan	Ringan	Ringan	Ringan	Ringan
25	Berat	Berat	Berat	Berat	Berat
26	Ringan	Ringan	Ringan	Ringan	Ringan
27	Berat	Berat	Berat	Berat	Berat
28	Berat	Ringan	Berat	Berat	Berat
29	Ringan	Ringan	Ringan	Ringan	Ringan
30	Ringan	Ringan	Ringan	Ringan	Ringan
31	Ringan	Berat	Ringan	Ringan	Ringan
32	Ringan	Ringan	Berat	Ringan	Ringan
33	Ringan	Ringan	Ringan	Ringan	Ringan
34	Ringan	Ringan	Ringan	Ringan	Ringan
35	Ringan	Ringan	Ringan	Ringan	Ringan
36	Ringan	Ringan	Ringan	Ringan	Ringan
37	Ringan	Ringan	Ringan	Ringan	Ringan
38	Ringan	Ringan	Ringan	Ringan	Ringan
39	Ringan	Ringan	Ringan	Ringan	Ringan

40	Ringan	Ringan	Ringan	Ringan	Ringan
41	Ringan	Ringan	Ringan	Ringan	Ringan
42	Ringan	Ringan	Ringan	Ringan	Ringan
43	Berat	Berat	Berat	Berat	Berat
44	Ringan	Ringan	Ringan	Ringan	Ringan
45	Ringan	Berat	Ringan	Ringan	Ringan
46	Ringan	Ringan	Ringan	Ringan	Ringan
47	Berat	Berat	Berat	Berat	Berat
48	Berat	Berat	Berat	Berat	Berat
49	Berat	Ringan	Berat	Berat	Berat
50	Ringan	Ringan	Ringan	Ringan	Ringan
51	Berat	Berat	Berat	Berat	Berat
52	Ringan	Ringan	Ringan	Ringan	Ringan
53	Ringan	Ringan	Ringan	Ringan	Ringan
54	Ringan	Ringan	Ringan	Ringan	Ringan
55	Ringan	Ringan	Berat	Berat	Berat
56	Berat	Berat	Berat	Berat	Berat
57	Ringan	Ringan	Ringan	Ringan	Ringan
58	Berat	Berat	Berat	Berat	Berat
59	Ringan	Ringan	Ringan	Ringan	Ringan
60	Ringan	Ringan	Ringan	Ringan	Ringan
61	Ringan	Ringan	Ringan	Ringan	Ringan
62	Ringan	Ringan	Ringan	Ringan	Ringan
63	Ringan	Ringan	Ringan	Ringan	Ringan
64	Berat	Berat	Berat	Berat	Berat
65	Ringan	Ringan	Ringan	Ringan	Ringan
66	Berat	Berat	Berat	Berat	Berat
67	Ringan	Ringan	Ringan	Ringan	Ringan
68	Ringan	Ringan	Berat	Berat	Berat
69	Berat	Berat	Berat	Berat	Berat
70	Berat	Berat	Berat	Berat	Berat
71	Ringan	Ringan	Ringan	Ringan	Ringan
72	Ringan	Ringan	Ringan	Ringan	Ringan
73	Berat	Berat	Berat	Berat	Berat
74	Ringan	Ringan	Ringan	Ringan	Ringan
75	Ringan	Ringan	Ringan	Ringan	Ringan
76	Ringan	Ringan	Ringan	Ringan	Ringan
77	Ringan	Ringan	Ringan	Ringan	Ringan
78	Ringan	Ringan	Ringan	Ringan	Ringan
79	Berat	Berat	Berat	Berat	Berat
80	Ringan	Ringan	Ringan	Ringan	Ringan
81	Ringan	Berat	Ringan	Ringan	Ringan
82	Berat	Berat	Berat	Berat	Berat
83	Berat	Ringan	Ringan	Ringan	Ringan
84	Ringan	Ringan	Ringan	Ringan	Ringan

85	Ringan	Ringan	Ringan	Ringan	Ringan
86	Berat	Berat	Berat	Berat	Berat
87	Ringan	Ringan	Ringan	Ringan	Ringan
88	Berat	Berat	Berat	Berat	Berat
89	Berat	Berat	Berat	Berat	Berat
90	Berat	Berat	Berat	Berat	Berat
91	Berat	Ringan	Berat	Berat	Berat
92	Ringan	Ringan	Ringan	Ringan	Ringan
93	Ringan	Ringan	Ringan	Ringan	Ringan
94	Ringan	Ringan	Ringan	Ringan	Ringan
95	Ringan	Ringan	Ringan	Ringan	Ringan
96	Berat	Berat	Berat	Berat	Berat
97	Berat	Berat	Berat	Berat	Berat
98	Ringan	Ringan	Ringan	Ringan	Ringan
99	Ringan	Ringan	Ringan	Ringan	Ringan
100	Ringan	Ringan	Ringan	Ringan	Ringan
101	Berat	Berat	Berat	Berat	Berat
102	Ringan	Ringan	Ringan	Ringan	Ringan
103	Ringan	Ringan	Ringan	Ringan	Ringan
104	Berat	Berat	Berat	Berat	Berat
105	Berat	Berat	Berat	Berat	Berat
106	Ringan	Ringan	Berat	Ringan	Ringan
107	Ringan	Ringan	Ringan	Ringan	Ringan
108	Ringan	Ringan	Ringan	Ringan	Ringan
109	Ringan	Ringan	Ringan	Ringan	Ringan
110	Ringan	Ringan	Ringan	Ringan	Ringan
111	Ringan	Ringan	Ringan	Ringan	Ringan
112	Ringan	Ringan	Ringan	Ringan	Ringan
113	Ringan	Ringan	Ringan	Ringan	Ringan
114	Ringan	Ringan	Ringan	Ringan	Ringan
115	Berat	Ringan	Berat	Berat	Berat
116	Berat	Berat	Berat	Berat	Berat
117	Ringan	Ringan	Ringan	Ringan	Ringan
118	Ringan	Ringan	Ringan	Ringan	Ringan
119	Ringan	Ringan	Ringan	Ringan	Ringan
120	Ringan	Ringan	Ringan	Ringan	Ringan
121	Ringan	Ringan	Ringan	Ringan	Ringan
122	Ringan	Ringan	Ringan	Ringan	Ringan
123	Ringan	Ringan	Ringan	Ringan	Ringan
124	Berat	Ringan	Berat	Berat	Berat
125	Ringan	Ringan	Ringan	Ringan	Ringan
126	Ringan	Ringan	Ringan	Ringan	Ringan
127	Berat	Berat	Berat	Berat	Berat
128	Ringan	Ringan	Ringan	Ringan	Ringan
129	Ringan	Ringan	Ringan	Ringan	Ringan

