

DAFTAR PUSTAKA

- Ahlgren, M. (2023, June 20). *Twitter Statistics, Facts and Trends for 2023*.
<https://www.websiterating.com/research/twitter-statistics/>
- CNBC Indonesia, T. (2021, December 17). *Mengenal Apa Itu Metaverse dan Bagaimana Cara Kerjanya*.
<https://www.cnbcindonesia.com/tech/20211216163806-37-299867/mengenal-apa-itu-metaverse-dan-bagaimana-cara-kerjanya>
- Firdaus, R., Asror, I., & Herdiani, A. (2021). Lexicon-Based Sentiment Analysis of Indonesian Language Student Feedback Evaluation. *Indonesian Journal on Computing (Indo-JC)*, 1-12 Pages.
<https://doi.org/10.34818/INDOJC.2021.6.1.408>
- Han, J., Khamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques 3rd Edition. DATA MINING*.
- Kanstrén, T. (2020, September 12). *A Look at Precision, Recall, and F1-Score*.
<https://towardsdatascience.com/a-look-at-precision-recall-and-f1-score-36b5fd0dd3ec>
- Khazri, A. (2019, August 8). *Self Organizing Maps*.
<https://towardsdatascience.com/self-organizing-maps-1b7d2a84e065>
- Koto, F., & Rahmaningtyas, G. Y. (2017). Inset lexicon: Evaluation of a word list for Indonesian sentiment analysis in microblogs. *2017 International Conference on Asian Language Processing (IALP)*, 391–394.
<https://doi.org/10.1109/IALP.2017.8300625>
- Kwak, H., Lee, C., Park, H., & Moon, S. (2010). What is Twitter, a social network or a news media? *Proceedings of the 19th International Conference on World Wide Web*, 591–600. <https://doi.org/10.1145/1772690.1772751>
- Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., & Liu, H. (2018). Feature Selection: A Data Perspective. *ACM Computing Surveys*, 50(6), 1–45. <https://doi.org/10.1145/3136625>
- Maharani, A. S. A. (2021, December 9). *Korsel Bangun Metaverse Seoul, Bakal Jadi Kota Digital Pertama di Dunia?*
<https://www.kompas.com/properti/read/2021/12/09/063000321/korsel-bangun-metaverse-seoul-bakal-jadi-kota-digital-pertama-di-dunia->
- Markoulidakis, I., Kopsiaftis, G., Rallis, I., & Georgoulas, I. (2021). Multi-Class Confusion Matrix Reduction method and its application on Net Promoter Score classification problem. *The 14th Pervasive Technologies Related to Assistive Environments Conference*, 412–419.
<https://doi.org/10.1145/3453892.3461323>

- Merinda Lestandy, Abdurrahim Abdurrahim, & Lailis Syafa'ah. (2021). Analisis Sentimen Tweet Vaksin COVID-19 Menggunakan Recurrent Neural Network dan Naïve Bayes. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 5(4), 802–808. <https://doi.org/10.29207/resti.v5i4.3308>
- Molina-Coronado, B., Mori, U., Mendiburu, A., & Miguel-Alonso, J. (2020). Survey of Network Intrusion Detection Methods From the Perspective of the Knowledge Discovery in Databases Process. *IEEE Transactions on Network and Service Management*, 17(4), 2451–2479. <https://doi.org/10.1109/TNSM.2020.3016246>
- Mystakidis, S. (2022). Metaverse. *Encyclopedia*, 2(1), 486–497. <https://doi.org/10.3390/encyclopedia2010031>
- Nugroho, K. S. (2019, November 13). *Confusion Matrix untuk Evaluasi Model pada Supervised Learning*. <https://ksnugroho.medium.com/confusion-matrix-untuk-evaluasi-model-pada-unsupervised-machine-learning-bc4b1ae9ae3f>
- Pane, S. F., & Ramdan, J. (2022). Pemodelan Machine Learning: Analisis Sentimen Masyarakat Terhadap Kebijakan PPKM Menggunakan Data Twitter. *Jurnal Sistem Cerdas*, 5(1), 12–20. <https://doi.org/10.37396/jsc.v5i1.191>
- Prattasha, N. J., Sami, A. A., Kowsher, M., Murad, S. A., Bairagi, A. K., Masud, M., & Baz, M. (2022). Transfer Learning for Sentiment Analysis Using BERT Based Supervised Fine-Tuning. *Sensors*, 22(11), 4157. <https://doi.org/10.3390/s22114157>
- Setiawan, S. (2020, July 12). *Membicarakan Precision, Recall, dan F1-Score*. <https://stevkarta.medium.com/membicarakan-precision-recall-dan-f1-score-e96d81910354>
- Undap, M. G., Rantung, V. P., & Rompas, P. T. D. (2021). Analisis Sentimen Situs Pembajak Artikel Penelitian Menggunakan Metode Lexicon-Based. 02(02).

LAMPIRAN

Lampiran 1 Dataset

https://drive.google.com/file/d/1r3_r1A-j1C9u-glcHNSfipRUUT_0cDYx/view?usp=drive_link

Lampiran 2 Stopword

adalah	bekerja	bulan	diminta
adapun	belakangan	cara	dimisalkan
agakny	belumlah	cukup	dimulailah
akan	benarkah	cukuplah	dimungkink
akhir	berada	dahulu	an
akhirnya	berakhirlah	dan	dipastikan
akulah	berapa	dari	diperbuatny
amatlah	berapalah	datang	a
andalah	berarti	demi	diperkirakan
antara	berbagai	demikianlah	diperlukan
apa	beri	depan	dipersoalka
apabila	berikut	dia	n
apalagi	berjumlah	diakhirinya	dipunyai
artinya	berkata	diantara	dirinya
asalkan	berkeingina	diberi	disebut
atau	n	diberikanny	disebutkann
ataupun	berlainan	a	ya
awalnya	berlangsung	dibuatnya	disinilah
bagaikan	bermacam	didatangkan	ditandakan
bagaimanak	bermaksud	diibaratkan	ditanyai
ah	bersama	diingat	ditegaskan
bagi	bersiap	diinginkan	ditunjuk
bahkan	bertanya	dijelaskan	ditunjukkan
bahwasanya	berturut	dikarenakan	ditunjuknya
bakal	bertutur	dikatakanny	dituturkann
balik	berupa	a	ya
bapak	betul	diketahui	diucapkann
bawah	biasa	dikira	ya
begini	bila	dilalui	dong
beginikah	bisa	dimaksud	dulu
begitu	boleh	dimaksudka	enggak
begitulah	bolehlah	nnya	entah

guna	kata	masih	mengibaratk
hal	katakanlah	masing	annya
hanya	ke	mau	mengingat
hari	kebetulan	melainkan	an
haruslah	kedua	melalui	mengira
hendak	keinginan	melihatnya	mengucapka
hendaknya	kelihatan	memastikan	nya
ia	kelima	memberikan	menjadi
ibarat	kembali	memerlukan	menjelaskan
ibaratnya	kemungkina	meminta	menunjuk
ikut	n	memisalkan	menunjukka
ingat-ingat	kenapa	mempergun	n
inginkah	kepadanya	akan	menurut
ini	keseluruhan	memperlihat	menyampai
inilah	khususnya	kan	kan
itukah	kinilah	mempunyai	menyatakan
jadi	kira-kira	memungkin	menyeluruh
jadinya	kita	kan	merasa
jauh	kok	menambahk	merekalah
jawaban	lagi	an	meski
jelas	lah	menanti	meyakini
jelaslah	lainnya	menantikan	minta
jika	lama	menanyai	misal
juga	lanjut	mendapat	misalnya
jumlahnya	lebih	mendatang	mulai
kala	lima	mendatangk	mulanya
kalaulah	macam	an	mungkinkah
kalian	makanya	mengakhiri	naik
kamilah	malah	mengatakan	nanti
kamulah	mampu	mengenai	nyaris
kapan	mana	mengetahui	oleh
kapanpun	manalagi	menghenda	pada
karenanya	masalah	ki	padanya

