DAFTAR PUSTAKA

Aidasani, L. K., Bhadkamkar, H. and Kashyap, A. K. (2017) 'IoT: The kernel of smart cities', *ICONSTEM 2017 - Proceedings: 3rd IEEE International Conference on Science Technology, Engineering and Management*, 2018-Janua, pp. 8–11. doi: 10.1109/ICONSTEM.2017.8261248.

Airaksinen, M. *et al.* (2018) 'Smart city performance measurement framework CITYkeys', *2017 International Conference on Engineering, Technology and Innovation: Engineering, Technology and Innovation Management Beyond 2020: New Challenges, New Approaches, ICE/ITMC 2017 - Proceedings*, 2018-Janua, pp. 718–723. doi: 10.1109/ICE.2017.8279956.

Dqj, K. H. Q. *et al.* (2018) '$Gydqfhg &Rpelqhg /670 &11 0Rgho Iru 7Zlwwhu 6Hqwlphqw $Qdo\Vlv', *Advanced Combined LSTM-CNN Model for Twitter Sentiment Analysis*, pp. 684–687.

Gurcan, F. (2018) 'Multi-Class Classification of Turkish Texts with Machine Learning Algotirthms', *ISMSIT 2018 - 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies, Proceedings*, pp. 18–22. doi: 10.1109/ISMSIT.2018.8567307.

Al Helal, E. and Mokhtar, H. (2018) 'Towards Smart Riyadh : Riyadh Wiki Information and Complaining System', *International Journal of Managing Information Technology*, 10(2), pp. 95–106. doi: 10.5121/ijmit.2018.10206.

Hidayat, T., Suhardi and Kurniawan, N. B. (2018) 'Smart city service system engineering based on microservices architecture: Case study: Government of tangerang city', *2017 International Conference on ICT for Smart Society, ICISS 2017*, 2018-Janua, pp. 1–7. doi: 10.1109/ICTSS.2017.8288864.

Jang, B., Kim, I. and Kim, J. W. (2019) 'Word2vec convolutional neural networks for classification of news articles and tweets', *PLoS ONE*, 14(8), pp. 1–20. doi: 10.1371/journal.pone.0220976.

Jia, L. *et al.* (2020) 'Tibetan text classification method based on BiLSTM model', *Proceedings - International Conference on Artificial Intelligence and Electromechanical Automation, AIEA 2020*, pp. 27–31. doi: 10.1109/AIEA51086.2020.00013.

Karthika, P., Murugeswari, R. and Manoranjithem, R. (2019) 'Sentiment Analysis of Social Media Network Using Random Forest Algorithm', *IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing, INCOS 2019*, pp. 1–5. doi: 10.1109/INCOS45849.2019.8951367.

Kastrati, Z. *et al.* (2021) 'A deep learning sentiment analyser for social media comments in low-resource languages', *Electronics (Switzerland)*, 10(10), pp. 1–19. doi: 10.3390/electronics10101133.

Limon-Ruiz, M. *et al.* (2019) 'User-oriented representation of Smart Cities indicators to support citizens governments decision-making processes', *5th IEEE International Smart Cities Conference, ISC2 2019*, pp. 396–401. doi: 10.1109/ISC246665.2019.9071742.

Listyarini, S. N. and Anggoro, D. A. (2021) 'Analisis Sentimen Pilkada di Tengah Pandemi Covid-19 Menggunakan Convolution Neural Network (CNN)', *Jurnal Pendidikan dan Teknologi Indonesia*, 1(7), pp. 261–268. doi: 10.52436/1.jpti.60.

ıan, Y. and Lin, S. (2019) 'Research on Text Classification Based on CNN and ɔceedings of 2019 IEEE International Conference on Artificial Intelligence and Applications, ICAICA 2019*, pp. 352–355. doi: 10.1109/ICAICA.2019.8873454.

andera, P., Keuleers, E. and Brysbaert, M. (2017) 'Explaining human ıce in psycholinguistic tasks with models of semantic similarity based on

prediction and counting: A review and empirical validation', *Journal of Memory and Language*, 92, pp. 57–78. doi: 10.1016/j.jml.2016.04.001.

Manisha Adhude, Faizan Sayyed, Wassyatullah Sayed, A. P. K. V. W. (2019) '" Public Complaint Sorting Using Image Processing "', *Journal of Analysis and Computation (JAC)*, XII(I), pp. 2–5.

Mareeswari, V. and Gopalakrishnan, V. (2017) 'Complaint go: An online complaint registration system using web services and android', *IOP Conference Series: Materials Science and Engineering*, 263(4). doi: 10.1088/1757-899X/263/4/042073.

MK Gurucharan (2022) *Basic CNN Architecture: Explaining 5 Layers of Convolutional Neural Network*. Available at: https://www.upgrad.com/blog/basic-cnn-architecture/ (Accessed: 27 August 2022).

Muslih, Akhmad Rizaldy, Ajib Susanto, Ibnu Utomo Wahyu Mulyono, E. R. P. (2020) 'Implementasi algoritma boyer moore untuk analisis sentimen komentar di akun instagram instansi pemerintah', pp. 571–578.

Nengsih, W. (2017) 'Analisis Data Mining Tingkat Kepuasan Layanan Publik di Lingkungan Pemko Pekanbaru dengan Menggunakan Regresion Analysis General Linear Model', (February).

Prastyo, P. H. *et al.* (2020) 'Tweets Responding to the Indonesian Government's Handling of COVID-19: Sentiment Analysis Using SVM with Normalized Poly Kernel', *Journal of Information Systems Engineering and Business Intelligence*, 6(2), p. 112. doi: 10.20473/jisebi.6.2.112-122.

Priyadarshini, I. and Cotton, C. (2021) 'A novel LSTM–CNN–grid search-based deep neural network for sentiment analysis', *Journal of Supercomputing*, 77(12), pp. 13911–13932. doi: 10.1007/s11227-021-03838-w.

Protokol (2019) *Pengertian Sekretariat Daerah, Dinas Daerah, Lembaga Teknis Daerah, dan OPD*. Available at: https://setda.bone.go.id/2019/02/15/pengertian-sekretariat-daerah-dinas-daerah-lembaga-teknis-daerah-dan-opd/4/?fbclid=IwAR3_XuI7rs2uMkVz3mD62GQaI49F5eFNrp65AdSnJwDocpNbKZyT_gKXhgM (Accessed: 9 June 2023).

Putri, N. T. *et al.* (2020) 'Analisis Sentimen Opini Masyarakat Terhadap Pembangunan Infrastruktur Kota Malang Melalui Twiter Dengan Menggunakan Metode Support Vector Machine', *Seminar Informatika Aplikatif Polinema (Siap) 2020*, pp. 118–123.

Rehman, A. U. *et al.* (2019) 'A Hybrid CNN Model for Improving Accuracy of Movie Reviews Sentiment Analysis', *Multimedia Tools and Applications*, 78(18), pp. 26597–26613. doi: 10.1007/s11042-019-07788-7.

Shen, L. *et al.* (2018) 'A holistic evaluation of smart city performance in the context of China', *Journal of Cleaner Production*, 200, pp. 667–679. doi: 10.1016/j.jclepro.2018.07.281.

Siji George, C. G. and Sumathi, B. (2021) 'A novel deep learning approach of convolutional neural network and random forest classifier for fine-grained sentiment classification', *International Journal on Electrical Engineering and Informatics*, 13(2), pp. 465–476. doi: 10.15676/ijeei.2020.13.2.13.