130	Ringan	Ringan	Ringan	Ringan	Ringan
131	Berat	Berat	Berat	Berat	Berat
132	Ringan	Ringan	Ringan	Ringan	Ringan
133	Ringan	Ringan	Ringan	Ringan	Ringan
134	Berat	Berat	Berat	Berat	Berat
135	Ringan	Ringan	Berat	Ringan	Ringan
136	Ringan	Ringan	Ringan	Ringan	Ringan
137	Ringan	Ringan	Ringan	Ringan	Ringan
138	Berat	Ringan	Berat	Berat	Berat
139	Berat	Berat	Berat	Berat	Berat
140	Ringan	Ringan	Ringan	Ringan	Ringan
141	Ringan	Ringan	Ringan	Ringan	Ringan
142	Ringan	Ringan	Ringan	Ringan	Ringan
143	Berat	Berat	Berat	Berat	Berat
144	Ringan	Ringan	Ringan	Ringan	Ringan
145	Ringan	Ringan	Ringan	Ringan	Ringan
146	Ringan	Ringan	Ringan	Ringan	Ringan
147	Berat	Ringan	Berat	Berat	Berat
148	Ringan	Ringan	Ringan	Ringan	Ringan
149	Ringan	Ringan	Ringan	Ringan	Ringan
150	Ringan	Ringan	Ringan	Ringan	Ringan
151	Ringan	Ringan	Ringan	Ringan	Ringan
152	Berat	Berat	Ringan	Berat	Berat
153	Ringan	Ringan	Ringan	Ringan	Ringan
154	Ringan	Ringan	Ringan	Ringan	Ringan
155	Ringan	Ringan	Ringan	Ringan	Ringan
156	Berat	Berat	Ringan	Ringan	Berat
157	Ringan	Ringan	Ringan	Ringan	Ringan
158	Berat	Berat	Berat	Berat	Berat
159	Ringan	Ringan	Ringan	Ringan	Ringan
160	Ringan	Ringan	Ringan	Ringan	Ringan
161	Ringan	Ringan	Ringan	Ringan	Ringan
162	Ringan	Ringan	Ringan	Ringan	Ringan
163	Ringan	Ringan	Ringan	Ringan	Ringan
164	Ringan	Ringan	Ringan	Ringan	Ringan
165	Ringan	Ringan	Ringan	Ringan	Ringan
166	Ringan	Ringan	Ringan	Ringan	Ringan
167	Ringan	Ringan	Ringan	Ringan	Ringan
168	Ringan	Ringan	Ringan	Ringan	Ringan
169	Ringan	Ringan	Ringan	Ringan	Ringan
170	Ringan	Ringan	Ringan	Ringan	Ringan
171	Berat	Berat	Berat	Berat	Berat
172	Ringan	Ringan	Ringan	Ringan	Ringan
173	Berat	Berat	Berat	Berat	Berat
174	Ringan	Ringan	Ringan	Ringan	Ringan

Lampiran 3. *Pseudocode K-Nearest Neighbor*

```
#Plotting Accuracy for Different Values of k
k = []
FOR i = 1 TO 10:
    Train KNN with n_neighbors=i
    Predict on x_test
    Compute accuracy, append to k

Plot k, Show plot

#Training KNN Classifier with k=5
Start timer
Train KNN with n_neighbors=5
Stop timer, Print time

#Making Predictions and Evaluating the Model
Start timer
Predict on x_test, Print accuracy
Stop timer, Print time

Compute predicted probabilities

Print formatted accuracy
Print classification report
Compute and Display confusion matrix
```

Lampiran 4. *Pseudocode Support Vector Machine*

```
#Testing SVM with Different Kernels
kernel = ['linear', 'rbf', 'poly', 'sigmoid']
FOR i IN kernel:
    Train SVM with kernel=i, C=1, gamma=0.0001, degree=3
    Print kernel
    Print accuracy on x_test and y_test

#Training SVM Classifier with Linear Kernel
Start timer
Train SVM with kernel='linear', C=1, gamma=0.0001, degree=3,
probability=True
Stop timer, Print time

#Making Predictions and Evaluating the Model
Start timer
Predict on x_test, Print accuracy
Stop timer, Print time

Compute predicted probabilities

Print formatted accuracy
Print classification report
Compute and Display confusion matrix
```

Lampiran 5. *Pseudocode Decision Tree*

```
#Testing Decision Tree with Different Max Depths
```

```
max_depth = []
```

```
FOR i = 1 TO 8:
```

```
    Train DecisionTree with max_depth=i
```

```
    Print max_depth
```

```
    Print accuracy on x_test and y_test
```

```
#Training SVM Classifier with Max Depth = 3
```

```
Start timer
```

```
Train DecisionTree with max_depth=3
```

```
Stop timer, Print time
```

```
#Making Predictions and Evaluating the Model
```

```
Start timer
```

```
Predict on x_test, Print accuracy
```

```
Stop timer, Print time
```

```
Compute predicted probabilities
```

```
Print formatted accuracy
```

```
Print classification report
```

```
Compute and Display confusion matrix
```

Lampiran 6. *Pseudocode Soft Voting*

```

#Optimizing Ensemble Model Using PSO
FUNCTION objective_function(weights, x_train, y_train, x_test, y_test):
    Normalize weights
    Compute ensemble using weights and predictions from clf1, clf2, clf3 on
    x_train
    Compute y_pred using argmax on ensemble
    Compute and return negative accuracy on y_train and y_pred

Set options for PSO
Initialize PSO optimizer with 50 particles, 3 dimensions, and options
Optimize using objective_function, x_train, y_train, x_test, y_test, for 100
iterations
Normalize best_weights
Print best_weights
Compute ensemble using best_weights and predictions from clf1, clf2, clf3 on
x_test
Compute y_pred using argmax on ensemble
Print Soft Voting accuracy with best_weights

#Training Soft Voting Classifier
Start timer
Set eclf1, eclf2, eclf3 to clf1, clf2, clf3
Define estimators with ('K-Nearest Neighbor', eclf1), ('Support Vector
Machine', eclf2), ('Decision Tree', eclf3)
Initialize VotingClassifier with estimators, voting='soft', and specific weights
Fit on x_train and y_train
Stop timer, Print time

```

```
#Making Predictions and Evaluating Soft Voting Classifier
```

```
Start timer
```

```
Predict on x_test, Print Soft Voting Classifier accuracy
```

```
Stop timer, Print time
```

```
Compute predicted probabilities
```

```
Print formatted Soft Voting Classifier accuracy
```

```
Print classification report
```

```
Compute and Display confusion matrix
```

Lampiran 7. Contoh Kasus Sederhana *K-Nearest Neighbor*, *Support Vector Machine*, *Decision Tree*, dan *Soft Voting Classifier*.

Frekuensi Napas	Leukosit	Trombosit	Penggunaan Vasopressor	Ventilasi Mekanis	Pneumonia
1	0	0	0	0	Ringan
1	1	1	0	1	Berat
0	1	0	0	0	Ringan
0	0	1	0	0	Ringan
1	1	0	0	0	Ringan
1	1	1	1	0	Berat
1	1	1	1	1	Berat
0	0	0	0	0	Ringan
1	0	1	0	1	Berat
1	0	1	0	0	Ringan
1	0	0	0	1	?

a. *K-Nearest Neighbor*

- 1) Tentukan nilai K, misalnya nilai K yang dipilih adalah K=3
- 2) Hitung *similarity/distance* menggunakan *Manhattan Distance*, dengan rumus dibawah ini:

$$d(x, y) = \sum_{i=1}^m |x_i - y_i|$$

Dimana x_i adalah data yang atributnya sudah dieksekusi dan y_i adalah data uji baru selain data latih.