paling	sebesar	semua	tanpa
pasti	sebisanya	semula	tanyakan
penting	sebut	sendirian	tapi
per	sebutnya	seolah	tegasnya
perlunya	secukupnya	seorang	tempat
pertama-	sedangkan	sepantasnya	tentang
tama	sedikit	seperlunya	tentulah
pertanyakan	segalanya	sepertinya	tepat
pihaknya	seharusnya	sering	terasa
pula	seingat	serta	terdahulu
punya	sejauh	sesaat	terdiri
rasanya	sejumlah	sesampai	terhadapnya
rupanya	sekadarnya	sesekali	teringat-
saatnya	sekali-kali	sesuatu	ingat
sajalah	sekaligus	sesudah	terjadilah
sama	sekarang	setelah	terkira
sambil	sekecil	setengah	terlebih
sampai-	sekiranya	setiap	termasuk
sampai	sekitarnya	setibanya	tersampaika
sana	sekurangnya	setidaknya	n
sangatlah	selain	seusai	tersebutlah
saya	selalu	siap	tertuju
se	selama-	siapakah	terutama
sebabnya	lamanya	sini	tetapi
sebagaiman	selanjutnya	soal	tiba
a	seluruhnya	suatu	tinggi
sebagian	semakin	sudahkah	tunjuk
sebaik-	semampuny	supaya	tutur
baiknya	a	tadinya	ucap
sebaliknya	semasih	tahun	ujar
sebegini	semata-mata	tambah	umum
sebelum	sementara	tampak	ungkap
sebenarnya	semisalnya	tandas	untuk

usai	hanyalah	saling	tentu
wah	harusnya	sampai	tentunya
waktu	hendaklah	sebab	tertentu
walau	itu	secara	tidakkah
wong	itulah	sejak	yaitu
yakin	ingin	sendiri	waduh
yang	inginkan	seperti	wahai
ada	jangan	seseorang	wakil
adanya	kalau	sebagainya	waktunya
anda	kali	sebanyak	warga
apakah	kamu	sebegitu	wib
atas	karena	sebetulnya	rt
akankah	kemudian	sedemikian	no
antar	ketika	sedikitnya	nyatanya
ataukah	kalaupun	sekali	olehnya
bagaimana	kapankah	seketika	padahal
baik	kiranya	selagi	pak
baru	langsung	semuanya	para
belum	luar	sendirinya	pentingnya
bagai	lamanya	sepanjang	percuma
beginilah	mengapa	sepantasnya	perlukah
bilakah	mereka	lah	pertama
buat	makin	serupa	pertanyaan
bukankah	manakala	sesama	pihak
contohnya	masihkah	sesuatunya	pukul
cuma	mungkinlah	sesudahnya	rasa
dapat	nah	seterusnya	rata
daripada	nantinya	sewaktu	saat
di	paparnya	tandasnya	sama-sama
dilakukan	pernah	terhadap	sampaikan
dialah	pun	tersebut	sangat
dini	pastilah	tetap	satu
harus	saja	tadi	sayalah

sebagai	setiba	tuturnya	hidup
sebaik	setidak-	ucapnya	senin
sebaiknya	tidaknya	ujarnya	maupun
sebelumnya	setinggi	umumnya	mantan
seberapa	siapa	ungkapnya	jenis
sebuah	siapapun	usah	juni
sebutlah	sinilah	waktunya	tinggal
sedang	soalnya	walaupun	asal
segala	sudah	yakni	sesuai
segera	sudahlah	dengan	berat
sehingga	tahu	orang	memberi
sejenak	tambahnya	bahwa	sabtu
sekadar	tampaknya	namun	mencari
sekalian	tanya	dua	ruang
sekalipun	tanyanya	kepada	biasanya
sekitar	tegas	lalu	berdasarkan
sekurang-	telah	lain	pekan
kurangnya	tengah	banyak	membawa
sela	terakhir	beberapa	tingkat
selaku	terbanyak	besar	dekat
selama	terdapat	merupakan	ketiga
selamanya	teringat	agar	ribu
seluruh	terjadi	persen	membantu
semacam	terjadinya	wib	khusus
semampu	terlalu	diri	ditemukan
semasa	terlihat	minggu	kegiatan
semata	ternyata	the	tampil
semisal	terus	selasa	bertemu
sempat	tiap	jumlah	justru
seolah-olah	tiba-tiba	kondisi	menyebutka
seringnya	tiga	hubungan	n
sesegera	toh	acara	milik
setempat	turut	masa	menjalani

sumber
upaya
mengambil
lewat
meningkatkan
an
kehidupan
penggunaan
menghadapi
aku
kami

beliau
lo
lu
kemana
gimana
kurang
kemarin
yg
ya
emang
gitu

amat
sekian
sekedar
&
telanjur
dimana
kah
kecuali
seraya
dsb
dll

dulunya
demikian
oh
tolong
bagaimanap
un
iya

Lampiran 3 Kamus Slang

1. @:di
 2. abis:habis
 3. ad:ada
 4. adlh:adalah
 5. afaik:as far as i know
 6. ahaha:haha
 7. aj:saja
 8. ajep-ajep:dunia gemerlap
 9. ak:saya
 10. akika:aku
 11. akkoh:aku
 12. akuwh:aku
 13. alay:norak
 14. alow:halo
 15. ambilin:ambilkan
 16. ancur:hancur
 17. anjrit:anjing
 18. anter:antar
 19. ap2:apa-apa
 20. apasih:apa sih
 21. apes:sial
 22. aps:apa
 23. aq:saya
 24. aquwh:aku
 25. asbun:asal bunyi
 26. aseekk:asyik
 27. asekk:asyik
 28. asem:asam
 29. aspal:asli tetapi palsu
 30. astul:asal tulis
 31. ato:atau
 32. au ah:tidak mau tahu
 33. awak:saya
 34. ay:sayang
 35. ayank:sayang
 36. b4:sebelum
 37. bakalan:akan
 38. bandes:bantuan desa
 39. bangedh:banget
 40. banpol:bantuan polisi
 41. banpur:bantuan tempur
 42. basbang:basi

43. bcanda:bercanda
 44. bdg:bandung
 45. begajulan:nakal
 46. beliin:belikan
 47. bencong:banci
 48. bentar:sebentar
 49. ber3:bertiga
 50. beresin:membereskan
 51. bete:bosan
 52. beud:banget
 53. bg:abang
 54. bgmn:bagaimana
 55. bgt:banget
 56. bijimane:bagaimana
 57. bintal:bimbingan mental
 58. bkl:akan
 59. bknya:bukannya
 60. blegug:bodoh
 61. blh:boleh
 62. bln:bulan
 63. blum:belum
 64. bnci:benci
 65. bnran:yang benar
 66. bodor:lucu
 67. bokap:ayah
 68. boker:buang air besar
 69. bokis:bohong
 70. boljug:boleh juga
 71. bonek:bocah nekat
 72. boyeh:boleh
 73. br:baru
 74. brg:bareng
 75. bro:saudara laki-laki
 76. bru:baru
 77. bs:bisa
 78. bsen:bosan
 79. bt:buat
 80. btw:ngomong-ngomong
 81. buaya:tidak setia
 82. bubbu:tidur
 83. bubu:tidur
 84. bumil:ibu hamil

85. bw:bawa
 86. bwt:buat
 87. byk:banyak
 88. byrin:bayarkan
 89. cabal:sabar
 90. cadas:keren
 91. calo:makelar
 92. can:belum
 93. capcus:pergi
 94. caper:cari perhatian
 95. ce:cewek
 96. cekal:cegah tangkal
 97. cemen:penakut
 98. cengengesan:tertawa
 99. cepet:cepat
 100. cew:cewek
 101. chuyunk:sayang
 102. cimeng:ganja
 103. cipika cipiki:cium pipi kanan
 cium pipi kiri
 104. ciyh:sih
 105. ckepp:cakep
 106. ckp:cakep
 107. cmiiw:correct me if i'm
 wrong
 108. cmpur:campur
 109. cong:banci
 110. conlok:cinta lokasi
 111. cowwyy:maaf
 112. cp:siapa
 113. cpe:capek
 114. cppe:capek
 115. cucok:cocok
 116. cuex:cuek
 117. cumi:Cuma miscall
 118. cups:culun
 119. curanmor:pencurian
 kendaraan bermotor
 120. curcol:curahan hati colongan
 121. cwek:cewek
 122. cyin:cinta
 123. d:di
 124. dah:deh

125. dapet:dapat
 126. de:adik
 127. dek:adik
 128. demen:suka
 129. deyh:deh
 130. dgn:dengan
 131. diancurin:dihancurkan
 132. dimaafin:dimaafkan
 133. dimintak:diminta
 134. disono:di sana
 135. dket:dekat
 136. dkk:dan kawan-kawan
 137. dll:dan lain-lain
 138. dlu:dulu
 139. dngn:dengan
 140. dodol:bodoh
 141. doku:uang
 142. dongs:dong
 143. dpt:dapat
 144. dri:dari
 145. drmn:darimana
 146. drtd:dari tadi
 147. dst:dan seterusnya
 148. dtg:datang
 149. duh:aduh
 150. duren:durian
 151. ed:edisi
 152. egp:emang gue pikirin
 153. eke:aku
 154. elu:kamu
 155. emangnya:memangnya
 156. emng:memang
 157. endak:tidak
 158. ga:tidak
 159. gk:tidak
 160. g:tidak
 161. enggak:tidak
 162. envy:iri
 163. ex:mantan
 164. fax:facsimile
 165. fifo:first in first out
 166. folbek:follow back
 167. fyi:sebagai informasi