Sun, X. and Zhou, M. (2010) 'Performance evaluation of transformation of industrial ecology in the resource-based cities', *Proceedings of the International e on E-Business and E-Government, ICEE 2010*, pp. 784–787. doi: CEE.2010.204.

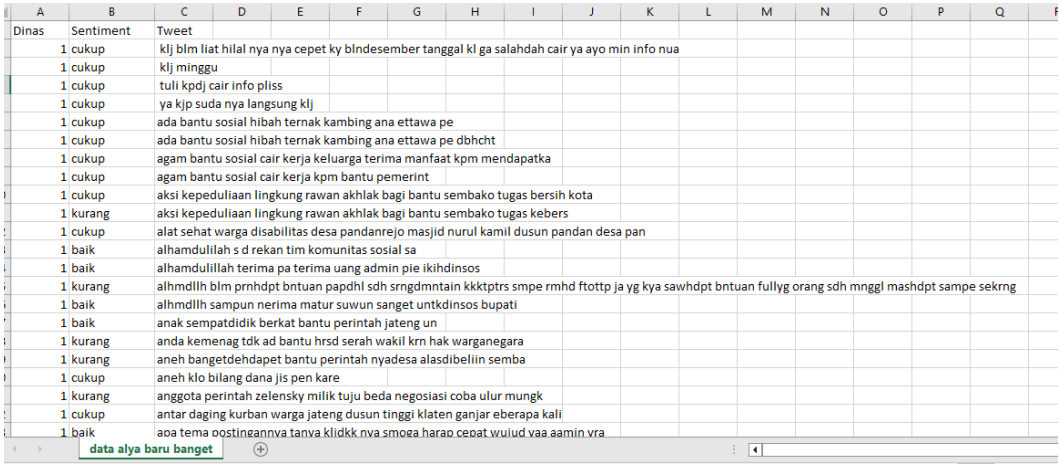
ıl, Q. *et al.* (2017) 'Sentiment Analysis Using Deep Learning Techniques: A nternational Journal of Advanced Computer Science and Applications*, 8(6). doi: ijacsa.2017.080657.

Vanaja, S. (2018) 'Aspect-Level Sentiment Analysis on E-Commerce Data', *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, (Icirca), pp. 1275–1279.

Xu, G. *et al.* (2019) 'Sentiment analysis of comment texts based on BiLSTM', *IEEE Access*, 7(c), pp. 51522–51532. doi: 10.1109/ACCESS.2019.2909919.

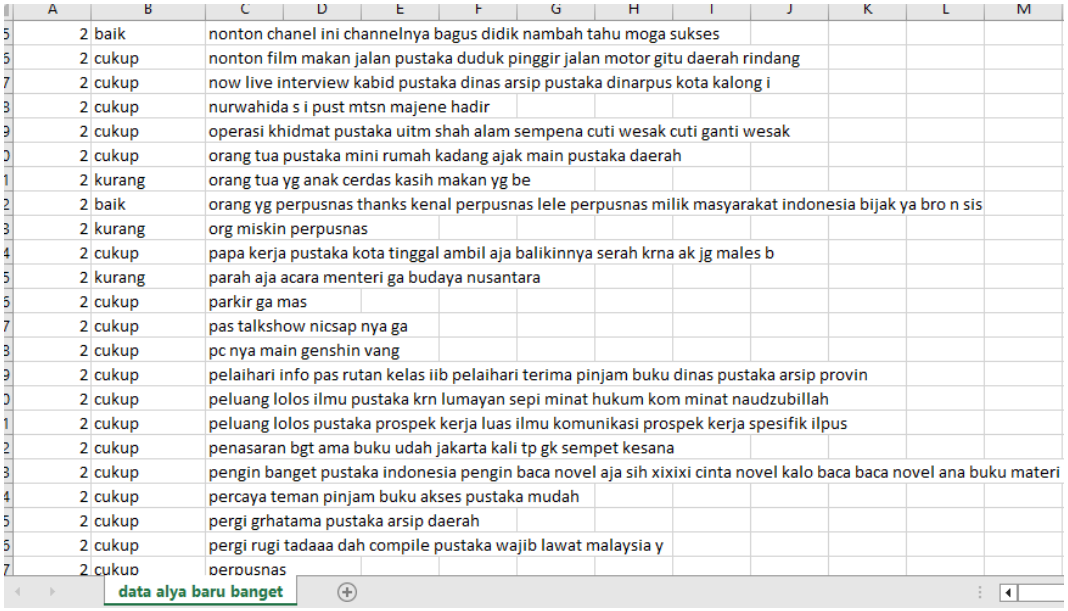
LAMPIRAN

Data Set Setelah Preprocessing



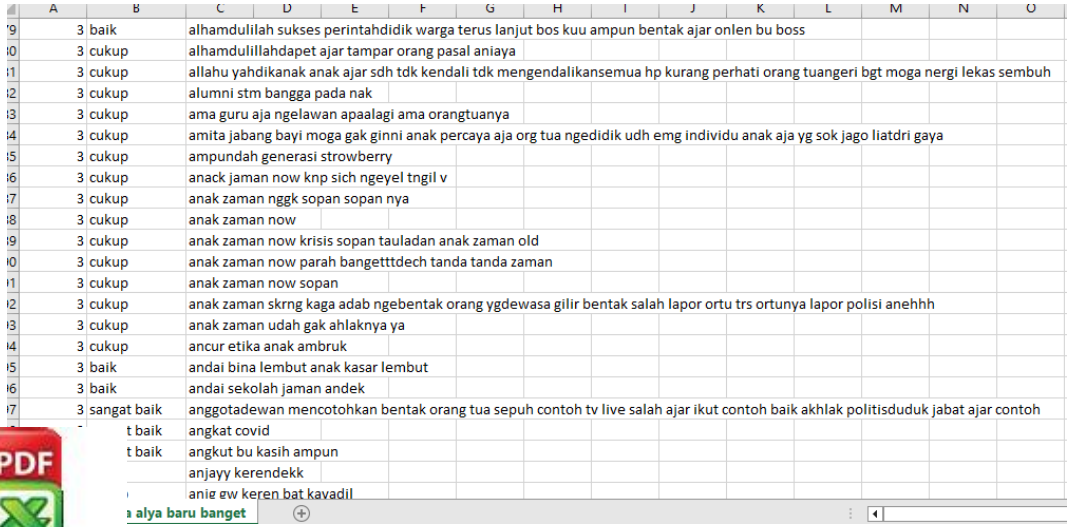
Dinas	Sentiment	Tweet
1	cukup	klj blm liat hilal nya nya cepet ky blndesember tanggal kl ga salahdah cair ya ayo min info nua
1	cukup	klj minggu
1	cukup	tuli kpdj cair info pliss
1	cukup	ya kjp suda nya langsung klj
1	cukup	ada bantu sosial hibah ternak kambing ana ettawa pe
1	cukup	ada bantu sosial hibah ternak kambing ana ettawa pe dbhcht
1	cukup	agam bantu sosial cair kerja keluarga terima manfaat kpm mendapatka
1	cukup	agam bantu sosial cair kerja kpm bantu pemerint
1	cukup	aksi kepeduliaan lingkung rawan akhlak bagi bantu sembako tugas bersih kota
1	kurang	aksi kepeduliaan lingkung rawan akhlak bagi bantu sembako tugas kebers
1	cukup	alat sehat warga disabilitas desa pandanrejo masjid nurul kamil dusun pandan desa pan
1	baik	alhamdulilah s d rekan tim komunitas sosial sa
1	baik	alhamdulillah terima pa terima uang admin pie ikihdinsos
1	kurang	alhmdllh blm prnhdpt bntuan papdhl sdh srngdmntain kkktptrs smpe rmhd ftottp ja yg kya sawhdpt bntuan fullyg orang sdh mnggl mashdpt sampe sekrng
1	baik	alhmdllh sampun nerima matur suwun sanget untkdinsos bupati
1	baik	anak sempatdidik berkat bantu perintah jateng un
1	kurang	anda kemenag tdk ad bantu hrsd serah wakil krn hak warganegara
1	kurang	aneh bangetdehdapet bantu perintah nyadesa alasdibeliin semba
1	cukup	aneh klo bilang dana jis pen kare
1	kurang	anggota perintah zelensky milik tuju beda negosiasi coba ulur mungk
1	cukup	antar daging kurban warga jateng dusun tinggi klaten ganjar eberapa kali
1	baik	apa tema postingannya tanya klidkk nya smoga harap cepat wujud yaa aamin yra