- Data 1
 $distance = (1-1) + (0-0) + (0-0) + (0-0) + (0-1) = 0 + 0 + 0 + 0 + 1 = 1$
- Data 2
 $distance = (1-1) + (1-0) + (1-0) + (0-0) + (1-1) = 0 + 1 + 1 + 0 + 0 = 2$
- Data 3
 $distance = (0-1) + (1-0) + (0-0) + (0-0) + (0-1) = 1 + 1 + 0 + 0 + 1 = 3$
- Data 4
 $distance = (0-1) + (0-0) + (1-0) + (0-0) + (0-1) = 1 + 0 + 1 + 0 + 1 = 3$
- Data 5
 $distance = (1-1) + (1-0) + (0-0) + (0-0) + (0-1) = 0 + 1 + 0 + 0 + 1 = 2$
- Data 6
 $distance = (1-1) + (1-0) + (1-0) + (1-0) + (0-1) = 0 + 1 + 1 + 1 + 1 = 4$
- Data 7
 $distance = (1-1) + (1-0) + (1-0) + (1-0) + (1-1) = 0 + 1 + 1 + 1 + 0 = 3$
- Data 8
 $distance = (0-1) + (0-0) + (0-0) + (0-0) + (0-1) = 1 + 0 + 0 + 0 + 1 = 2$
- Data 9
 $distance = (1-1) + (0-0) + (1-0) + (0-0) + (1-1) = 0 + 0 + 1 + 0 + 0 = 1$
- Data 10
 $distance = (1-1) + (0-0) + (1-0) + (0-0) + (0-1) = 0 + 0 + 1 + 0 + 1 = 2$

Berdasarkan hasil perhitungan *distance* diatas, maka didapatkan hasil sebagai berikut.

Data	Distance
D1, D11	1
D2, D11	2
D3, D11	3
D4, D11	3
D5, D11	2
D6, D11	4
D7, D11	3
D8, D11	2
D9, D11	1
D10, D11	2

3) Urutkan data berdasarkan *similarity/distance*.

Data	Distance	Ranking
D1, D11	1	1
D2, D11	2	3
D3, D11	3	7
D4, D11	3	8
D5, D11	2	4
D6, D11	4	10
D7, D11	3	9
D8, D11	2	5
D9, D11	1	2
D10, D11	2	6

4) Tentukan K klasifikasi (K terbaik berdasarkan ranking), dengan nilai $K = 3$, maka didapatkan hasil sebagai berikut.

Data	Distance	Ranking
D1, D11	1	1
D9, D11	1	2
D2, D11	2	3

5) Tentukan klasifikasi menggunakan kategori mayoritas.

Doc	Distance	Ranking	Pneumonia
D1, D11	1	1	Ringan
D9, D11	1	2	Berat
D2, D11	2	3	Berat

Karena label mayoritas adalah “Berat”, maka pada data 11 dilabelkan ke dalam pneumonia “Berat”.

6) Probabilitas

- Jarak terdekat dengan $K=3$
 Kelas Ringan = 1 (D1)
 Kelas Berat = 2 (D9 dan D2)
- $P(\text{Ringan}|\text{D11}) = \text{Jumlah D11 yang berkelas Ringan}/K = 1/3 = 0.33$
 $P(\text{Berat}|\text{D11}) = \text{Jumlah D11 yang berkelas Berat}/K = 2/3 = 0.67$

Jadi, berdasarkan perhitungan probabilitas, data baru (D11) diklasifikasikan ke dalam kelas Berat.

b. *Support Vector Machine*

- 1) Menerapkan fungsi *kernel* dengan menggunakan *kernel* linear.

$$K(x_i, x_j) = x_i \cdot x_j$$

- 2) Hitung *dot product* antara setiap pasangan data dalam *dataset* untuk mendapatkan matriks Gram. Adapun perhitungan dari *dot product* antara setiap pasangan data dalam *dataset* adalah sebagai berikut.

- K Data 1:

$$\begin{aligned} K(D1*D1) &= [1, 0, 0, 0, 0] * [1, 0, 0, 0, 0] = 1 + 0 + 0 + 0 + 0 = 1 \\ K(D1*D2) &= [1, 0, 0, 0, 0] * [1, 1, 1, 0, 1] = 1 + 0 + 0 + 0 + 0 = 1 \\ K(D1*D3) &= [1, 0, 0, 0, 0] * [0, 1, 0, 0, 0] = 0 + 0 + 0 + 0 + 0 = 0 \\ K(D1*D4) &= [1, 0, 0, 0, 0] * [0, 0, 1, 0, 0] = 0 + 0 + 0 + 0 + 0 = 0 \\ K(D1*D5) &= [1, 0, 0, 0, 0] * [1, 1, 0, 0, 0] = 1 + 0 + 0 + 0 + 0 = 1 \\ K(D1*D6) &= [1, 0, 0, 0, 0] * [1, 1, 1, 1, 0] = 1 + 0 + 0 + 0 + 0 = 1 \\ K(D1*D7) &= [1, 0, 0, 0, 0] * [1, 1, 1, 1, 1] = 1 + 0 + 0 + 0 + 0 = 1 \\ K(D1*D8) &= [1, 0, 0, 0, 0] * [0, 0, 0, 0, 0] = 0 + 0 + 0 + 0 + 0 = 0 \\ K(D1*D9) &= [1, 0, 0, 0, 0] * [1, 0, 1, 0, 1] = 1 + 0 + 0 + 0 + 0 = 1 \\ K(D1*D10) &= [1, 0, 0, 0, 0] * [1, 0, 1, 0, 0] = 1 + 0 + 0 + 0 + 0 = 1 \end{aligned}$$

- K Data 2:

$$\begin{aligned} K(D2*D1) &= [1, 1, 1, 0, 1] * [1, 0, 0, 0, 0] = 1 + 0 + 0 + 0 + 0 = 1 \\ K(D2*D2) &= [1, 1, 1, 0, 1] * [1, 1, 1, 0, 1] = 1 + 1 + 1 + 0 + 1 = 4 \\ K(D2*D3) &= [1, 1, 1, 0, 1] * [0, 1, 0, 0, 0] = 0 + 1 + 0 + 0 + 0 = 1 \\ K(D2*D4) &= [1, 1, 1, 0, 1] * [0, 0, 1, 0, 0] = 0 + 0 + 1 + 0 + 0 = 1 \\ K(D2*D5) &= [1, 1, 1, 0, 1] * [1, 1, 0, 0, 0] = 1 + 1 + 0 + 0 + 0 = 2 \\ K(D2*D6) &= [1, 1, 1, 0, 1] * [1, 1, 1, 1, 0] = 1 + 1 + 1 + 0 + 0 = 3 \\ K(D2*D7) &= [1, 1, 1, 0, 1] * [1, 1, 1, 1, 1] = 1 + 1 + 1 + 0 + 1 = 4 \\ K(D2*D8) &= [1, 1, 1, 0, 1] * [0, 0, 0, 0, 0] = 0 + 0 + 0 + 0 + 0 = 0 \\ K(D2*D9) &= [1, 1, 1, 0, 1] * [1, 0, 1, 0, 1] = 1 + 0 + 1 + 0 + 1 = 3 \\ K(D2*D10) &= [1, 1, 1, 0, 1] * [1, 0, 1, 0, 0] = 1 + 0 + 1 + 0 + 0 = 2 \end{aligned}$$