168. gaada:tidak ada uang
 169. gag:tidak
 170. gaje:tidak jelas
 171. gak papa:tidak apa-apa
 172. gan:juragan
 173. gaptek:gagap teknologi
 174. gatek:gagap teknologi
 175. gawe:kerja
 176. gbs:tidak bisa
 177. gebetan:orang yang disuka
 178. geje:tidak jelas
 179. gepeng:gelandangan dan pengemis
 180. ghiy:lagi
 181. gile:gila
 182. gimana:bagaimana
 183. gino:gigi nongol
 184. githu:gitu
 185. gj:tidak jelas
 186. gmana:bagaimana
 187. gn:begini
 188. goblok:bodoh
 189. golput:golongan putih
 190. gowes:mengayuh sepeda
 191. gpny:tidak punya
 192. gr:gede rasa
 193. gretongan:gratisan
 194. gtau:tidak tahu
 195. gua:saya
 196. guoblok:goblok
 197. gw:saya
 198. ha:tertawa
 199. haha:tertawa
 200. hallow:halo
 201. hankam:pertahanan dan keamanan
 202. hehe:he
 203. helo:halo
 204. hey:hai
 205. hlm:halaman
 206. hny:hanya
 207. hoax:isu bohong
 208. hr:hari

209. hrus:harus
 210. hubdar:perhubungan darat
 211. huff:mengeluh
 212. hum:rumah
 213. humz:rumah
 214. ilang:hilang
 215. ilfil:tidak suka
 216. imho:in my humble opinion
 217. imoetz:imut
 218. item:hitam
 219. itungan:hitungan
 220. iye:iya
 221. ja:saja
 222. jadiin:jadi
 223. jaim:jaga image
 224. jayus:tidak lucu
 225. jdi:jadi
 226. jem:jam
 227. jga:juga
 228. jgnkan:jangan
 229. jir:anjing
 230. jln:jalan
 231. jomblo:tidak punya pacar
 232. jubir:juru bicara
 233. jutek:galak
 234. k:ke
 235. kab:kabupaten
 236. kabor:kabur
 237. kacrut:kacau
 238. kativ:kepala divisi
 239. kagak:tidak
 240. kalo:kalau
 241. kampret:sialan
 242. kamtibmas:keamanan dan ketertiban masyarakat
 243. kamuwh:kamu
 244. kanwil:kantor wilayah
 245. karna:karena
 246. kasubbag:kepala subbagian
 247. katrok:kampung
 248. kayanya:kayaknya
 249. kbr:kabar
 250. kdu:harus

251. kec:kecamatan
 252. kejurnas:kejuaraan nasional
 253. kekeuh:keras kepala
 254. kel:kelurahan
 255. kemaren:kemarin
 256. kepengen:mau
 257. kepingin:mau
 258. kepsek:kepala sekolah
 259. kesbang:kesatuan bangsa
 260. kesra:kesejahteraan rakyat
 261. ketrima:diterima
 262. kegiatan:kegiatan
 263. kibul:bohong
 264. kimpoi:kawin
 265. kl:kalau
 266. klianz:kalian
 267. kloter:kelompok terbang
 268. klw:kalau
 269. km:kamu
 270. kmpts:kampus
 271. kmrn:kemarin
 272. knal:kenal
 273. knp:kenapa
 274. kodya:kota madya
 275. komdis:komisi disiplin
 276. komsov:komunis sovyet
 277. kongkow:kumpul bareng
 teman-teman
 278. kopdar:kopi darat
 279. korup:korupsi
 280. kpn:kapan
 281. krenz:keren
 282. krm:kirim
 283. kt:kita
 284. ktmu:ketemu
 285. ktr:kantor
 286. kuper:kurang pergaulan
 287. kw:imitasi
 288. kyk:seperti
 289. la:lah
 290. lam:salam
 291. lamp:lampiran
 292. lanud:landasan udara

293. latgab:latihan gabungan
 294. lebay:berlebihan
 295. leh:boleh
 296. lelet:lambat
 297. lemot:lambat
 298. lgi:lagi
 299. lgsg:langsung
 300. liat:lihat
 301. litbang:penelitian dan
 pengembangan
 302. lmyan:lumayan
 303. lo:kamu
 304. loe:kamu
 305. lola:lambat berfikir
 306. loup:h:cinta
 307. low:kalau
 308. lp:lupa
 309. luber:langsung, umum, bebas,
 dan rahasia
 310. luchuw:lucu
 311. lum:belum
 312. luthu:lucu
 313. lwn:lawan
 314. maacih:terima kasih
 315. mabal:bolos
 316. macem:macam
 317. macih:masih
 318. maem:makan
 319. magabut:makan gaji buta
 320. maho:homo
 321. mak jang:kaget
 322. maksain:memaksa
 323. malem:malam
 324. mam:makan
 325. maneh:kamu
 326. maniez:manis
 327. mao:mau
 328. masukin:masukkan
 329. melu:ikut
 330. mepet:dekat sekali
 331. mgu:minggu
 332. migas:minyak dan gas bumi
 333. mikol:minuman beralkohol

334. miras:minuman keras
 335. mlah:malah
 336. mngkn:mungkin
 337. mo:mau
 338. mokad:mati
 339. moso:masa
 340. mpe:sampai
 341. msk:masuk
 342. mslh:masalah
 343. mt:makan teman
 344. mubes:musyawarah besar
 345. mulu:melulu
 346. mumpung:selagi
 347. munas:musyawarah nasional
 348. muntaber:muntah dan berak
 349. musti:mesti
 350. muupz:maaf
 351. mw:now watching
 352. n:dan
 353. nanam:menanam
 354. nanya:bertanya
 355. napa:kenapa
 356. napi:narapidana
 357. napza:narkotika, alkohol,
 psikotropika, dan zat adiktif
 358. narkoba:narkotika,
 psikotropika, dan obat terlarang
 359. nasgor:nasi goreng
 360. nda:tidak
 361. ndiri:sendiri
 362. ne:ini
 363. nekolin:neokolonialisme
 364. nembak:menyatakan cinta
 365. ngabuburit:menunggu
 berbuka puasa
 366. ngaku:mengaku
 367. ngambil:mengambil
 368. nganggur:tidak punya
 pekerjaan
 369. ngapah:kenapa
 370. ngaret:terlambat
 371. ngasih:memberikan
 372. ngebandel:berbuat bandel

373. ngegosip:bergosip
 374. ngeklaim:mengklaim
 375. ngeksis:menjadi eksis
 376. ngeles:berkilah
 377. ngelidur:menggigau
 378. ngerampok:merampok
 379. ngga:tidak
 380. ngibul:berbohong
 381. ngiler:mau
 382. ngiri:iri
 383. ngisiin:mengisikan
 384. ngmng:bicara
 385. ngomong:bicara
 386. ngubek2:mencari-cari
 387. ngurus:mengurus
 388. nie:ini
 389. nih:ini
 390. niyh:nih
 391. nmr:nomor
 392. nntn:nonton
 393. nobar:nonton bareng
 394. np:now playing
 395. ntar:nanti
 396. ntn:nonton
 397. numpuk:bertumpuk
 398. nutupin:menutupi
 399. nyari:mencari
 400. nyekar:menyekar
 401. nyicil:mencicil
 402. nyoblos:mencoblos
 403. nyokap:ibu
 404. ogah:tidak mau
 405. ol:online
 406. ongkir:ongkos kirim
 407. oot:out of topic
 408. org2:orang-orang
 409. ortu:orang tua
 410. otda:otonomi daerah
 411. otw:on the way, sedang di
 jalan
 412. pacal:pacar
 413. pake:pakai
 414. pala:kepala