data alya baru banget



2	baik	nonton chanel ini channelnya bagus didik nambah tahu moga sukses
2	cukup	nonton film makan jalan pustaka duduk pinggir jalan motor gitu daerah rindang
2	cukup	now live interview kabid pustaka dinas arsip pustaka dinarpus kota kalong i
2	cukup	nurwahida s i pust mtsn majene hadir
2	cukup	operasi khidmat pustaka uitm shah alam sempena cuti wesak cuti ganti wesak
2	cukup	orang tua pustaka mini rumah kadang ajak main pustaka daerah
2	kurang	orang tua yg anak cerdas kasih makan yg be
2	baik	orang yg perpusnas thanks kenal perpusnas lele perpusnas milik masyarakat indonesia bijak ya bro n sis
2	kurang	org miskin perpusnas
2	cukup	papa kerja pustaka kota tinggal ambil aja balikinnya serah krna ak jg males b
2	kurang	parah aja acara menteri ga budaya nusantara
2	cukup	parkir ga mas
2	cukup	pas talkshow nicsap nya ga
2	cukup	pc nya main genshin vang
2	cukup	pelaihari info pas rutan kelas iib pelaihari terima pinjam buku dinas pustaka arsip provin
2	cukup	peluang lolos ilmu pustaka krn lumayan sepi minat hukum kom minat naudzubillah
2	cukup	peluang lolos pustaka prospek kerja luas ilmu komunikasi prospek kerja spesifik ilpus
2	cukup	penasaran bgt ama buku udah jakarta kali tp gk sempet kesana
2	cukup	pengin banget pustaka indonesia pengin baca novel aja sih xixixi cinta novel kalo baca baca novel ana buku materi
2	cukup	percaya teman pinjam buku akses pustaka mudah
2	cukup	pergi grhatama pustaka arsip daerah
2	cukup	pergi rugi tadaaa dah compile pustaka wajib lawat malaysia y
2	cukup	perpusnas

data alya baru banget



3	baik	alhamdulilah sukses perintahdidik warga terus lanjut bos kuu ampun bentak ajar onlen bu boss
3	cukup	alhamdulillahdapet ajar tampar orang pasal aniaya
3	cukup	allahu yahdikanak anak ajar sdh tdk kendali tdk mengendalikansemua hp kurang perhati orang tuangeri bgt moga nergi lekas sembuh
3	cukup	alumni stm bangga pada nak
3	cukup	ama guru aja ngelawan apaalagi ama orangtuanya
3	cukup	amita jabang bayi moga gak ginni anak percaya aja org tua ngedidik udh emg individu anak aja yg sok jago liatdri gaya
3	cukup	ampundah generasi strowberry
3	cukup	anack jaman now knp sich ngeyel tngil v
3	cukup	anak zaman nggk sopan sopan nya
3	cukup	anak zaman now
3	cukup	anak zaman now krisis sopan tauladan anak zaman old
3	cukup	anak zaman now parah bangetttdech tanda tanda zaman
3	cukup	anak zaman now sopan
3	cukup	anak zaman skrng kaga adab ngebentak orang ygdewasa gilir bentak salah lapor ortu trs ortunya lapor polisi anehhh
3	cukup	anak zaman udah gak ahlaknya ya
3	cukup	ancur etika anak ambruk
3	baik	andai bina lembut anak kasar lembut
3	baik	andai sekolah jaman andek
3	sangat baik	anggotadewan mencotohkan bentak orang tua sepuh contoh tv live salah ajar ikut contoh baik akhlak politisduduk jabat ajar contoh
	t baik	angkat covid
	t baik	angkut bu kasih ampun
		anjayy kerendekk
		anig gw keren bat kayadil

a alya baru banget

Data Set Hasil Sistem

	A	B	C	D	E	F	G
1		id	Dinas_x	Sentimen	predictior	predictior	Tweet
2	0	5697	12	1	8	1	jogging aja gerbang sampe upt terna masjid batu
3	1	211	1	4	14	4	januari februari zonkkk
4	1	211	1	4	14	4	januari februari zonkkk
5	1	211	1	4	14	4	januari februari zonkkk
6	1	211	1	4	14	4	januari februari zonkkk
7	1	211	1	4	14	4	januari februari zonkkk
8	1	211	1	4	14	4	januari februari zonkkk
9	1	211	1	4	14	4	januari februari zonkkk
10	1	211	1	4	14	4	januari februari zonkkk
11	1	211	1	4	14	4	januari februari zonkkk
12	1	211	1	4	14	4	januari februari zonkkk
13	1	211	1	4	14	4	januari februari zonkkk
14	1	211	1	4	14	4	januari februari zonkkk
15	1	211	1	4	14	4	januari februari zonkkk
16	1	211	1	4	14	4	januari februari zonkkk
17	1	211	1	4	14	4	januari februari zonkkk
18	1	211	1	4	14	4	januari februari zonkkk
19	1	211	1	4	14	4	januari februari zonkkk
20	1	211	1	4	14	4	januari februari zonkkk
21	1	211	1	4	14	4	januari februari zonkkk
22	1	211	1	4	14	4	januari februari zonkkk
23	1	211	1	4	14	4	januari februari zonkkk

hasil prediksi test (1)