- K Data 3:

$$\begin{aligned} K(D3*D1) &= [0, 1, 0, 0, 0] * [1, 0, 0, 0, 0] = 0 + 0 + 0 + 0 + 0 = 0 \\ K(D3*D2) &= [0, 1, 0, 0, 0] * [1, 1, 1, 0, 1] = 0 + 1 + 0 + 0 + 0 = 1 \\ K(D3*D3) &= [0, 1, 0, 0, 0] * [0, 1, 0, 0, 0] = 0 + 1 + 0 + 0 + 0 = 1 \\ K(D3*D4) &= [0, 1, 0, 0, 0] * [0, 0, 1, 0, 0] = 0 + 0 + 0 + 0 + 0 = 0 \\ K(D3*D5) &= [0, 1, 0, 0, 0] * [1, 1, 0, 0, 0] = 0 + 1 + 0 + 0 + 0 = 1 \\ K(D3*D6) &= [0, 1, 0, 0, 0] * [1, 1, 1, 1, 0] = 0 + 1 + 0 + 0 + 0 = 1 \\ K(D3*D7) &= [0, 1, 0, 0, 0] * [1, 1, 1, 1, 1] = 0 + 1 + 0 + 0 + 0 = 1 \\ K(D3*D8) &= [0, 1, 0, 0, 0] * [0, 0, 0, 0, 0] = 0 + 0 + 0 + 0 + 0 = 0 \end{aligned}$$

$$K(D3*D9) = [0, 1, 0, 0, 0] * [1, 0, 1, 0, 1] = 0 + 0 + 0 + 0 + 0 = 0$$

$$K(D3*D10) = [0, 1, 0, 0, 0] * [1, 0, 1, 0, 0] = 0 + 0 + 0 + 0 + 0 = 0$$

- K Data 4:

$$K(D4*D1) = [0, 0, 1, 0, 0] * [1, 0, 0, 0, 0] = 0 + 0 + 0 + 0 + 0 = 0$$

$$K(D4*D2) = [0, 0, 1, 0, 0] * [1, 1, 1, 0, 1] = 0 + 0 + 1 + 0 + 0 = 1$$

$$K(D4*D3) = [0, 0, 1, 0, 0] * [0, 1, 0, 0, 0] = 0 + 0 + 0 + 0 + 0 = 0$$

$$K(D4*D4) = [0, 0, 1, 0, 0] * [0, 0, 1, 0, 0] = 0 + 0 + 1 + 0 + 0 = 1$$

$$K(D4*D5) = [0, 0, 1, 0, 0] * [1, 1, 0, 0, 0] = 0 + 0 + 0 + 0 + 0 = 0$$

$$K(D4*D6) = [0, 0, 1, 0, 0] * [1, 1, 1, 1, 0] = 0 + 0 + 1 + 0 + 0 = 1$$

$$K(D4*D7) = [0, 0, 1, 0, 0] * [1, 1, 1, 1, 1] = 0 + 0 + 1 + 0 + 0 = 1$$

$$K(D4*D8) = [0, 0, 1, 0, 0] * [0, 0, 0, 0, 0] = 0 + 0 + 0 + 0 + 0 = 0$$

$$K(D4*D9) = [0, 0, 1, 0, 0] * [1, 0, 1, 0, 1] = 0 + 0 + 1 + 0 + 0 = 1$$

$$K(D4*D10) = [0, 0, 1, 0, 0] * [1, 0, 1, 0, 0] = 0 + 0 + 1 + 0 + 0 = 1$$

- K Data 5:

$$K(D5*D1) = [1, 1, 0, 0, 0] * [1, 0, 0, 0, 0] = 1 + 0 + 0 + 0 + 0 = 1$$

$$K(D5*D2) = [1, 1, 0, 0, 0] * [1, 1, 1, 0, 1] = 1 + 1 + 0 + 0 + 0 = 2$$

$$K(D5*D3) = [1, 1, 0, 0, 0] * [0, 1, 0, 0, 0] = 0 + 1 + 0 + 0 + 0 = 1$$

$$K(D5*D4) = [1, 1, 0, 0, 0] * [0, 0, 1, 0, 0] = 0 + 0 + 0 + 0 + 0 = 0$$

$$K(D5*D5) = [1, 1, 0, 0, 0] * [1, 1, 0, 0, 0] = 1 + 1 + 0 + 0 + 0 = 2$$

$$K(D5*D6) = [1, 1, 0, 0, 0] * [1, 1, 1, 1, 0] = 1 + 1 + 0 + 0 + 0 = 2$$

$$K(D5*D7) = [1, 1, 0, 0, 0] * [1, 1, 1, 1, 1] = 1 + 1 + 0 + 0 + 0 = 2$$

$$K(D5*D8) = [1, 1, 0, 0, 0] * [0, 0, 0, 0, 0] = 0 + 0 + 0 + 0 + 0 = 0$$

$$K(D5*D9) = [1, 1, 0, 0, 0] * [1, 0, 1, 0, 1] = 1 + 0 + 0 + 0 + 0 = 1$$

$$K(D5*D10) = [1, 1, 0, 0, 0] * [1, 0, 1, 0, 0] = 1 + 0 + 0 + 0 + 0 = 1$$

- K Data 6:

$$K(D6*D1) = [1, 1, 1, 1, 0] * [1, 0, 0, 0, 0] = 1 + 0 + 0 + 0 + 0 = 1$$

$$K(D6*D2) = [1, 1, 1, 1, 0] * [1, 1, 1, 0, 1] = 1 + 1 + 1 + 0 + 0 = 3$$

$$K(D6*D3) = [1, 1, 1, 1, 0] * [0, 1, 0, 0, 0] = 0 + 1 + 0 + 0 + 0 = 1$$

$$K(D6*D4) = [1, 1, 1, 1, 0] * [0, 0, 1, 0, 0] = 0 + 0 + 1 + 0 + 0 = 1$$

$$K(D6*D5) = [1, 1, 1, 1, 0] * [1, 1, 0, 0, 0] = 1 + 1 + 0 + 0 + 0 = 2$$

$$K(D6*D6) = [1, 1, 1, 1, 0] * [1, 1, 1, 1, 0] = 1 + 1 + 1 + 1 + 0 = 4$$

$$K(D6*D7) = [1, 1, 1, 1, 0] * [1, 1, 1, 1, 1] = 1 + 1 + 1 + 1 + 0 = 4$$

$$K(D6*D8) = [1, 1, 1, 1, 0] * [0, 0, 0, 0, 0] = 0 + 0 + 0 + 0 + 0 = 0$$

$$K(D6*D9) = [1, 1, 1, 1, 0] * [1, 0, 1, 0, 1] = 1 + 0 + 1 + 0 + 0 = 2$$

$$K(D6*D10) = [1, 1, 1, 1, 0] * [1, 0, 1, 0, 0] = 1 + 0 + 1 + 0 + 0 = 2$$

- K Data 7:

$$K(D7*D1) = [1, 1, 1, 1, 1] * [1, 0, 0, 0, 0] = 1 + 0 + 0 + 0 + 0 = 1$$

$$K(D7*D2) = [1, 1, 1, 1, 1] * [1, 1, 1, 0, 1] = 1 + 1 + 1 + 0 + 1 = 4$$

$$K(D7*D3) = [1, 1, 1, 1, 1] * [0, 1, 0, 0, 0] = 0 + 1 + 0 + 0 + 0 = 1$$

$$K(D7*D4) = [1, 1, 1, 1, 1] * [0, 0, 1, 0, 0] = 0 + 0 + 1 + 0 + 0 = 1$$

$$K(D7*D5) = [1, 1, 1, 1, 1] * [1, 1, 0, 0, 0] = 1 + 1 + 0 + 0 + 0 = 2$$

$$K(D7*D6) = [1, 1, 1, 1, 1] * [1, 1, 1, 1, 0] = 1 + 1 + 1 + 1 + 0 = 4$$

$$K(D7*D7) = [1, 1, 1, 1, 1] * [1, 1, 1, 1, 1] = 1 + 1 + 1 + 1 + 1 = 5$$

$$K(D7*D8) = [1, 1, 1, 1, 1] * [0, 0, 0, 0, 0] = 0 + 0 + 0 + 0 + 0 = 0$$

$$K(D7*D9) = [1, 1, 1, 1, 1] * [1, 0, 1, 0, 1] = 1 + 0 + 1 + 0 + 1 = 3$$

$$K(D7*D10) = [1, 1, 1, 1, 1] * [1, 0, 1, 0, 0] = 1 + 0 + 1 + 0 + 0 = 2$$

- K Data 8:

$$K(D8*D1) = [0, 0, 0, 0, 0] * [1, 0, 0, 0, 0] = 0 + 0 + 0 + 0 + 0 = 0$$

$$K(D8*D2) = [0, 0, 0, 0, 0] * [1, 1, 1, 0, 1] = 0 + 0 + 0 + 0 + 0 = 0$$

$$K(D8*D3) = [0, 0, 0, 0, 0] * [0, 1, 0, 0, 0] = 0 + 0 + 0 + 0 + 0 = 0$$

$$K(D8*D4) = [0, 0, 0, 0, 0] * [0, 0, 1, 0, 0] = 0 + 0 + 0 + 0 + 0 = 0$$

$$K(D8*D5) = [0, 0, 0, 0, 0] * [1, 1, 0, 0, 0] = 0 + 0 + 0 + 0 + 0 = 0$$

$$K(D8*D6) = [0, 0, 0, 0, 0] * [1, 1, 1, 1, 0] = 0 + 0 + 0 + 0 + 0 = 0$$

$$K(D8*D7) = [0, 0, 0, 0, 0] * [1, 1, 1, 1, 1] = 0 + 0 + 0 + 0 + 0 = 0$$

$$K(D8*D8) = [0, 0, 0, 0, 0] * [0, 0, 0, 0, 0] = 0 + 0 + 0 + 0 + 0 = 0$$

$$K(D8*D9) = [0, 0, 0, 0, 0] * [1, 0, 1, 0, 1] = 0 + 0 + 0 + 0 + 0 = 0$$

$$K(D8*D10) = [0, 0, 0, 0, 0] * [1, 0, 1, 0, 0] = 0 + 0 + 0 + 0 + 0 = 0$$

- K Data 9:

$$K(D9*D1) = [1, 0, 1, 0, 1] * [1, 0, 0, 0, 0] = 1 + 0 + 0 + 0 + 0 = 1$$

$$K(D9*D2) = [1, 0, 1, 0, 1] * [1, 1, 1, 0, 1] = 1 + 0 + 1 + 0 + 1 = 3$$

$$K(D9*D3) = [1, 0, 1, 0, 1] * [0, 1, 0, 0, 0] = 0 + 0 + 0 + 0 + 0 = 0$$

$$K(D9*D4) = [1, 0, 1, 0, 1] * [0, 0, 1, 0, 0] = 0 + 0 + 1 + 0 + 0 = 1$$

$$K(D9*D5) = [1, 0, 1, 0, 1] * [1, 1, 0, 0, 0] = 1 + 0 + 0 + 0 + 0 = 1$$

$$K(D9*D6) = [1, 0, 1, 0, 1] * [1, 1, 1, 1, 0] = 1 + 0 + 1 + 0 + 0 = 2$$

$$K(D9*D7) = [1, 0, 1, 0, 1] * [1, 1, 1, 1, 1] = 1 + 0 + 1 + 0 + 1 = 3$$

$$K(D9*D8) = [1, 0, 1, 0, 1] * [0, 0, 0, 0, 0] = 0 + 0 + 0 + 0 + 0 = 0$$

$$K(D9*D9) = [1, 0, 1, 0, 1] * [1, 0, 1, 0, 1] = 1 + 0 + 1 + 0 + 1 = 3$$

$$K(D9*D10) = [1, 0, 1, 0, 1] * [1, 0, 1, 0, 0] = 1 + 0 + 1 + 0 + 0 = 2$$

- K Data 10:

$$K(D10*D1) = [1, 0, 1, 0, 0] * [1, 0, 0, 0, 0] = 1 + 0 + 0 + 0 + 0 = 1$$

$$K(D10*D2) = [1, 0, 1, 0, 0] * [1, 1, 1, 0, 1] = 1 + 0 + 1 + 0 + 0 = 2$$

$$K(D10*D3) = [1, 0, 1, 0, 0] * [0, 1, 0, 0, 0] = 0 + 0 + 0 + 0 + 0 = 0$$

$$K(D10*D4) = [1, 0, 1, 0, 0] * [0, 0, 1, 0, 0] = 0 + 0 + 1 + 0 + 0 = 1$$

$$K(D10*D5) = [1, 0, 1, 0, 0] * [1, 1, 0, 0, 0] = 1 + 0 + 0 + 0 + 0 = 1$$

$$K(D10*D6) = [1, 0, 1, 0, 0] * [1, 1, 1, 1, 0] = 1 + 0 + 1 + 0 + 0 = 2$$

$$K(D10*D7) = [1, 0, 1, 0, 0] * [1, 1, 1, 1, 1] = 1 + 0 + 1 + 0 + 0 = 2$$

$$K(D10*D8) = [1, 0, 1, 0, 0] * [0, 0, 0, 0, 0] = 0 + 0 + 0 + 0 + 0 = 0$$

$$K(D10*D9) = [1, 0, 1, 0, 0] * [1, 0, 1, 0, 1] = 1 + 0 + 1 + 0 + 0 = 2$$

$$K(D10*D10) = [1, 0, 1, 0, 0] * [1, 0, 1, 0, 0] = 1 + 0 + 1 + 0 + 0 = 2$$

Dari hasil perhitungan diatas maka didapatkan Matriks Gram sebagai berikut

1	1	0	0	1	1	1	0	1	1
1	4	1	1	2	3	4	0	3	2
0	1	1	0	1	1	1	0	0	0
0	1	0	1	0	1	1	0	1	1
1	2	1	0	2	2	2	0	1	1
1	3	1	1	2	4	4	0	2	2
1	4	1	1	2	4	5	0	3	2
0	0	0	0	0	0	0	0	0	0
1	3	0	1	1	2	3	0	3	2
1	2	0	1	1	2	2	0	2	2

- 3) Pisahkan setiap kelas untuk setiap data pneumonia dalam *dataset*, dan selesaikan masalah optimasi untuk mencari vektor bobot (w) dan nilai bias (b).