415. pansus:panitia khusus
 416. parpol:partai politik
 417. pasutri:pasangan suami istri
 418. pd:pada
 419. pede:percaya diri
 420. pelatnas:pemusatan latihan nasional
 421. pemda:pemerintah daerah
 422. pemkot:pemerintah kota
 423. pemred:pemimpin redaksi
 424. penjas:pendidikan jasmani
 425. perda:peraturan daerah
 426. perhatiin:perhatikan
 427. pesenan:pesanan
 428. pgang:pegang
 429. pi:tapi
 430. pilkada:pemilihan kepala daerah
 431. pisan:sangat
 432. pk:penjahat kelamin
 433. plg:paling
 434. pmrnth:pemerintah
 435. polantas:polisi lalu lintas
 436. ponpes:pondok pesantren
 437. pp:pulang pergi
 438. prg:pergi
 439. prnh:pernah
 440. psen:pesan
 441. pst:pasti
 442. pswt:pesawat
 443. pw:posisi nyaman
 444. qmu:kamu
 445. rakor:rapat koordinasi
 446. ranmor:kendaraan bermotor
 447. re:reply
 448. ref:referensi
 449. rehab:rehabilitasi
 450. rempong:sulit
 451. repp:balas
 452. restik:reserse narkotika
 453. rhs:rahasia
 454. rmh:rumah
 455. ru:baru

456. ruko:rumah toko
 457. rusunawa:rumah susun sewa
 458. ruz:terus
 459. saia:saya
 460. salting:salah tingkah
 461. sampe:sampai
 462. samsek:sama sekali
 463. sapose:siapa
 464. satpam:satuan pengamanan
 465. sbb:sebagai berikut
 466. sbh:sebuah
 467. sbnrny:sebenarnya
 468. scr:secara
 469. sdgkn:sedangkan
 470. sdt:sedikit
 471. se7:setuju
 472. sebelas dua belas:mirip
 473. sembako:sembilan bahan pokok
 474. sempet:sempat
 475. sendratari:seni drama tari
 476. sgt:sangat
 477. shg:sehingga
 478. siech:sih
 479. sikon:situasi dan kondisi
 480. sinetron:sinema elektronik
 481. siramin:siramkan
 482. sj:saja
 483. skalian:sekalian
 484. sklh:sekolah
 485. skt:sakit
 486. slesai:selesai
 487. sll:selalu
 488. slma:selama
 489. slsai:selesai
 490. smpt:sempat
 491. smw:semua
 492. sndiri:sendiri
 493. soljum:sholat jumat
 494. songong:sombong
 495. sory:maaf
 496. sosek:sosial-ekonomi
 497. sotoy:sok tahu

498. spa:siapa
 499. sppa:siapa
 500. spt:seperti
 501. srtfkt:sertifikat
 502. stiap:setiap
 503. stlh:setelah
 504. suk:masuk
 505. sumpek:sempit
 506. syg:sayang
 507. t4:tempat
 508. tajir:kaya
 509. tau:tahu
 510. taw:tahu
 511. td:tadi
 512. tdk:tidak
 513. teh:kakak perempuan
 514. telat:terlambat
 515. telmi:telat berpikir
 516. temen:teman
 517. tengil:menyebalkan
 518. tepar:terkapar
 519. tggu:tunggu
 520. tgu:tunggu
 521. thankz:terima kasih
 522. thn:tahun
 523. tilang:bukti pelanggaran
 524. tipiwon:TV One
 525. tks:terima kasih
 526. tlp:telepon
 527. tls:tulis
 528. tmbah:tambah
 529. tmen2:teman-teman
 530. tmpah:tumpah
 531. tmpt:tempat
 532. tngu:tunggu
 533. tnyta:ternyata
 534. tokai:tai
 535. toserba:toko serba ada
 536. tpi:tapi
 537. trdhulu:terdahulu
 538. trima:terima kasih
 539. trm:terima
 540. trs:terus

541. trutama:terutama
 542. ts:penulis
 543. tst:tahu sama tahu
 544. ttg:tentang
 545. tuch:tuh
 546. tuir:tua
 547. tw:tahu
 548. u:kamu
 549. ud:sudah
 550. udah:sudah
 551. ujj:ujung
 552. ul:ulangan
 553. unyu:lucu
 554. uplot:unggah
 555. urang:saya
 556. usah:perlu
 557. utk:untuk
 558. valas:valuta asing
 559. w/:dengan
 560. wadir:wakil direktur
 561. wamil:wajib militer
 562. warkop:warung kopi
 563. warteg:warung tegal
 564. wat:buat
 565. wkt:waktu
 566. wtf:what the fuck
 567. xixixi:tertawa
 568. ya:iya
 569. yap:iya
 570. yaudah:ya sudah
 571. yawdah:ya sudah
 572. yg:yang
 573. yl:yang lain
 574. yo:iya
 575. yowes:ya sudah
 576. yup:iya
 577. 7an:tujuan
 578. ababil:abg labil
 579. acc:accord
 580. adlah:adalah
 581. adoh:aduh
 582. aha:tertawa
 583. aing:saya

584. aja:saja
 585. aji:saja
 586. aka:dikenal juga sebagai
 587. akko:aku
 588. akku:aku
 589. akyu:aku
 590. aljasa:asal jadi saja
 591. ama:sama
 592. ambl:ambil
 593. anjir:anjing
 594. ank:anak
 595. ap:apa
 596. apaan:apa
 597. ape:apa
 598. aplot:ungguh
 599. apva:apa
 600. aqu:aku
 601. asap:sesegera mungkin
 602. aseek:asyik
 603. asek:asyik
 604. aseknnya:asyiknya
 605. asoy:asyik
 606. astrojim:astagfirullahaladzim
 607. ath:kalau begitu
 608. atuh:kalau begitu
 609. ava:avatar
 610. aws:awas
 611. ayang:sayang
 612. ayok:ayo
 613. bacot:banyak bicara
 614. bales:balas
 615. bangdes:pembangunan desa
 616. bangkotan:tua
 617. banpres:bantuan presiden
 618. bansarkas:bantuan sarana kesehatan
 619. basis:badan amal, zakat, infak, dan sedekah
 620. bcoz:karena
 621. beb:sayang
 622. bejibun:banyak
 623. belum:belum
 624. bener:benar

625. ber2:berdua
 626. berdikari:berdiri di atas kaki sendiri
 627. bet:banget
 628. beti:beda tipis
 629. beut:banget
 630. bgd:banget
 631. bgs:bagus
 632. bhubu:tidur
 633. bimbuluh:bimbingan dan penyuluhan
 634. bisi:kalau-kalau
 635. bkn:bukan
 636. bl:beli
 637. blg:bilang
 638. blm:belum
 639. bls:balas
 640. bnchi:benci
 641. bngung:bingung
 642. bnyk:banyak
 643. bohay:badan aduhai
 644. bokep:porno
 645. bokin:pacar
 646. bole:boleh
 647. bolot:bodoh
 648. bonyok:ayah ibu
 649. bpk:bapak
 650. brb:segera kembali
 651. brngkt:berangkat
 652. brp:berapa
 653. brur:saudara laki-laki
 654. bsa:bisa
 655. bsk:besok
 656. bu_bu:tidur
 657. bubarin:bubarkan
 658. buber:buka bersama
 659. bujubune:luar biasa
 660. buser:buru sergap
 661. bwhn:bawahan
 662. byar:bayar
 663. byr:bayar
 664. c8:chat
 665. cabut:pergi

666. caem:cakep
 667. cama-cama:sama-sama
 668. cangcut:celana dalam
 669. cape:capek
 670. caur:jelek
 671. cekak:tidak ada uang
 672. cekidot:coba lihat
 673. cemplungin:ceplungkan
 674. ceper:pendek
 675. ceu:kakak perempuan
 676. cewe:cewek
 677. cibuk:sibuk
 678. cin:cinta
 679. ciye:cie
 680. ckck:ck
 681. clbk:cinta lama bersemi kembali
 682. cmpr:campur
 683. cnenk:senang
 684. congor:mulut
 685. cow:cowok
 686. coz:karena
 687. cpa:siapa
 688. gokil:gila
 689. gombal:suka merayu
 690. gpl:tidak pakai lama
 691. gpp:tidak apa-apa
 692. gretong:gratis
 693. gt:begitu
 694. gtw:tidak tahu
 695. gue:saya
 696. guys:teman-teman
 697. gws:cepat sembuh
 698. haghaghag:tertawa
 699. hakhak:tertawa
 700. handak:bahan peledak
 701. hansip:pertahanan sipil
 702. hellow:halo
 703. helow:halo
 704. hi:hai
 705. hlng:hilang
 706. hnya:hanya
 707. houm:rumah