A	B	C	D	E	F	G
2110	614	2	3	2	3	guna bantu trimakasih mas rotmi moga e-class mudah jangkau barokimullah mas rotmi tim
2110	614	2	3	2	3	guna bantu trimakasih mas rotmi moga e-class mudah jangkau barokimullah mas rotmi tim
2110	614	2	3	2	3	guna bantu trimakasih mas rotmi moga e-class mudah jangkau barokimullah mas rotmi tim
2110	614	2	3	2	3	guna bantu trimakasih mas rotmi moga e-class mudah jangkau barokimullah mas rotmi tim
2110	614	2	3	2	3	guna bantu trimakasih mas rotmi moga e-class mudah jangkau barokimullah mas rotmi tim
2110	614	2	3	2	3	guna bantu trimakasih mas rotmi moga e-class mudah jangkau barokimullah mas rotmi tim
2110	614	2	3	2	3	guna bantu trimakasih mas rotmi moga e-class mudah jangkau barokimullah mas rotmi tim
2110	614	2	3	2	3	guna bantu trimakasih mas rotmi moga e-class mudah jangkau barokimullah mas rotmi tim
2110	614	2	3	2	3	guna bantu trimakasih mas rotmi moga e-class mudah jangkau barokimullah mas rotmi tim
2110	614	2	3	2	3	guna bantu trimakasih mas rotmi moga e-class mudah jangkau barokimullah mas rotmi tim
2110	614	2	3	2	3	guna bantu trimakasih mas rotmi moga e-class mudah jangkau barokimullah mas rotmi tim
2110	614	2	3	2	3	guna bantu trimakasih mas rotmi moga e-class mudah jangkau barokimullah mas rotmi tim
2110	614	2	3	2	3	guna bantu trimakasih mas rotmi moga e-class mudah jangkau barokimullah mas rotmi tim
2110	614	2	3	2	3	guna bantu trimakasih mas rotmi moga e-class mudah jangkau barokimullah mas rotmi tim
2110	614	2	3	2	3	guna bantu trimakasih mas rotmi moga e-class mudah jangkau barokimullah mas rotmi tim
2110	614	2	3	2	3	guna bantu trimakasih mas rotmi moga e-class mudah jangkau barokimullah mas rotmi tim
2110	614	2	3	2	3	guna bantu trimakasih mas rotmi moga e-class mudah jangkau barokimullah mas rotmi tim
2110	614	2	3	2	3	guna bantu trimakasih mas rotmi moga e-class mudah jangkau barokimullah mas rotmi tim
2110	614	2	3	2	3	guna bantu trimakasih mas rotmi moga e-class mudah jangkau barokimullah mas rotmi tim
3239	614	2	3	2	3	guna bantu trimakasih mas rotmi moga e-class mudah jangkau barokimullah mas rotmi tim
3239	614	2	3	2	3	guna bantu trimakasih mas rotmi moga e-class mudah jangkau barokimullah mas rotmi tim
3239	614	2	3	2	3	guna bantu trimakasih mas rotmi moga e-class mudah jangkau barokimullah mas rotmi tim
3239	614	2	3	2	3	guna bantu trimakasih mas rotmi moga e-class mudah jangkau barokimullah mas rotmi tim

hasil prediksi test (1)

	A	B	C	D	E	F	G	H	I	J
42833	620	3886	8	4	8	4	stop tipu masyarakat muak			
42834	620	3886	8	4	8	4	stop tipu masyarakat muak			
42835	620	3886	8	4	8	4	stop tipu masyarakat muak			
42836	620	3886	8	4	8	4	stop tipu masyarakat muak			
42837	620	3886	8	4	8	4	stop tipu masyarakat muak			
42838	620	3886	8	4	8	4	stop tipu masyarakat muak			
42839	620	3886	8	4	8	4	stop tipu masyarakat muak			
42840	620	3886	8	4	8	4	stop tipu masyarakat muak			
42841	620	3886	8	4	8	4	stop tipu masyarakat muak			
42842	620	3886	8	4	8	4	stop tipu masyarakat muak			
42843	620	3886	8	4	8	4	stop tipu masyarakat muak			
42844	620	3886	8	4	8	4	stop tipu masyarakat muak			
42845	620	3886	8	4	8	4	stop tipu masyarakat muak			
42846	620	3886	8	4	8	4	stop tipu masyarakat muak			
42847	620	3886	8	4	8	4	stop tipu masyarakat muak			
42848	620	3886	8	4	8	4	stop tipu masyarakat muak			
42849	620	3886	8	4	8	4	stop tipu masyarakat muak			
42850	620	3886	8	4	8	4	stop tipu masyarakat muak			
42851	620	3886	8	4	8	4	stop tipu masyarakat muak			
42852	620	3886	8	4	8	4	stop tipu masyarakat muak			
42853	620	3886	8	4	8	4	stop tipu masyarakat muak			
42854	620	3886	8	4	8	4	stop tipu masyarakat muak			
42855	620	3886	8	4	8	4	stop tipu masyarakat muak			

hasil prediksi test (1)

	A	B	C	D	E	F	G	H	I	J
61572	459	414	1	0	14	4	sekian kljadm nyadiam bingung			
61573	459	414	1	0	14	4	sekian kljadm nyadiam bingung			
61574	459	414	1	0	14	4	sekian kljadm nyadiam bingung			
61575	459	414	1	0	14	4	sekian kljadm nyadiam bingung			
61576	459	414	1	0	14	4	sekian kljadm nyadiam bingung			
61577	459	414	1	0	14	4	sekian kljadm nyadiam bingung			
61578	459	414	1	0	14	4	sekian kljadm nyadiam bingung			
61579	459	414	1	0	14	4	sekian kljadm nyadiam bingung			
61580	459	414	1	0	14	4	sekian kljadm nyadiam bingung			
61581	460	7508	16	0	14	0	alhamdulillah			
61582	460	7508	16	0	14	0	alhamdulillah			
61583	460	7508	16	0	14	0	alhamdulillah			
61584	460	7508	16	0	14	0	alhamdulillah			
61585	461	5515	12	0	12	1	alhamdulilah bimtekdinas terna			
61586	461	5515	12	0	12	1	alhamdulilah bimtekdinas terna			
61587	461	5515	12	0	12	1	alhamdulilah bimtekdinas terna			
61588	461	5515	12	0	12	1	alhamdulilah bimtekdinas terna			
61589	461	5515	12	0	12	1	alhamdulilah bimtekdinas terna			
61590	461	5515	12	0	12	1	alhamdulilah bimtekdinas terna			
61591	461	5515	12	0	12	1	alhamdulilah bimtekdinas terna			
61592	461	5515	12	0	12	1	alhamdulilah bimtekdinas terna			
61593	461	5515	12	0	12	1	alhamdulilah bimtekdinas terna			
61594	2013	5515	12	0	12	1	alhamdulilah bimtekdinas terna			

hasil prediksi test (1)

SOURCE CODE & PROCESS

(Berdasarkan blog diagram pada halaman 35)