Label Kelas : [0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0]

Dalam contoh ini, kita akan menggunakan metode solver optimasi untuk menemukan nilai α_i yang memenuhi batasan. Mari asumsikan kita memiliki nilai α_i yang telah dihitung sebagai berikut.

$$\alpha = [0.0, 0.3, 0.0, 0.0, 0.0, 0.2, 0.1, 0.0, 0.4, 0]$$

Setelah menentukan nilai α , selanjutnya cari vektor bobot (w) dan nilai bias (b) menggunakan rumus sebagai berikut.

$$w = \sum \alpha_i y_i x_i$$

- $w_1 = \alpha[0] * y_0 * x_0$
 $= 0.0 * 0 * [1, 0, 0, 0, 0]$
 $= [0, 0, 0, 0, 0]$
- $w_2 = \alpha[1] * y_1 * x_1$
 $= 0.3 * 1 * [1, 1, 1, 0, 1]$
 $= [0.3, 0.3, 0.3, 0, 0.3]$
- $w_3 = \alpha[2] * y_2 * x_2$
 $= 0.0 * 0 * [0, 1, 0, 0, 0]$
 $= [0, 0, 0, 0, 0]$
- $w_4 = \alpha[3] * y_3 * x_3$
 $= 0.0 * 0 * [0, 0, 1, 0, 0]$
 $= [0, 0, 0, 0, 0]$
- $w_5 = \alpha[4] * y_4 * x_4$
 $= 0.0 * 0 * [1, 1, 0, 0, 0]$
 $= [0, 0, 0, 0, 0]$
- $w_6 = \alpha[5] * y_5 * x_5$
 $= 0.2 * 1 * [1, 1, 1, 1, 0]$
 $= [0.2, 0.2, 0.2, 0.2, 0]$
- $w_7 = \alpha[6] * y_6 * x_6$
 $= 0.1 * 1 * [1, 1, 1, 1, 1]$
 $= [0.1, 0.1, 0.1, 0.1, 0.1]$
- $w_8 = \alpha[7] * y_7 * x_7$
 $= 0.0 * 0 * [0, 0, 0, 0, 0]$
 $= [0, 0, 0, 0, 0]$
- $w_9 = \alpha[8] * y_8 * x_8$
 $= 0.4 * 1 * [1, 0, 1, 0, 1]$
 $= [0.4, 0, 0.4, 0, 0.4]$
- $w_{10} = \alpha[9] * y_9 * x_9$
 $= 0.0 * 0 * [1, 0, 1, 0, 0]$
 $= [0, 0, 0, 0, 0]$
- $w = (w_1 + w_2 + w_3 + w_4 + w_5 + w_6 + w_7 + w_8 + w_9 + w_{10})$
 $= ([0, 0, 0, 0, 0] + [0.3, 0.3, 0.3, 0, 0.3] + [0, 0, 0, 0, 0] + [0, 0, 0, 0, 0]$
 $+ [0, 0, 0, 0, 0] + [0.2, 0.2, 0.2, 0.2, 0] + [0.1, 0.1, 0.1, 0.1, 0.1] + [0, 0,$
 $0, 0, 0] + [0.4, 0, 0.4, 0, 0.4] + [0, 0, 0, 0, 0])$
 $= [1, 0.6, 1, 0.3, 0.8]$

Selanjutnya, menghitung nilai bias dengan rumus sebagai berikut.

$$b = y_j - \sum \alpha_i y_i K(x_i, x_j)$$

Kita akan memilih nilai α_i yang tidak nol [alpha[1], alpha[5], alpha[6], alpha[8]], maka vektor pendukung (j=1)

$$\begin{aligned} - \quad b &= 1 - ((0.3 \cdot 1 \cdot 4) + (0.2 \cdot 1 \cdot 4) + (0.1 \cdot 1 \cdot 5) + (0.4 \cdot 1 \cdot 3)) \\ &= 1 - (1.2 + 0.8 + 0.5 + 1.2) \\ &= 1 - 3.7 \\ &= -2.7 \end{aligned}$$

Sehingga mendapatkan vektor bobot (w) = [1, 0.6, 1, 0.3, 0.8] dan nilai bias (b) = -2.7.

- 4) Menentukan *hyperplane* dengan fungsi keputusan yang akan memisahkan kedua kelas “Ringan” dan “Berat” dengan margin maksimal. Berikut rumus dari fungsi keputusan.

$$f(x_{new}) = w * x_{new} + b$$

Kita memiliki x_{new} : [1, 0, 0, 0, 1]

$$\begin{aligned} f(x_{new}) &= [1, 0.6, 1, 0.3, 0.8] * [1, 0, 0, 0, 1] + -3.713 \\ &= (1 + 0 + 0 + 0 + 0.8) - 2.7 \\ &= 1.8 - 2.7 \\ &= -0.9 \end{aligned}$$

Dari hasil diatas didapatkan nilai $f(x_{new}) = -0.9$, dengan menggunakan fungsi sign dari $f(x)$ dalam menentukan kelas prediksi. Jika $f(x) > 0$, maka kelas prediksi adalah 1, dan jika $f(x) \leq 0$, maka kelas prediksi adalah 0. Jadi dengan hasil tersebut, maka prediksi dari SVM adalah 0 atau dilabelkan ke dalam pneumonia “Ringan”.

- 5) Probabilitas

$$P(y=1|x) = 1 / (1 + \exp(A * f(x) + B)),$$

karena nilai A dan B diperoleh dengan metode seperti cross validation atau platt scaling, jadi kita langsung saja menentukan secara acak dengan misalkan A = 0.4 dan B = 0.6, $f(x_{new}) = -0.9$

$$\begin{aligned} - \quad P(y=1|D11) &= 1 / (1 + \exp(0.4 * -0.9 + 0.6)) \\ &= 1 / (1 + \exp(-0.36 + 0.6)) \\ &= 1 / (1 + \exp(0.24)) \\ &= 1 / (1 + 1.271) \\ &= 1 / 2.271 \\ &= 0.44 \\ P(y=0|D11) &= 1 - P(y=1|D11) \\ &= 1 - 0.44 \\ &= 0.56 \end{aligned}$$

Jadi, berdasarkan perhitungan probabilitas, data baru (D11) diklasifikasikan ke dalam kelas Ringan

c. *Decision Tree*

- 1) Tentukan fitur terbaik untuk split pertama dengan menggunakan Gini *impurity* sebagai metrik. Pertama-tama, pembentukan *root node* dengan menghitung nilai Gini *impurity* untuk setiap fitur dengan rumus sebagai berikut.