708. hrs:harus
 709. hubad:hubungan angkatan darat
 710. hubla:perhubungan laut
 711. huft:mengeluh
 712. humas:hubungan masyarakat
 713. idk:saya tidak tahu
 714. ilfeel:tidak suka
 715. imba:jago sekali
 716. imoet:imut
 717. info:informasi
 718. itung:hitung
 719. isengin:bercanda
 720. iyala:iya lah
 721. iyo:iya
 722. jablay:jarang dibelai
 723. jadul:jaman dulu
 724. jancuk:anjing
 725. jd:jadi
 726. jdikan:jadikan
 727. jg:juga
 728. jgn:jangan
 729. jijay:jijik
 730. jkt:jakarta
 731. jnj:janji
 732. jth:jatuh
 733. jurdil:jujur adil
 734. jwb:jawab
 735. ka:kakak
 736. kabag:kepala bagian
 737. kacian:kasihan
 738. kadit:kepala direktorat
 739. kaga:tidak
 740. kaka:kakak
 741. kamtib:keamanan dan ketertiban
 742. kamuh:kamu
 743. kamyu:kamu
 744. kapt:kapten
 745. kasat:kepala satuan
 746. kasubbid:kepala subbidang
 747. kau:kamu
 748. kbar:kabar

749. kcian:kasih
 750. keburu:terlanjur
 751. kedubes:kedutaan besar
 752. kek:seperti
 753. keknya:kayaknya
 754. keliatan:kelihatan
 755. keneh:masih
 756. kepikiran:terpikirkan
 757. kepo:mau tahu urusan orang
 758. kere:tidak punya uang
 759. kesian:kasih
 760. ketauan:ketahuan
 761. keukeuh:keras kepala
 762. khan:kan
 763. kibus:kaki busuk
 764. kk:kartu keluarga
 765. klian:kalian
 766. klo:kalau
 767. kluarga:keluarga
 768. klwrga:keluarga
 769. kmari:kemari
 770. kmpus:kampus
 771. kn:kan
 772. knl:kenal
 773. knpa:kenapa
 774. kog:kok
 775. kompi:komputer
 776. komtiong:komunis Tiongkok
 777. konjen:konsulat jenderal
 778. koq:kok
 779. kpd:kepada
 780. kptsan:keputusan
 781. krik:garing
 782. krn:karena
 783. ktauan:ketahuan
 784. ktny:katanya
 785. kudu:harus
 786. kuq:kok
 787. ky:seperti
 788. kykny:kayanya
 789. laka:kecelakaan
 790. lambreta:lambat
 791. lansia:lanjut usia

792. lapas:lembaga
 pemasyarakatan
 793. libur:libur
 794. lekong:laki-laki
 795. lg:lagi
 796. lgkp:lengkap
 797. lht:lihat
 798. linmas:perlindungan
 masyarakat
 799. lmyan:lumayan
 800. lngkp:lengkap
 801. loch:loh
 802. lol:tertawa
 803. lom:belum
 804. loupz:cinta
 805. lowh:kamu
 806. lu:kamu
 807. luchu:lucu
 808. luff:cinta
 809. luph:cinta
 810. lw:kamu
 811. lwt:lewat
 812. maaciw:terima kasih
 813. mabes:markas besar
 814. macem-macem:macam-
 macam
 815. madesu:masa depan suram
 816. maen:main
 817. mahatma:maju sehat bersama
 818. mak:ibu
 819. makasih:terima kasih
 820. malah:bahkan
 821. malu2in:memalukan
 822. mamz:makan
 823. manies:manis
 824. mantep:mantap
 825. markus:makelar kasus
 826. mba:mbak
 827. mending:lebih baik
 828. mgkn:mungkin
 829. mhn:mohon
 830. miker:minuman keras
 831. milis:mailing list

832. mksd:maksud
 833. mls:malas
 834. mnt:minta
 835. moge:motor gede
 836. mokat:mati
 837. mosok:masa
 838. msh:masih
 839. mskpn:meskipun
 840. msng2:masing-masing
 841. muahal:mahal
 842. muker:musyawarah kerja
 843. mumet:pusing
 844. muna:munafik
 845. munaslub:musyawarah nasional luar biasa
 846. musda:musyawarah daerah
 847. muup:maaf
 848. muuy:maaf
 849. nal:kenal
 850. nangis:menangis
 851. naon:apa
 852. napol:narapidana politik
 853. naq:anak
 854. narsis:bangga pada diri sendiri
 855. nax:anak
 856. ndak:tidak
 857. ndut:gendut
 858. nekolim:neokolonialisme
 859. nelfon:menelepon
 860. ngabis2in:menghabiskan
 861. ngakak:tertawa
 862. ngambek:marah
 863. ngampus:pergi ke kampus
 864. ngantri:mengantri
 865. ngapain:sedang apa
 866. ngaruh:berpengaruh
 867. ngawur:berbicara sembarangan
 868. ngeceng:kumpul bareng-bareng
 869. ngeh:sadar
 870. ngekos:tinggal di kos

871. ngelamar:melamar
 872. ngeliat:melihat
 873. ngemeng:bicara terus-terusan
 874. ngerti:mengerti
 875. nggak:tidak
 876. ngikut:ikut
 877. nginep:menginap
 878. ngisi:mengisi
 879. ngmg:bicara
 880. ngocol:lucu
 881. ngomongin:membicarakan
 882. ngumpul:berkumpul
 883. ni:ini
 884. nyasar:tersesat
 885. nyariin:mencari
 886. nyiapin:mempersiapkan
 887. nyiram:menyiram
 888. nyok:ayo
 889. o/:oleh
 890. ok:ok
 891. priksa:periksa
 892. pro:profesional
 893. psn:pesan
 894. psti:pasti
 895. puanas:panas
 896. qmo:kamu
 897. qt:kita
 898. rame:ramai
 899. raskin:rakyat miskin
 900. red:redaksi
 901. reg:register
 902. rejeki:rezeki
 903. renstra:rencana strategis
 904. reskrim:reserse kriminal
 905. sni:sini
 906. somse:sombong sekali
 907. sorry:maaf
 908. sosbud:sosial-budaya
 909. sospol:sosial-politik
 910. sowry:maaf
 911. spd:sepeda
 912. sprti:seperti
 913. spy:supaya

914. stelah:setelah
 915. subbag:subbagian
 916. sumbangan:sumbangan
 917. sy:saya
 918. syp:siapa
 919. tabanas:tabungan
 pembangunan nasional
 920. tar:nanti
 921. taun:tahun
 922. tawh:tahu
 923. tdi:tadi
 924. te2p:tetap
 925. tekor:rugi
 926. telkom:telekomunikasi
 927. telp:telepon
 928. temen2:teman-teman
 929. tengok:menjenguk
 930. terbitin:terbitkan
 931. tgl:tanggal
 932. thanks:terima kasih
 933. thd:terhadap
 934. thx:terima kasih
 935. tipi:TV
 936. tkg:tukang
 937. tll:terlalu
 938. tlpn:telepon
 939. tman:teman
 940. tmbh:tambah
 941. tmn2:teman-teman
 942. tmph:tumpah
 943. tnda:tanda
 944. tnh:tanah
 945. togel:toto gelap
 946. tp:tapi
 947. tq:terima kasih
 948. trgntg:tergantung
 949. trims:terima kasih
 950. cb:coba
 951. y:ya
 952. munfik:munafik
 953. reklamuk:reklamasi
 954. sma:sama
 955. tren:trend

956. ngehe:kesal
 957. mz:mas
 958. analisis:analisis
 959. sadaar:sadar
 960. sept:september
 961. nmenarik:menarik
 962. zonk:bodoh
 963. rights:benar
 964. simiskin:miskin
 965. ngumpet:sembunyi
 966. hardcore:keras
 967. akhirx:akhirnya
 968. solve:solusi
 969. watuk:batuk
 970. ngebully:intimidasi
 971. masy:masyarakat
 972. still:masih
 973. tauk:tahu
 974. mbual:bual
 975. tioghoa:tionghoa
 976. ngentotin:senggama
 977. kentot:senggama
 978. faktakta:fakta
 979. sohib:teman
 980. rubahnn:rubah
 981. trlalu:terlalu
 982. nyela:cela
 983. heters:pembenci
 984. nyembah:sembah
 985. most:paling
 986. ikon:lambang
 987. light:terang
 988. pndukung:pendukung
 989. setting:atur
 990. seting:akting
 991. next:lanjut
 992. waspadalah:waspada
 993. gantengsaya:ganteng
 994. parte:partai
 995. nyerang:serang
 996. nipu:tipu
 997. ktipu:tipu
 998. jentelmen:berani