1. CRAWLING DATA

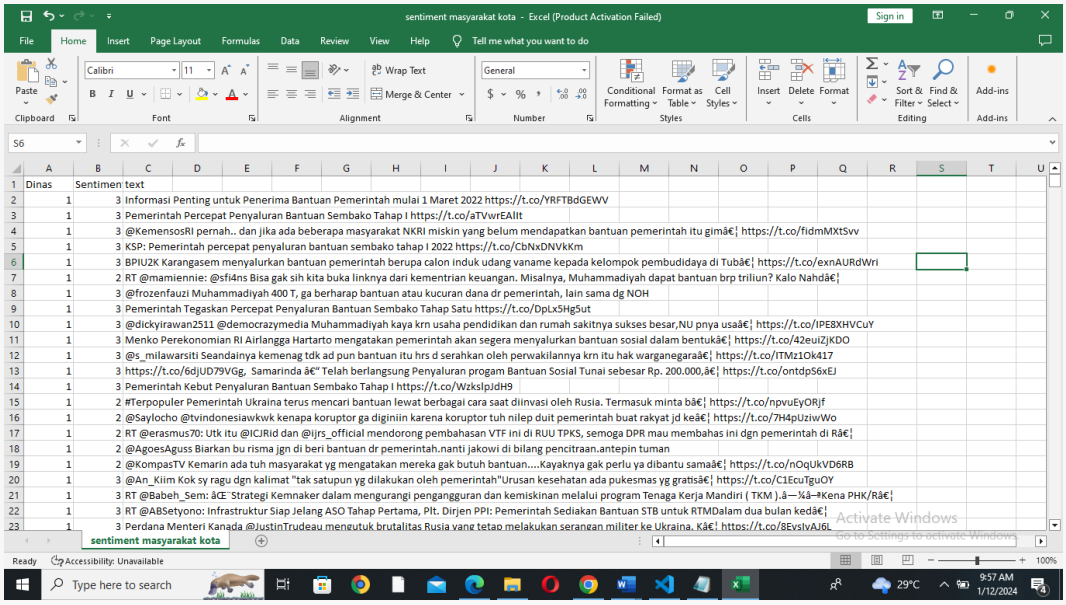
Data diambil dari berbagai media sosial seperti twitter, instagram, facebook dan berita di youtube. Adapun data dari Facebook, Instagram, Youtube kami ambil dari situs www.exportcomments.com dan berikut ini adalah source code untuk mengambil data dari twitter :

```
import tweepy,sys,jsonpickle


consumer_key = 'PEJ97NNDjxBBZuoRpo0kcni5m'        #silahkan isi sesuai akun
twitter developer kalian
consumer_secret =
'w7C7OKtE47yZTf7ZnPeotkAVLUw2BFb1OTiy00NIOq9QnHC1TQ'
#silahkan isi sesuai akun twitter developer kalian
print("#==============================================#")
print("CRAWLING METHOD")
print("\t Alya Rohalia")
print("\t D082192002")
print("#==============================================#")

# qry='peternakan'
qry ='peternakan OR "perikanan"'
maxTweets = 5000# Isi sembarang nilai sesuai kebutuhan anda
tweetsPerQry = 100    # Jangan isi lebih dari 100, tidak boleh oleh Twitter
fName='crawling_peternakan.json' # Nama File hasil Crawling
#fName='Hasil_perpusnas.json' # Nama File hasil Crawling 12
auth = tweepy.AppAuthHandler(consumer_key,consumer_secret)
api = tweepy.API(auth, wait_on_rate_limit=True,wait_on_rate_limit_notify
=True)
if (not api):
        sys.exit('Autentikasi gagal, mohon cek "Consumer Key" & "Consumer
Secret" Twitter anda')
sinceId=None;max_id=-1;tweetCount=0
print("Mulai mengunduh maksimum {0} tweets" .format(maxTweets))
with open(fName,'w') as f:
    while tweetCount < maxTweets:
        try:
            if (max_id <= 0):
                if (not sinceId):
                    new_tweets=api.search(q=qry,count=tweetsPerQry)
```

```
                    else:

new_tweets=api.search(q=qry,count=tweetsPerQry,since_id=sinceId)
                else:
                    if (not sinceId):

new_tweets=api.search(q=qry,count=tweetsPerQry,max_id=str(max_id - 1
                        ))
                    else:

new_tweets=api.search(q=qry,count=tweetsPerQry,max_id=str(max_id - 1
                        ),since_id=sinceId)
                if not new_tweets:
                    print('Tidak ada lagi Tweet ditemukan dengan Query="{0}"'
.format(qry));
                    break
                for tweet in new_tweets:
                    f.write(jsonpickle.encode(tweet._json,unpicklable=False)+'\n')
                tweetCount+=len(new_tweets)
                sys.stdout.write("\r");sys.stdout.write("Jumlah Tweets telah
tersimpan: %.0f" %tweetCount);sys.stdout.flush()
                max_id=new_tweets[-1].id
            except tweepy.TweepError as e:
                print("some error : " + str(e));break # Aya error, keluar
    print ('\nSelesai! {0} tweets tersimpan di "{1}"' .format(tweetCount,fName))
```

2. LABELLING DATA

Data yang sudah di crawling/ yang telah diambil dari berbagai media sosial kita beri label seperti dibawah ini dan kita simpan dalam format .csv :



	A	B	C
1	Dinas	Sentimen	text
2	1	3	Informasi Penting untuk Penerima Bantuan Pemerintah mulai 1 Maret 2022 https://t.co/YRFTBdGEWV
3	1	3	Pemerintah Percepat Penyaluran Bantuan Sembako Tahap I https://t.co/aTVwrEAlIt
4	1	3	@KemensosRI pernah.. dan jika ada beberapa masyarakat NKRI miskin yang belum mendapatkan bantuan pemerintah itu gimâ€¦ https://t.co/fidmMXtSvv
5	1	3	KSP: Pemerintah percepat penyaluran bantuan sembako tahap I 2022 https://t.co/CbNxDNVkKm
6	1	3	BPIU2K Karangasem menyalurkan bantuan pemerintah berupa calon induk udang vaname kepada kelompok pembudidaya di Tubâ€¦ https://t.co/exnAURdWri
7	1	2	RT @mamiennie: @sfi4ns Bisa gak sih kita buka linknya dari kementrian keuangan. Misalnya, Muhammadiyah dapat bantuan brp triliun? Kalo Nahdâ€¦
8	1	3	@frozenfauzi Muhammadiyah 400 T, ga berharap bantuan atau kucuran dana dr pemerintah, lain sama dg NOH
9	1	3	Pemerintah Tegaskan Percepat Penyaluran Bantuan Sembako Tahap Satu https://t.co/DpLx5Hg5ut
10	1	3	@dickyirawan2511 @democrazymedia Muhammadiyah kaya krn usaha pendidikan dan rumah sakitnya sukses besar,NU pnya usaâ€¦ https://t.co/IPE8XHVCuY
11	1	3	Menko Perekonomian RI Airlangga Hartarto mengatakan pemerintah akan segera menyalurkan bantuan sosial dalam bentukâ€¦ https://t.co/42euiZjKDO
12	1	3	@s_milawarsiti Seandainya kemenag tdk ad pun bantuan itu hrs d serahkan oleh perwakilannya krn itu hak warganegaraâ€¦ https://t.co/iTMz1Ok417
13	1	3	https://t.co/6djUD79VGg, Samarinda â€" Telah berlangsung Penyaluran progam Bantuan Sosial Tunai sebesar Rp. 200.000,â€¦ https://t.co/ontdpS6xEJ
14	1	3	Pemerintah Kebut Penyaluran Bantuan Sembako Tahap I https://t.co/WzkslpJdH9
15	1	2	#Terpopuler Pemerintah Ukraina terus mencari bantuan lewat berbagai cara saat diinvasi oleh Rusia. Termasuk minta bâ€¦ https://t.co/npvuEyORjf
16	1	2	@Saylocho @tvindonesiawkwk kenapa koruptor ga digiiiin karena koruptor tuh nilep duit pemerintah buat rakyat jd keâ€¦ https://t.co/7H4pUziwWo
17	1	2	RT @erasmus70: Utk itu @ICJRid dan @ijrs_official mendorong pembahasan VTF ini di RUU TPKS, semoga DPR mau membahas ini dgn pemerintah di Râ€¦
18	1	2	@AgoesAguss Biarkan bu risma jgn di beri bantuan dr pemerintah.nanti jakowi di bilang pencitraan.antepin tuman
19	1	2	@KompasTV Kemarin ada tuh masyarakat yg mengatakan mereka gak butuh bantuan....Kayaknya gak perlu ya dibantu samaâ€¦ https://t.co/nOqUkVD6RB
20	1	3	@An_Kiim Kok sy ragu dgn kalimat "tak satupun yg dilakukan oleh pemerintah"Urusan kesehatan ada pukesmas yg gratisâ€¦ https://t.co/C1EcuTguOY
21	1	3	RT @Babeh_Sem: âŒ Strategi Kemnaker dalam mengurangi pengangguran dan kemiskinan melalui program Tenaga Kerja Mandiri (TKM).â—¾Kena PHK/Râ€¦
22	1	3	RT @ABSetyono: Infrastruktur Siap Jelang ASO Tahap Pertama, Plt. Dirjen PPI: Pemerintah Sediakan Bantuan STB untuk RTMDalam dua bulan kedâ€¦
23	1	3	Perdana Menteri Kanada @JustinTrudeau mengutuk brutalitas Rusia yang tetap melakukan serangan militer ke Ukraina. Kâ€¦ https://t.co/8EvsIvAJ6L