$$Gini = 1 - \sum_{i=1}^c P(i|t)^2$$

- F1 = (1, 1, 0, 0, 1, 1, 1, 0, 1, 1)
T = (Ringan, Berat, Ringan, Ringan, Ringan, Berat, Berat, Ringan, Berat, Ringan)
- Impurity(F1=0) = $1 - (P(\text{Kelas Ringan}/F1=0)^2 + P(\text{Kelas Berat}/F1=0)^2)$
= $1 - ((3/3)^2 + (0/3)^2)$
= 0
- Impurity(F1=1) = $1 - (P(\text{Kelas Ringan}/F1=1)^2 + P(\text{Kelas Berat}/F1=1)^2)$
= $1 - ((3/7)^2 + (4/7)^2)$
= 0.49

Impurity Gini setelah split F1 adalah rata-rata tertimbang dari *impurity* kedua kondisi tersebut

$$\begin{aligned} \text{Impurity}(F1) &= (P(F1=0) * \text{Impurity}(F1=0)) + (P(F1=1) * \text{Impurity}(F1=1)) \\ &= (0.3 * 0) + (0.7 * 0.49) \\ &= 0.343 \end{aligned}$$

Setelah itu, lakukan hal yang sama untuk semua fitur lainnya, dan pilih fitur dengan *impurity* terendah setelah *split* sebagai *node* pertama. Lalu ulangi proses ini untuk setiap *sub-node* sampai semua data dalam satu *node* berada dalam kelas yang sama atau sampai kita mencapai batas kedalaman pohon yang telah ditentukan. Berikut hasil dari perhitungan *impurity* gini setiap fitur.

Fitur	Gini Impurity
Frekuensi Napas (F1)	0.343
Leukosit (F2)	0.4
Trombosit (F3)	0.267
Penggunaan Vasopressor (F4)	0.3
Ventilasi Mekanis (F5)	0.171

Dengan hasil *impurity* gini yang didapatkan, maka F5 (Ventilasi Mekanis) akan menjadi fitur pertama (*root node*) dalam pohon keputusan karena memiliki *impurity* Gini terendah 0.171.

- 2) Pembentukan cabang (*branches*)

Setelah menentukan *root node*, kita akan membuat cabang (*branches*) dari *root node* tersebut berdasarkan kondisi yang ditentukan oleh nilai fitur di *root node*. Fitur dengan Gini *impurity* terendah (F5) dalam kasus ini dipilih sebagai *node root* dari pohon. Dengan cara yang sama, proses diulang untuk setiap anak dari *root* berdasarkan *subset* data yang sesuai (misalnya, semua *instance* di mana

F5=0 menjadi satu cabang, dan semua instance di mana F5=1 menjadi cabang lain).

- Dataset F5 = 0

F1	F2	F3	F4	F5	Label
1	0	0	0	0	Ringan
0	1	0	0	0	Ringan
0	0	1	0	0	Ringan
1	1	0	0	0	Ringan
1	1	1	1	0	Berat
0	0	0	0	0	Ringan
1	0	1	0	0	Ringan

- Dataset F5 = 1

F1	F2	F3	F4	F5	Label
1	1	1	0	1	Berat
1	1	1	1	1	Berat
1	0	1	0	1	Berat

Sekarang kita harus menghitung Gini *impurity* untuk setiap fitur di setiap *subset* dan memilih fitur dengan *impurity* terendah untuk memisahkan data lebih lanjut. Dengan rumus Gini *Impurity*, maka didapatkan hasil untuk setiap perhitungan.

Fitur	Gini Impurity F5 = 0	Gini Impurity F5 = 1
Frekuensi Napas (F1)	0.214	0
Leukosit (F2)	0.191	0
Trombosit (F3)	0.191	0
Penggunaan Vasopressor (F4)	0	0

Berdasarkan tabel diatas, pada *subset* F5 = 0 akan dipilih F4 sebagai *node* berikutnya karena memiliki Gini *impurity* yang paling rendah. Sedangkan pada subset F5 = 1 akan berhenti membagi karena semua target pada *subset* nya sama dengan 1, maka Gini *impurity* nya adalah 0 untuk semua fitur.

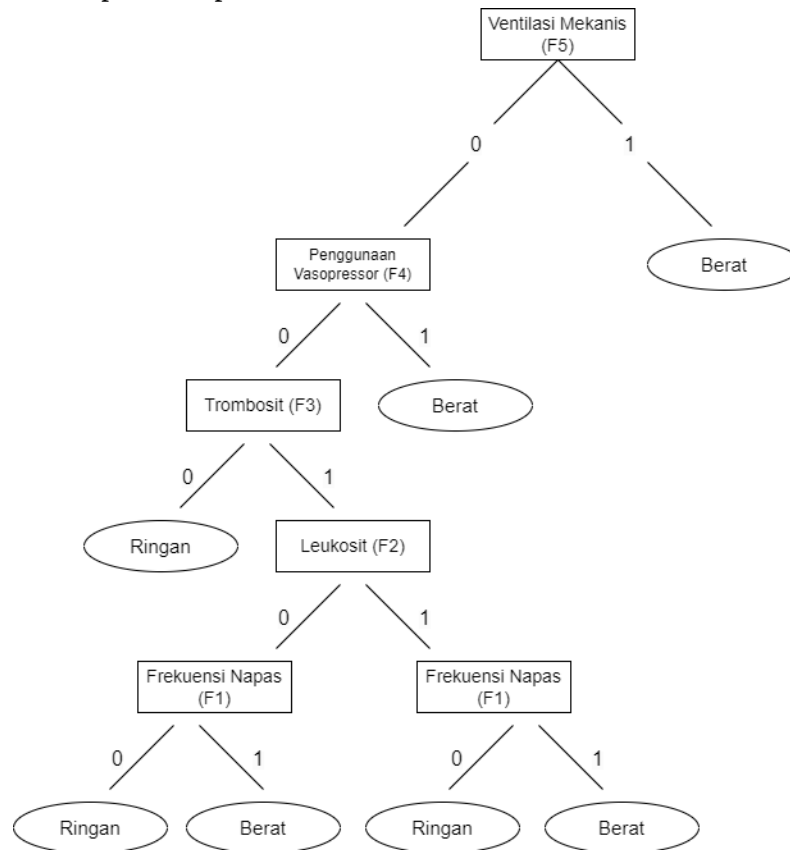
- 3) Melanjutkan ke cabang – cabang berikutnya sampai ke cabang terakhir

Fitur	Gini Impurity F4 = 0	Gini Impurity F4 = 1
Frekuensi Napas (F1)	0.3	0
Leukosit (F2)	0.367	0
Trombosit (F3)	0.25	0

Fitur	Gini Impurity F3 = 0	Gini Impurity F3 = 1
Frekuensi Napas (F1)	0	0.267
Leukosit (F2)	0	0.222

Fitur	Gini Impurity F2 = 0	Gini Impurity F2 = 1
Frekuensi Napas (F1)	0.267	0.3

4) Pembentukan pohon keputusan



5) Menentukan kelas data baru berdasarkan pohon keputusan

Frekuensi Napas (F1)	Leukosit (F2)	Trombosit (F3)	Penggunaan Vasopressor (F4)	Ventilasi Mekanis (F5)	Pneumonia
1	0	0	0	1	?