999. buangbuang:buang
1000. tsangka:tersangka
1001. kurng:kurang
1002. ista:nista
1003. less:kurang
1004. koar:teriak
1005. paranoid:takut
1006. problem:masalah
1007. tahi:kotoran
1008. tirani:tiran
1009. tilep:tilap
1010. happy:bahagia
1011. tak:tidak
1012. penertiban:tertib
1013. uasai:kuasa
1014. mnolak:tolak
1015. trending:trend
1016. taik:tahi
1017. wkwkkw:tertawa

1018. ahokncc:ahok
1019. istaa:nista
1020. benarjujur:jujur
1021. mgkin:mungkin
1022. gak:tidak
1023. trus:lalu
1024. dlm:dalam
1025. tetep:tetap
1026. skrg:sekarang
1027. sm:dengan
1028. udh:sudah
1029. cm:cuma
1030. org:orang
1031. bangor:nakal
1032. ngamuk:mengamuk
1033. iso:bisa
1034. mbuh:tidak tahu
1035. kzl:kesal
1036. name:nama

Lampiran 4 Kamus Lexicon

<https://drive.google.com/file/d/1-dGJG64QpH39opo6WcZ2L5pGYRC0z3-I/view?usp=sharing>

Lampiran 5 Source Code Scraping Dataset

```
import snsrape.modules.twitter as sntwitter
import pandas as pd

# Creating list to append tweet data to
tweets_list2 = []

# Using TwitterSearchScraprer to scrape data and append tweets to
list
query = 'id'
for i,tweet in enumerate(sntwitter.TwitterSearchScraprer(
    'metaverse since:2021-10-01 until:2022-12-31 lang:id -
filter:retweets ').get_items()):
    if i>20000:
        break
    tweets_list2.append([tweet.date, tweet.id, tweet.content,
tweet.user.username, tweet.user.location])

# Creating a dataframe from the tweets list above
tweets_df2 = pd.DataFrame(tweets_list2, columns=['Datetime',
'Tweet Id', 'Text', 'Username', 'Location'])
tweets_df2.to_csv("path.csv")
```

Lampiran 6 Source Code Preprocessing

a. Data Integration

```

df = pd.read_csv('path.csv', encoding='utf-8')
jakarta = pytz.timezone('Asia/Jakarta')
time_date = "%m/%d/%Y %H:%M"
df['datetime_created'] = df['Datetime'].apply(lambda
x:datetime.strptime(x,time_date))
df['date_created'] = df['datetime_created'].apply(lambda
x:x.date())
df['time_created'] = df['datetime_created'].apply(lambda
x:x.time())
df = df.drop(['datetime_created'],axis=1)

#emoticon senang
emoticon_senang = set([
    ':-)', ':)', ';)', ':o)', ':]', ':3', ':c)', ':>', '=]',
'8)', '=)', ':}',
    ':^)', ':-D', ':D', '8-D', '8D', 'x-D', 'xD', 'X-D', 'XD',
'=-D', '=D',
    '=-3', '=3', ':-))', ":'-)", ":')", ":'*", ":'^*", ":'>:P', ":'-
P', ":'P', 'X-P',
    'x-p', 'xp', 'XP', ":'-p', ":'p', '=p', ":'-b', ":'b', ":'>:)",
'>:)', ":'>:-)",
    '<3'
])

#emoticon sedih
emoticon_sedih = set([
    ':L', ":'-/', ":'>:/", ":'S', ":'>:[", ":'@', ":'-((", ":':[", ":'-||",
'=L', ":'<',
    ":'-["', ":'-<', ":'=\\", ":'=/', ":'>:((", ":':((", ":'>.<', ":'-'(",
":'('", ":'\\\", ":'-c',
    ':c', ":'{'', ":'>:\\\", ":';(('
])

#emoji patterns
emoji_pattern = re.compile("[
                                u"\U0001F600-\U0001F64F" #
emoticons
                                u"\U0001F300-\U0001F5FF" # symbols
& pictographs

```

```

        u"\U0001F680-\U0001F6FF" #
transport & map symbols
        u"\U0001F1E0-\U0001F1FF" # flags
(iOS)
        u"\U00002702-\U000027B0"
        u"\U000024C2-\U0001F251"
        "]" + ", flags=re.UNICODE)

#gabung emoticon senang dan sedih
emoticons = emoticon_senang.union(emoticon_sedih)

my_file = open('cleaning_source/combined_stop_words.txt','r')
content = my_file.read()
stop_words = content.split('\n')
file_2 =
open('cleaning_source/update_combined_slang_words.txt','r')
content2 = file_2.read()
slang_words = ast.literal_eval(content2)
file_3 = open('cleaning_source/indonesia_dictionary.txt', 'r',
encoding="utf-8")
content3 = file_3.read()
indonesia_dictionary = content3.split('\n')
my_file.close()
file_2.close()
file_3.close()

```

b. Clean Text

```

def clean_text(text):
    # Remove url
    text = re.sub(r'https?:\/\/[^\s]+', '', text)
    # Remove hashtag
    text = re.sub(r'#\w+', '', text)
    # Remove mentions
    text = re.sub(r'@\w+', '', text)

    # Remove word that containing number
    text = re.sub(r'\b\w*\d\w*\b', '', text)

    text = re.sub(r':', '', text)
    text = re.sub(r',', '', text)

```

```

#replace consecutive non-ASCII characters with a space
text = re.sub(r'^[\x00-\x7F]+',' ', text)

#remove emojis from text
text = emoji_pattern.sub(r'',text)

#remove punctuation
text = re.sub('[^a-zA-Z]',' ', text)

#remove tags
text=re.sub('&lt;/?.*?&gt;","&lt;&gt;','',text)

#remove digits and special chars
text = re.sub("(\d|\W)+"," ",text)

#remove other symbol from tweet
text = re.sub(r'â', '', text)
text = re.sub(r'€', '', text)
text = re.sub(r'|', '', text)
text = text.lower()

return text

```

c. Normalisasi Text Slang

```

word_tokens = word_tokenize(text)

for w in word_tokens:
    if w in slang_words.keys():
        word_tokens[word_tokens.index(w)] = slang_words[w]

```

d. Stopwords dan Stemming

```

filtered_tweet = [w for w in word_tokens if w not in stop_words]
filtered_tweet = []

for w in word_tokens:
    #check tokens pada emoticons, punctuations dan stopwords

```

```

        if w not in emoticons and w not in string.punctuation
            and w not in stop_words:
                filtered_tweet.append(w.lower())

    #stem kata
    filtered_tweet = [stemmer.stem(word) for word in
                      filtered_tweet]
    return ' '.join(filtered_tweet)

```

e. Labeling

```

import pandas as pd
import numpy as np
from transformers import pipeline

df = pd.read_csv("path.csv", encoding='UTF-8')

pretrained_name = "w11wo/indonesian-roberta-base-sentiment-
classifier"

nlp = pipeline(
    "sentiment-analysis",
    model=pretrained_name,
    tokenizer=pretrained_name
)

def labell(text):
    label = nlp(text)
    return label

def extract_label_score(row):
    df = pd.json_normalize(row)
    return pd.Series([df['label'][0], df['score'][0]])

df['label_score'] = df['Clean_Text'].apply(lambda x:labell(x))

df[['indonlp_sentiment', 'score']] =
df['label_score'].apply(extract_label_score)

df.to_csv("path.csv",)

```

f. Normalisasi Lokasi Tweet

```

import pandas as pd
import numpy as np
import ast
import re
import json

df = pd.read_csv('Clean_Dataset.csv', encoding='UTF-8')

my_file = open('cleaning_source/kota_baru.txt', 'r')
content = my_file.read()
kota = content.split('\n')

file_2 = open('cleaning_source/provinsi_kota.txt', 'r')
content2 = file_2.read()
provinsi = ast.literal_eval(content2)

def city(location):
    location = str(location).lower()
    for daerah in kota:
        if daerah.lower() in location:
            return daerah

df['City'] = df['Location'].apply(lambda x: city(x))

def province(location):
    location = str(location).lower()
    for key,value in daerah.items():
        if key.lower() == location:
            return key.title()
    else:
        for x in value:
            if x.lower() == location:
                return key.title()
    for key,value in provinsi.items():
        if key.lower() == location:
            return key.title()
        for x in value:
            if x.lower() == location:

```

```
        return key.title()

df['Province'] = df['City'].apply(lambda x:province(x))

df.to_csv('Clean_Dataset.csv', index=False)
```

Lampiran 7 Source Code Lexicon-Based

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import pytz
import re
import nltk
import ast
import string
import itertools
import seaborn as sns

from datetime import datetime, timedelta
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from wordcloud import WordCloud
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
import random

from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report

df = pd.read_csv('path.csv', encoding='utf-8')