3. PRE PROCESSING NLP

Hapus link dan mention

hapus tanda baca

Tokenization - Converting a sentence into list of words

Remove stopwords

Lammetization/stemming - Tranforming any form of a word to its root word

#hapus link dan mention

```
def cleaner(tweet):
    tweet = re.sub("@[A-Za-z0-9]+","",tweet) #Remove @ sign
    tweet = re.sub(r"(?:\@|http?\://|https?\://|www)\S+", "",
tweet) #Remove http links
    tweet = " ".join(tweet.split())
      eet = ''.join(c for c in tweet if c not in
      NICODE_EMOJI) #Remove Emojis. untuk ini butuh package

      et = tweet.replace("#", "").replace("_", " ") #Remove
       sign but keep the text
```

```
    tweet = tweet.lower()
    return tweet
df_preprocessed['Tweet'] = df_preprocessed['text'].map(lambda
x: cleaner(x))
df_preprocessed.head(10)
```

#fungsi hapus tanda baca

```
print('tanda baca: ',string.punctuation)
def remove_punct(text):
    text  = "".join([char for char in text if char not in
string.punctuation])
    text = re.sub('[0-9]+', ' ', text)
    return text

df_preprocessed['Tweet'] =
df_preprocessed['Tweet'].apply(lambda x: remove_punct(x))
df_preprocessed.head(10)
```

#Hapus stopword

```
#ambil list stopwordnya
from Sastrawi.StopWordRemover.StopWordRemoverFactory import
StopWordRemoverFactory
stop_factory = StopWordRemoverFactory()
more_stopword = ['nya', 'ia','bahwa','oleh'] #untuk tambah
stopword kalau dirasa perlu
data = stop_factory.get_stop_words()+more_stopword #pake ini
kalau ada tambahan stopword
data = stop_factory.get_stop_words()
stopword = stop_factory.create_stop_word_remover()

df_preprocessed['Tweet'] =
df_preprocessed['Tweet'].apply(lambda x: stopword.remove(x))
df_preprocessed.head()
```

#Bikin stemmer

```
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
factory = StemmerFactory()
stemmer = factory.create_stemmer()

rocessed['Tweet'] =
rocessed['Tweet'].apply(lambda x: stemmer.stem(x))
rocessed.head(10)
```

Export data yang suudah di clean ke csv baru

```
df_preprocessed.to_csv('data_cleaned.csv')
```

4. MEMBANGUN MODELS

Kita akan membangun models CNNRF dengan mengimport library yang kita butuhkan :

```
from __future__ import absolute_import, division,
print_function, unicode_literals

try:
  # %tensorflow_version only exists in Colab.
  %tensorflow_version 2.x
except Exception:
  pass
import tensorflow as tf

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from gensim.corpora import WikiCorpus
from urllib.request import urlopen
from collections import Counter
import gzip
import numpy as np

from tensorflow.keras import regularizers

from tensorflow.keras import layers
from tensorflow.keras import losses
```

```
! pip install Sastrawi
```

5.

Load & Preprocess

```
an load dataset terlebih dahulu
ew = pd.read_csv("data alya baru banget..csv")
```

```
# drop unused column
df_preprocessed = df_review.copy()

#ubah data dinas dan sentimen dari categorical jadi ordinal.
karena datanya kotor sekali
from sklearn.preprocessing import LabelEncoder
LE = LabelEncoder()
#df_preprocessed['Dinas'] =
LE.fit_transform(df_preprocessed['Dinas'])
df_preprocessed['Sentiment'] =
LE.fit_transform(df_preprocessed['Sentiment'])

print(df_preprocessed.head())
print("------------")
print(df_preprocessed["Dinas"].value_counts())
print("------------")
print(df_preprocessed["Sentiment"].value_counts())
print("------------")
list(LE.inverse_transform([0, 1, 2,3,4,5]))
```

```
#print dataframe data input yang sudah di preprocessed
#komen data ini kalau jadinya terlalu banyak dan bikin jadi
lambat

%load_ext google.colab.data_table
df_preprocessed
```

STEMMER

Buat stemmer lalu preprocess data tweet nya :

```
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
factory = StemmerFactory()
stemmer = factory.create_stemmer()

# contoh
kalimat = df_preprocessed['Tweet'].iloc[0]
print("kalimat test stemmer: ",kalimat)
katadasar = stemmer.stem(kalimat)

print("kalimat hasil stemm: ",katadasar)
```

```
alimat tweet
= []
ex, row in df_preprocessed.iterrows():
iew.append(stemmer.stem(row["Tweet"]))
```

```
df_preprocessed["Tweet"] = review
#df_preprocessed.head()
#cetak hasil preprocess yang sudah di stem
df_preprocessed
```

```
#hasil cleaning di taruh ke csv
df_preprocessed.to_csv('data alya kotor dibersihkan.csv')
```

```
# #jalankan ini cuma sudah punya data yang sudah dibersihkan.
# # lakukan load dataset terlebih dahulu

# df_preprocessed = pd.read_csv("data alya kotor
dibersihkan.csv")
# df_preprocessed.drop(columns=df_preprocessed.columns[0],
inplace=True, axis=1)
# df_preprocessed =
df_preprocessed.astype({'Sentiment':'int'})
# df_preprocessed = df_preprocessed.astype({'Dinas':'int'})
# print(df_preprocessed.head())
# print("------------")
# print(df_preprocessed["Dinas"].value_counts())
# print("------------")
# print(df_preprocessed["Sentiment"].value_counts())
# print("------------")
# list(LE.inverse_transform([0, 1, 2,3,4,5]))
```

```
df_preprocessed
```

```
# balancing dataset oversampling sederhana
#limit data ke 5000
s_0 =
df_preprocessed[df_preprocessed['Sentiment']==0].sample(5000,r
eplace=True)
s_1 =
df_preprocessed[df_preprocessed['Sentiment']==1].sample(5000,r
eplace=True)
s_2 =
df_preprocessed[df_preprocessed['Sentiment']==2].sample(5000,r
eplace=True)
s_3 =
df_preprocessed[df_preprocessed['Sentiment']==3].sample(5000,r
eplace=True)
rocessed[df_preprocessed['Sentiment']==4].sample(5000,r
True)
rocessed = pd.concat([s_0, s_1, s_2, s_3, s_4])
```

```
print(df_preprocessed.shape)

print(df_preprocessed['Sentiment'].value_counts(normalize=True
))
print(Counter(df_preprocessed["Sentiment"]))
```

```
#Dataset hasil oversampling
df_preprocessed
```