Pada data baru, nilai F5 adalah 1. Jadi, berdasarkan pohon keputusan, kita langsung dapat mengambil keputusan bahwa Label = 1 tanpa perlu melihat nilai fitur lainnya. Jadi, hasil prediksi untuk data baru [1, 0, 0, 0, 1] adalah Label = 1 dengan dilabelkan ke dalam pneumonia “Berat”.

6) Probabilitas:

Root node adalah F5 dengan data baru bernilai 1

Kelas Ringan = 0

Kelas Berat = 3 (D2, D7, D9)

- $P(R|D11)$ = Jumlah D11 yang berkelas Ringan/Jumlah Total Bernilai 1
 $= 0/3$
 $= 0$

$$\begin{aligned}
 P(B|D11) &= \text{Jumlah D11 yang berkelas Berat} / \text{Jumlah Total Bernilai 1} \\
 &= 3/3 \\
 &= 1
 \end{aligned}$$

Jadi, berdasarkan perhitungan probabilitas, data baru (D11) diklasifikasikan ke dalam kelas “Berat”

d. *Soft Voting Classifier*

1) Memanggil hasil probabilitas dari setiap algoritma klasifikasi yang telah diuji

- K-NN : $P(\text{Ringan}) = 0.33$, $P(\text{Berat}) = 0.67$
- SVM : $P(\text{Ringan}) = 0.56$, $P(\text{Berat}) = 0.44$
- Decision Tree : $P(\text{Ringan}) = 0$, $P(\text{Berat}) = 1$

2) Memasukkan bobot dengan nilai yang telah ditentukan sebelumnya

$$w = (0.520, 0.087, 0.393)$$

3) Menghitung nilai probabilitas

- $P(\text{Ringan}) = (0.520 \cdot 0.33) + (0.087 \cdot 0.56) + (0.393 \cdot 0)$
 $= 0.172 + 0.049 + 0$
 $= 0.221$
- $P(\text{Berat}) = (0.520 \cdot 0.67) + (0.087 \cdot 0.44) + (0.393 \cdot 1)$
 $= 0.348 + 0.038 + 0.393$
 $= 0.779$

4) Hasil dari *Soft Voting*

Berdasarkan hasil perhitungan probabilitas diatas, dapat dikatakan bahwa *Soft Voting* memberikan hasil prediksi Berat

Lampiran 9. Bukti Validasi Data Rekam Medis Rumah Sakit Y.W-UMI Makassar

SURAT KETERANGAN VALIDASI

Makassar, 8 Maret 2023

Hal : Permohonan Validasi Data Rekam Medis

Lampiran : -

Kepada Yth.

Kepala Instalasi Rekam Medis Rumah Sakit Ibnu Sina YW-UMI Makassar

Jl. Urip Sumoharjo No. 264, Kota Makassar, Sulawesi Selatan

Dengan hormat, kami sampaikan bahwa dalam rangka penyelesaian Skripsi / Tugas Akhir pada Program Studi Teknik Informatika Fakultas Teknik Informatika Universitas Hasanuddin, maka kami mohon Bapak berkenan memberikan validasi terhadap instrumen penelitian Tugas Akhir yang telah disusun oleh mahasiswa :

Nama (NIM) : Aries Wahyu Syaputra D121 17 1528

Judul TA : Implementasi Soft Voting Classifier untuk Diagnosis Pneumonia

Tujuan : Validasi Data Rekam Medis Pasien Pneumonia

Atas perhatian dan kerjasama yang baik kami sampaikan terima kasih.

a.n. Validator,
Kepala Instalasi Rekam Medis
Asnur Indrawan, Amd.PK
NIP. 500 191313

Lampiran 10. Sampel Data Pasien Pneumonia

frekuensi_napas	multilobar_infiltrate	confusion	kadar_ureum	leukosit	trombosit	suhu_tubuh	hipotensi	penggunaan_vasopresor	ventilasi_mekanis	pneumonia
24	tidak	tidak	10	13.100	309.000	38	tidak	tidak	tidak	ringan
26	tidak	tidak	10	7.000	107.000	37	tidak	tidak	tidak	bingan
29	tidak	tidak	12	25.200	304.000	37	tidak	tidak	tidak	ringan
30	ya	ya	52	3.200	63.000	35	ya	ya	ya	berat
30	tidak	tidak				37		tidak	tidak	ringan
40	tidak	tidak	14	3.900	126.000	37	tidak	tidak	ya	berat
30	tidak	tidak		14.400	145.000	38	tidak	tidak	tidak	ringan
43	tidak	ya	11	16.100	422.000	37	tidak	tidak	ya	berat
54	tidak	tidak	10	20.100	494.000	37	tidak	tidak	ya	berat
24	tidak	tidak	9	8.000	310.000	39	tidak	tidak	tidak	ringan
40	tidak	tidak	10	10.200	243.000	37	tidak	tidak	ya	berat
20	tidak	tidak	12	17.400	242.000	38	tidak	tidak	tidak	ringan
20	tidak	tidak	18	13.800	818.000	37	tidak	tidak	tidak	ringan
30	tidak	tidak	6	8.000	159.000	39	ya	tidak	tidak	ringan
30	tidak	tidak	7	16.600	99.000	37	tidak	tidak	tidak	ringan
30	tidak	tidak	12	3.700	114.000	37	tidak	tidak	tidak	ringan
28	tidak	tidak		16.100	256.000	40	tidak	tidak	tidak	ringan
30	tidak	tidak		8.700	562.000	38	tidak	tidak	tidak	ringan
30	ya	ya	100	3.200	28.800	36	ya	ya	ya	berat
30	tidak	tidak	11	17.700	435.000	40	tidak	tidak	tidak	ringan
32	tidak	tidak	11	16.100	454.000	37	tidak	tidak	tidak	ringan
33	tidak	tidak	13	4.700	251.000	38	tidak	tidak	tidak	ringan
28	tidak	tidak	10	9.200	361.000	39	tidak	tidak	tidak	ringan
22	tidak	tidak	12	19.500	323.000	37	tidak	tidak	tidak	ringan
32	tidak	ya	11	3.200	155.000	37	tidak	tidak	ya	berat

LEMBAR PERBAIKAN SKRIPSI

“IMPLEMENTASI SOFT VOTING CLASSIFIER UNTUK DIAGNOSIS PNEUMONIA”

OLEH:

**ARIES WAHYU SYAPUTRA
D121171528**

Skrripsi ini telah dipertahankan pada Ujian Akhir Sarjana tanggal 23 Agustus 2023.

Telah dilakukan perbaikan penulisan dan isi skripsi berdasarkan usulan dari penguji dan pembimbing skripsi.

Persetujuan perbaikan oleh tim penguji:

	Nama	Tanda Tangan
Ketua	Dr. Amil Ahmad Ilham, S.T., M.IT.	
Sekretaris	A. Ais Prayogi Alimuddin, S.T., M.Eng	
Anggota	Dr. Eng. Muhammad Niswar, S.T., M.IT.	
	Elly Warni S.T., M.T	

Persetujuan Perbaikan oleh pembimbing:

Pembimbing	Nama	Tanda Tangan
I	Dr. Amil Ahmad Ilham, S.T., M.IT.	
II	A. Ais Prayogi Alimuddin, S.T., M.Eng	