# Import Lexicon-Data dan buat list kata negasi
negasi = ['bukan', 'tidak', 'ga', 'gk', 'g', 'ngga', 'nggak', 'no']
lexicon = pd.read_csv('path.csv')
lexicon = lexicon.drop(lexicon[(lexicon['word'] == 'bukan') |
                               (lexicon['word'] == 'tidak') |
                               (lexicon['word'] == 'ga') |
                               (lexicon['word'] == 'gk') |
                               (lexicon['word'] == 'ngga') |
                               (lexicon['word'] == 'nggak') |
                               (lexicon['word'] == 'no') |
                               (lexicon['word'] ==
                               'gk')].index, axis=0)
lexicon = lexicon.reset_index(drop=True)

```

```

#Hitung nilai sentimen tiap kalimat

sencol =[]
senrow =np.array([])
nsen = 0
factory = StemmerFactory()
stemmer = factory.create_stemmer()
sentiment_list = []
# fungsi untuk menulis nilai kata jika ditemukan
def found_word(ind,words,word,sen,sencol,sentiment,add):
    # jika terdapat pada matrix Bag of Words, tingkatan
    nilainya
    if word in sencol:
        sen[sencol.index(word)] += 1
    else:
        # jika tidak, tambahkan kata baru
        sencol.append(word)
        sen.append(1)
        add += 1
        # jika terdapat kata negasi sebelumnya, nilai sentimen akan
        menjadi negatif
        if (words[ind-1] in negasi):
            sentiment += -
lexicon['weight'][lexicon_word.index(word)]
    else:
        sentiment += lexicon['weight'][lexicon_word.index(word)]

    return sen,sencol,sentiment,add

# memeriksa setiap kata, jika terdapat pada kamus lexicon, maka
hitung nilai sentimennya
for i in range(len(df)):
    nsen = senrow.shape[0]
    words = word_tokenize(df['Clean_Text'][i])
    sentiment = 0
    add = 0
    prev = [0 for ii in range(len(words))]
    n_words = len(words)
    if len(sencol)>0:

```

```

        sen =[0 for j in range(len(sencol))]
    else:
        sen =[]

    for word in words:
        ind = words.index(word)
        # Memeriksa apakah terdapat pada kamus lexicon
        if word in lexicon_word :
            sen,sencol,sentiment,add=
found_word(ind,words,word,sen,sencol,sentiment,add)
        else:
            # jika tidak, periksa kata dasarnya
            kata_dasar = stemmer.stem(word)
            if kata_dasar in lexicon_word:
                sen,sencol,sentiment,add=
found_word(ind,words,kata_dasar,sen,sencol,sentiment,add)
            # jika masih tidak ditemukan, coba gabungkan dengan kata
            sebelumnya
            elif(n_words>1):
                if ind-1>-1:
                    back_1 = words[ind-1]+' '+word
                    if (back_1 in lexicon_word):
                        sen,sencol,sentiment,add=
found_word(ind,words,back_1,sen,sencol,sentiment,add)
                    elif(ind-2>-1):
                        back_2 = words[ind-2]+' '+back_1
                        if back_2 in lexicon_word:
                            sen,sencol,sentiment,add=
found_word(ind,words,back_2,sen,sencol,sentiment,add)
            # jika ditemukan kata baru, maka perluas matrix
            if add>0:
                if i>0:
                    if (nsen==0):
                        senrow = np.zeros([i,add],dtype=int)
                    elif(i!=nsen):
                        padding_h = np.zeros([nsen,add],dtype=int)
                        senrow = np.hstack((senrow,padding_h))
                        padding_v = np.zeros([(i-
nsen),senrow.shape[1]],dtype=int)
                        senrow = np.vstack((senrow,padding_v))

```

```

        else:
            padding = np.zeros([nsen, add], dtype=int)
            senrow = np.hstack((senrow, padding))
            senrow = np.vstack((senrow, sen))
        if i==0:
            senrow = np.array(sen).reshape(1, len(sen))
        # jika tidak, perbarui matrix lama
        elif(nsen>0):
            senrow = np.vstack((senrow, sen))

        sentiment_list.append(sentiment)

sencol.append('lexicon_sentiment')
sentiment_array =
np.array(sentiment_list).reshape(senrow.shape[0], 1)
sentiment_data = np.hstack((senrow, sentiment_array))
df_sen = pd.DataFrame(sentiment_data, columns = sencol)

scaler = MinMaxScaler()
sentiment_scores = df['Lexicon_Score'].values

scale_score = scaler.fit_transform(sentiment_scores.reshape(-
1, 1))

df['Score_Scaled'] = scale_score

df.to_csv('Clean_Dataset.csv', index=False)

```

Lampiran 8 Source Code LSTM

```

import numpy as np
import pandas as pd
import re
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score

from keras.models import Sequential
from keras import layers
from keras.layers import Embedding

```

```

import seaborn as sns
sns.set(style = 'whitegrid')

from mlxtend.plotting import plot_confusion_matrix
from sklearn.metrics import confusion_matrix
from sklearn.preprocessing import MinMaxScaler

from keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import
pad_sequences
from keras import regularizers
from sklearn.metrics import classification_report,
confusion_matrix

df = pd.read_csv('../Cleaning/Clean_Dataset.csv', encoding='UTF-
8')

train = df[['Clean_Text', 'Stop_Words_Text', 'Stemmed_Text',
'indonlp_sentiment', 'Lexicon_Score']]

#preprocessing
data = df['Clean_Text'].values.tolist()

#label encoding kelas sentimen
import tensorflow as tf
labels = np.array(train['indonlp_sentiment'])
y = []
for i in range(len(labels)):
    if labels[i] == 'neutral':
        y.append(0)
    if labels[i] == 'negative':
        y.append(1)
    if labels[i] == 'positive':
        y.append(2)
y = np.array(y)
labels = tf.keras.utils.to_categorical(y, 3, dtype="float32")
del y

#tokenizing dan padding

```

```

max_words = 50000
max_len = 100

tokenizer = Tokenizer(num_words=max_words)
tokenizer.fit_on_texts(data)
sequences = tokenizer.texts_to_sequences(data)
tweets = pad_sequences(sequences, maxlen=max_len)
print(tweets)

# split dataset
X_train, X_test, y_train, y_test = train_test_split(tweets,
labels, test_size = 0.2, random_state = 42)
a, X_val, b, y_val = train_test_split(X_train, y_train,
test_size=0.25, random_state=42)
print(X_train.shape, y_train.shape)
print(X_test.shape, y_test.shape)

# mencari parameter terbaik dengan fungsi GridSearchCV
def create_model(learning_rate=0.001, units=20, dropout=0.3):
    model = Sequential()
    model.add(Embedding(max_words, 40, input_length=max_len))
    model.add(BatchNormalization())
    model.add(Bidirectional(LSTM(units, dropout=dropout)))
    model.add(Dense(3, activation='softmax'))
    optimizer = Adam(learning_rate=learning_rate)
    model.compile(optimizer=optimizer,
loss='categorical_crossentropy', metrics=['accuracy'])
    return model
model_wrapper = KerasClassifier(build_fn=create_model,
epochs=15, batch_size=32, verbose=0)

# definisikan parameter untuk Grid Search
param_grid = {
    'learning_rate': [0.001, 0.01],
    'units': [20,32,40, 50],
    'dropout': [0.2, 0.25,0.3,0.4,]
}

# membuat objek Grid Search

```

```

grid = GridSearchCV(estimator=model_wrapper,
param_grid=param_grid, cv=3)

# Latih Model dengan Grid Search
grid_result = grid.fit(X_train, y_train, validation_data=(X_val,
y_val))

# mendapatkan parameter dan hasil terbaik
best_params = grid_result.best_params_
best_score = grid_result.best_score_

print("Best parameters:", best_params)
print("Best accuracy:", best_score)

#bangun model dengan beberapa layers
model2 = Sequential()
model2.add(layers.Embedding(max_words, 40,
input_length=max_len))
model2.add(layers.BatchNormalization())
model2.add(layers.Bidirectional(layers.LSTM(20,dropout=0.25)))
model2.add(layers.Dense(3,activation='softmax'))
model2.compile(optimizer=Adam(learning_rate=0.001),loss='categor
ical_crossentropy', metrics=['accuracy'])

#latih model
history = model2.fit(X_train,y_train,
epochs=20,validation_data=(X_val, y_val))