6. WORD EMBEDDING (FASTTEKS VECTORS)

Embedding pake fasttext :

```
file =
gzip.open(urlopen('https://dl.fbaipublicfiles.com/fasttext/vec
tors-crawl/cc.id.300.vec.gz'))

vocab_and_vectors = {}
# put words as dict indexes and vectors as words values
for line in file:
    values = line.split()
    word = values [0].decode('utf-8')
    vector = np.asarray(values[1:], dtype='float32')
    vocab_and_vectors[word] = vector
```

```
print(df_preprocessed.head())
print("------------")
print(df_preprocessed["Dinas"].value_counts())
print("------------")
print(df_preprocessed["Sentiment"].value_counts())
print("------------")
list(LE.inverse_transform([0, 1, 2,3,4,5]))
```

```
# more imports
from sklearn.model_selection import train_test_split
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import
pad_sequences
from tensorflow.keras.utils import to_categorical
# berapa fitur yang mau di extract
    s = 500
    er = Tokenizer(num_words = features)
    okenizer ke text
    er.fit_on_texts(df_preprocessed["Tweet"].tolist())
```

```
# ambil semua kata yang ditahu sama tokenizernya
word_index = tokenizer.word_index
print(len(word_index))
# masukkan token ke matrix
X =
tokenizer.texts_to_sequences(df_preprocessed["Tweet"].tolist()
)
print("x yang telah diubah jadi token",X)
X = pad_sequences(X)
print("x setelah di beri padding pad",X)

# Siapkan labelnya
y = df_preprocessed.iloc[:, 0:2]
print("======== Y Label ==========")
print(y["Dinas"].value_counts())
print(y["Sentiment"].value_counts())

#cetak token x yang telah di pad
X
```

```
# split data train dan test
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42) #tambah stratify=y
print(X_train.shape, y_train.shape)
print(X_test.shape, y_test.shape)
```

```
embedding_matrix = np.zeros((len(word_index) + 1, 300))
for word, i in word_index.items():
    embedding_vector = vocab_and_vectors.get(word)
    # words that cannot be found will be set to 0
    if embedding_vector is not None:
        embedding_matrix[i] = embedding_vector

embedding_matrix.shape
```

7. . RANDOM FOREST CLASSIFICATION PARAMETER

CNN Classifier

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Embedding

= tf.keras.Sequential([
.keras.layers.Embedding(len(word_index)+1, 300,
ength=X.shape[1], weights=[embedding_matrix],
le=False),
```

```
#    tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(64,  r
eturn_sequences=True)),
#    tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(32)),
#    tf.keras.layers.Dense(64, activation='relu'),
#    tf.keras.layers.Dropout(0.5),
#    tf.keras.layers.Dense(1, activation='sigmoid')
#])
embedding_dim = 300
input_length=X.shape[1]

model = tf.keras.Sequential()

#1. Embedding Layer & Weight
model.add(tf.keras.layers.Embedding(len(word_index)+1,
embedding_dim, input_length=input_length,
         weights=[embedding_matrix], trainable=False))


model.add(tf.keras.layers.Conv1D(filters=16, kernel_size=3,
strides=1, padding='same',
activation='relu',kernel_initializer=tf.keras.initializers.ran
dom_normal()))
model.add(tf.keras.layers.BatchNormalization())
model.add(tf.keras.layers.Activation('relu'))

model.add(tf.keras.layers.Conv1D(filters=16, kernel_size=3,
strides=1, padding='same',
activation='relu',kernel_initializer=tf.keras.initializers.ran
dom_normal()))
model.add(tf.keras.layers.BatchNormalization())
model.add(tf.keras.layers.Activation('relu'))

model.add(tf.keras.layers.MaxPooling1D())

model.add(tf.keras.layers.Conv1D(filters=32, kernel_size=3,
strides=1, padding='same',
activation='relu',kernel_initializer=tf.keras.initializers.ran
dom_normal()))
model.add(tf.keras.layers.BatchNormalization())
model.add(tf.keras.layers.Activation('relu'))

model.add(tf.keras.layers.Conv1D(filters=32, kernel_size=3,
=1, padding='same',
ion='relu',kernel_initializer=tf.keras.initializers.ran
mal()))
dd(tf.keras.layers.BatchNormalization())
dd(tf.keras.layers.Activation('relu'))
```

```
model.add(tf.keras.layers.MaxPooling1D())

model.add(tf.keras.layers.Conv1D(filters=64, kernel_size=3,
strides=1, padding='same',
activation='relu',kernel_initializer=tf.keras.initializers.ran
dom_normal()))
model.add(tf.keras.layers.BatchNormalization())
model.add(tf.keras.layers.Activation('relu'))

model.add(tf.keras.layers.Conv1D(filters=64, kernel_size=3,
strides=1, padding='same',
activation='relu',kernel_initializer=tf.keras.initializers.ran
dom_normal()))
model.add(tf.keras.layers.BatchNormalization())
model.add(tf.keras.layers.Activation('relu'))

model.add(tf.keras.layers.MaxPooling1D())

model.add(tf.keras.layers.Conv1D(filters=128, kernel_size=3,
strides=1, padding='same',
activation='relu',kernel_initializer=tf.keras.initializers.ran
dom_normal()))
model.add(tf.keras.layers.BatchNormalization())
model.add(tf.keras.layers.Activation('relu'))

model.add(tf.keras.layers.Conv1D(filters=128, kernel_size=3,
strides=1, padding='same',
activation='relu',kernel_initializer=tf.keras.initializers.ran
dom_normal()))
model.add(tf.keras.layers.BatchNormalization())
model.add(tf.keras.layers.Activation('relu'))

model.add(tf.keras.layers.Conv1D(filters=128, kernel_size=3,
strides=1, padding='same',
activation='relu',kernel_initializer=tf.keras.initializers.ran
dom_normal()))
model.add(tf.keras.layers.BatchNormalization())
model.add(tf.keras.layers.Activation('relu'))

model.add(tf.keras.layers.MaxPooling1D())

dd(tf.keras.layers.Conv1D(filters=128, kernel_size=3,
=1, padding='same',
ion='relu',kernel_initializer=tf.keras.initializers.ran
mal()))
dd(tf.keras.layers.BatchNormalization())
```

```
model.add(tf.keras.layers.Activation('relu'))

model.add(tf.keras.layers.Conv1D(filters=128, kernel_size=3,
strides=1, padding='same',
activation='relu',kernel_initializer=tf.keras.initializers.ran
dom_normal()))
model.add(tf.keras.layers.BatchNormalization())
model.add(tf.keras.layers.Activation('relu'))

model.add(tf.keras.layers.Conv1D(filters=128, kernel_size=3,
strides=1, padding='same',
activation='relu',kernel_initializer=tf.keras.initializers.ran
dom_normal()))
model.add(tf.keras.layers.BatchNormalization())
model.add(tf.keras.layers.Activation('relu'))

model.add(tf.keras.layers.MaxPooling1D())


model.add(tf.keras.layers.GlobalAveragePooling1D())

#dense layer
model.add(tf.keras.layers.Dense(100, activation='sigmoid',\
        kernel_regularizer=regularizers.l2(0.001),\
                                bias_regularizer=regularizers.
l2(0.001),))
model.add(tf.keras.layers.Dropout(0.5))
model.add(tf.keras.layers.Dense(50, activation='sigmoid',\
        kernel_regularizer=regularizers.l2(0.001),\
                                bias_regularizer=regularizers.
l2(0.001),))
model.add(tf.keras.layers.Dropout(0.5))
model.add(tf.keras.layers.Dense(2, activation='sigmoid',\
        kernel_regularizer=regularizers.l2(0.001),\
        bias_regularizer=regularizers.l2(0.001),))

opt = tf.keras.optimizers.Adam(
    learning_rate=0.0001, #default 0.001 #best record 0.001 @
500 epoch 81% accuracy
    beta_1=0.9,
    beta_2=0.999,
    epsilon=1e-07,
        grad=False,
        e="Adam",


        ummary()
```

```
model.compile(loss='binary_crossentropy',
              optimizer=opt,
              metrics=['accuracy'])
```

```
history = model.fit(X_train, y_train, epochs=100,
                    validation_data=(X_test, y_test))
test_loss, test_acc = model.evaluate(X_test, y_test)

print('Test Loss: {}'.format(test_loss))
print('Test Accuracy: {}'.format(test_acc))

def plot_graphs(history, string):
    plt.plot(history.history[string])
    plt.plot(history.history['val_'+string], '')
    plt.xlabel("Epochs")
    plt.ylabel(string)
    plt.legend([string, 'val_'+string])
    plt.show()

plot_graphs(history, 'accuracy')
plot_graphs(history, 'loss')

model.save("CNN_feature_extractor")
```

Feature Extraction CNN

```
from keras.models import Model, Sequential

#ambil dense layer pertama buat layer ekstraksi fitur
layer_name = 'dense' #<< ganti namanya sesuai layer dense
pertama
FC_layer_model = Model(inputs=model.input,
                       outputs=model.get_layer(layer_name).out
put)
FC_layer_model.summary()
```

```
i=0
number_of_feature = 100
features=np.zeros(shape=(X_train.shape[0],number_of_feature))
print('X_train shape:',X_train.shape)
    traksi fitur
    ut = FC_layer_model.predict(X_train)
    s=FC_output
    eature
```

```
np.save('features',features)

features
```

```
#bikin kolom buat fitur biar gampang saya liat
feature_col=[]
for i in range(number_of_feature):
    feature_col.append("f_"+str(i))
    i+=1

#feature_col

#Bikin dataframe supaya bisa dimasukkan ke random forest
train_features=pd.DataFrame(data=features,columns=feature_col)
feature_col = np.array(feature_col)
feature_col
```

```
train_class = np.array(y_train)
print('Training Features Shape:', train_features.shape)
print('Training Labels Shape:', y_train.shape)
train_class
```

8. RANDOM FOREST CLASSIFIER: MULTI-CLASS MULTI-OUTPUT

```
from sklearn.preprocessing import LabelBinarizer
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.multioutput import MultiOutputClassifier

forest = RandomForestClassifier(n_estimators = 20,
random_state = 42,)
rfmcmo = MultiOutputClassifier(forest, n_jobs=-1)

rfmcmo.fit(train_features, y_train).predict(train_features)
```

9. CNN FEATURE EXTRACTION + RANDOM FOREST CLASSIFIER

```
#data untuk validasi
#i   ta tweet yang sudah di tokenize dan serialisasi. ini
     suk ke CNN & RF


 untuk validasi
```

```
# Indexnya tidak berurut karena fungsi split train test di
atas
# ada data yang dobel karena hasil oversampling
# y_test
y_test
```

```
#tweet Test
#di bikinkan variable baru supaya bisa keliatan datanya karena
di atas sebelumnya sudah di padding dan tokenize. tapi masih
ada indexnya di y_test
# tabel ini cuma untuk evaluasi saja, biar gampang dibaca.

tweet_test = pd.merge(y_test, df_preprocessed['Tweet'],
left_index=True, right_index=True, how='left')
tweet_test =
tweet_test[~tweet_test.index.duplicated(keep='first')]
tweet_test
```

```
i=0
number_of_feature = 100
features=np.zeros(shape=(X_test.shape[0],number_of_feature))
print('X_train shape:',X_test.shape)
#test etraksi fitur
FC_output = FC_layer_model.predict(X_test)
features_test=FC_output

features_test
```

```
#Bikin dataframe supaya bisa dikasi masuk ke random forest
test_features=pd.DataFrame(data=features_test,columns=feature_
col)
feature_col = np.array(feature_col)

print('Test Features Shape:', test_features.shape)
print('Test Labels Shape:', y_test.shape)
```

```
test_features
```

```
#masukkan fitur dari CNN ke random forest
predictions = rfmcmo.predict(test_features)
```

```
dfpredict = pd.DataFrame(predictions, columns = ['prediction
'prediction sentiment'])

ct.to_csv('hasil prediksi.csv')
to_csv('test true.csv')
```

```
ac_pred = predictions[:,0]
ac_y_test = y_test['Dinas']
accuracy_dinas=accuracy_score(ac_pred , ac_y_test)
ac_pred = predictions[:,1]
ac_y_test = y_test['Sentiment']
accuracy_sentiment=accuracy_score(ac_pred , ac_y_test)
print('Accuracy Dinas: %f'% accuracy_dinas)
print('Accuracy Sentiment: %f'% accuracy_sentiment)
```

```
#f1 score
from sklearn import metrics
y_pred_dinas = predictions[:,0]
y_pred_sentiment = predictions[:,1]
y_true_dinas = y_test['Dinas']
y_true_sentiment = y_test['Sentiment']

# Print confusion matrix & presicion recal
print("================= confusion matrix dinas
================")
print(metrics.confusion_matrix(y_true_dinas, y_pred_dinas))
print(metrics.classification_report(y_true_dinas,
y_pred_dinas, digits=3))

print("============================================================
==")

print("================= confusion matrix Sentiment
============")
print(metrics.confusion_matrix(y_true_sentiment,
y_pred_sentiment))
print(metrics.classification_report(y_true_sentiment,
y_pred_sentiment, digits=3))
print("============================================================
==")

test_true = pd.read_csv("test true.csv")
test_true.rename(columns={"Unnamed: 0": "id"}, inplace=True)
test_true
# df_preprocessed.drop(columns=df_preprocessed.columns[0],
inplace=True, axis=1)
# df_preprocessed =
df_preprocessed.astype({'Sentiment':'int'})
eprocessed = df_preprocessed.astype({'Dinas':'int'})
(df_preprocessed.head())
("------------")
(df_preprocessed["Dinas"].value_counts())
("------------")
```

```
# print(df_preprocessed["Sentiment"].value_counts())
# print("------------")
# list(LE.inverse_transform([0, 1, 2,3,4,5]))
```

```
test_predict = pd.read_csv("hasil prediksi.csv")
test_predict
```

```
#perbandingan prediksi dengan hasil prediksi
res = pd.concat([test_true, test_predict],
axis=1).drop(columns='Unnamed: 0')
res
df_cd = pd.merge(res, df_preprocessed, how='inner',left_on =
'id',right_index =
True).drop(columns=['Dinas_y','Sentiment_y'])
df_cd.to_csv('hasil prediksi test.csv')

df_cd
```