#visualisasikan grafik untuk melihat performa model
def plot_training_hist(history):

    fig, ax = plt.subplots(1,2,figsize=(10,4))
    #first plot
    ax[0].plot(history.history['accuracy'])
    ax[0].plot(history.history['val_accuracy'])
    ax[0].set_title('Model Accuracy')
    ax[0].legend(['Train', 'Validation'], loc='best')
    #second plot
    ax[1].plot(history.history['loss'])
    ax[1].plot(history.history['val_loss'])

```

```

ax[1].set_title('Model Loss')
ax[1].legend(['Train', 'Validation'], loc='best')

plot_training_hist(history)

# Prediksi sentimen pada data test menggunakan model yang telah
dibangun, lalu visualisasikan confusion matrix
y_pred = np.argmax(model2.predict(tweets), axis=1)
y_true = np.argmax(labels, axis=1)
accuracy = accuracy_score(y_true, y_pred)
print('Model Accuracy on Test Data:', accuracy)
confusion_matrix(y_true=y_true, y_pred=y_pred)
fig, ax = plt.subplots(figsize=(8,6))
sns.heatmap(confusion_matrix(y_true=y_true, y_pred=y_pred),
fmt='g', annot=True)
ax.xaxis.set_label_position('top')
ax.xaxis.set_ticks_position('top')
ax.set_xlabel('Prediksi', fontsize=14)
ax.set_xticklabels(['Negatif', 'Netral', 'Positif'])
ax.set_ylabel('Aktual', fontsize=14)
ax.set_yticklabels(['Negative', 'Netral', 'Positif'])
plt.show()

# prediksi kelas pada data test
y_pred = np.argmax(model2.predict(X_test), axis=-1)

# ubah one-hot encoded true label menjadi label kelas integer
y_test_int = np.argmax(y_test, axis=1)

# hitung metrics
print(classification_report(y_test_int, y_pred))

# hitung confusion matrix
conf_mat = confusion_matrix(y_test_int, y_pred)
print(conf_mat)

# Hitung akurasi, presisi, recall, dan F1-score
accuracy = accuracy_score(y_test_int, y_pred)
precision = precision_score(y_test_int, y_pred, average='macro')
recall = recall_score(y_test_int, y_pred, average='macro')

```

```
f1 = f1_score(y_test_int, y_pred, average='macro')

# Print hasil
print('Accuracy: {:.2f}%'.format(accuracy * 100))
print('Precision: {:.2f}%'.format(precision * 100))
print('Recall: {:.2f}%'.format(recall * 100))
print('F1-score: {:.2f}%'.format(f1 * 100))

#prediksi dan hasil
sentiment = ['Neutral', 'Negative', 'Positive']

text_input = input('Masukkan Teks untuk diprediksi:')
sequence = tokenizer.texts_to_sequences([text_input])
test = pad_sequences(sequence, maxlen=max_len)
sentiment[np.argmax(model2.predict(test),
decimals=0).argmax(axis=1) [0]]
```

Lampiran 9 Source Code Pendekatan Gabungan LSTM dan Lexicon-Based

```

import numpy as np
import pandas as pd
import re

from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score

from keras.models import Sequential
from keras import layers
from keras.layers import Embedding

import seaborn as sns
sns.set(style = 'whitegrid')

from mlxtend.plotting import plot_confusion_matrix
from sklearn.metrics import confusion_matrix
from sklearn.preprocessing import MinMaxScaler

from keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import
pad_sequences
from keras import regularizers
from sklearn.metrics import classification_report,
confusion_matrix

df = pd.read_csv('../Cleaning/Clean_Dataset.csv', encoding='UTF-
8')

train = df[['Clean_Text', 'Stop_Words_Text', 'Stemmed_Text',
'indonlp_sentiment', 'Lexicon_Score']]

#preprocessing
data = df['Clean_Text'].values.tolist()

#label encoding kelas sentimen
import tensorflow as tf
labels = np.array(train['indonlp_sentiment'])
y = []

```

```

for i in range(len(labels)):
    if labels[i] == 'neutral':
        y.append(0)
    if labels[i] == 'negative':
        y.append(1)
    if labels[i] == 'positive':
        y.append(2)
y = np.array(y)
labels = tf.keras.utils.to_categorical(y, 3, dtype="float32")
del y

#tokenizing dan padding
max_words = 50000
max_len = 100

tokenizer = Tokenizer(num_words=max_words)
tokenizer.fit_on_texts(data)
sequences = tokenizer.texts_to_sequences(data)
X = pad_sequences(sequences, maxlen=max_len)
print(X)

scaler = MinMaxScaler()

# extract sentiment scores
sentiment_scores = df['Lexicon_Score'].values

# perform min-max scaling on sentiment scores
scaler = MinMaxScaler()
sentiment_scaled =
scaler.fit_transform(sentiment_scores.reshape(-1, 1))

df['Sentiment_Scaled'] = sentiment_scaled

#Combine sentiment scores with text data
X = X[:, :-1]
X = np.hstack((sentiment_scaled, X))
y = labels

# split dataset
X_train, X_test, y_train, y_test = train_test_split(tweets,
labels, test_size = 0.2, random_state = 42)

```

```

a, X_val, b, y_val = train_test_split(X_train, y_train,
test_size=0.25, random_state=42)
print(X_train.shape, y_train.shape)
print(X_test.shape, y_test.shape)

# mencari parameter terbaik dengan fungsi GridSearchCV
def create_model(learning_rate=0.001, units=20, dropout=0.3):
    model = Sequential()
    model.add(Embedding(max_words, 40, input_length=max_len))
    model.add(BatchNormalization())
    model.add(Bidirectional(LSTM(units, dropout=dropout)))
    model.add(Dense(3, activation='softmax'))
    optimizer = Adam(learning_rate=learning_rate)
    model.compile(optimizer=optimizer,
loss='categorical_crossentropy', metrics=['accuracy'])
    return model
model_wrapper = KerasClassifier(build_fn=create_model,
epochs=15, batch_size=32, verbose=0)

# mendefinisikan parameter untuk Grid Search
param_grid = {
    'learning_rate': [0.001, 0.01],
    'units': [20,32,40, 50],
    'dropout': [0.2, 0.25,0.3,0.4,]
}

# membuat objek Grid Search
grid = GridSearchCV(estimator=model_wrapper,
param_grid=param_grid, cv=3)

# Latih Model dengan Grid Search
grid_result = grid.fit(X_train, y_train, validation_data=(X_val,
y_val))

# mendapatkan parameter dan hasil terbaik
best_params = grid_result.best_params_
best_score = grid_result.best_score_

print("Best parameters:", best_params)
print("Best accuracy:", best_score)

#bangun model dengan beberapa layers

```

```

model2 = Sequential()
model2.add(layers.Embedding(max_words, 40,
input_length=max_len))
model2.add(layers.BatchNormalization())
model2.add(layers.Bidirectional(layers.LSTM(20,dropout=0.25)))
model2.add(layers.Dense(3,activation='softmax'))
model2.compile(optimizer=Adam(learning_rate=0.001),loss='categor
ical_crossentropy', metrics=['accuracy'])

#latih model
history = model2.fit(X_train,y_train,
epochs=20,validation_data=(X_val, y_val))

#visualisasikan grafik untuk melihat performa model
def plot_training_hist(history):

    fig, ax = plt.subplots(1,2,figsize=(10,4))
    #first plot
    ax[0].plot(history.history['accuracy'])
    ax[0].plot(history.history['val_accuracy'])
    ax[0].set_title('Model Accuracy')
    ax[0].legend(['Train', 'Validation'], loc='best')
    #second plot
    ax[1].plot(history.history['loss'])
    ax[1].plot(history.history['val_loss'])
    ax[1].set_title('Model Loss')
    ax[1].legend(['Train', 'Validation'], loc='best')

plot_training_hist(history)

# Prediksi sentimen pada data test menggunakan model yang telah
dibangun, lalu visualisasikan confusion matrix
y_pred = np.argmax(model2.predict(tweets), axis=1)
y_true = np.argmax(labels, axis=1)
accuracy = accuracy_score(y_true, y_pred)
print('Model Accuracy on Test Data:', accuracy)
confusion_matrix(y_true=y_true, y_pred=y_pred)
fig, ax = plt.subplots(figsize=(8,6))
sns.heatmap(confusion_matrix(y_true=y_true, y_pred=y_pred),
fmt='g', annot=True)
ax.xaxis.set_label_position('top')
ax.xaxis.set_ticks_position('top')

```

```

ax.set_xlabel('Prediksi', fontsize=14)
ax.set_xticklabels(['Negatif', 'Netral', 'Positif'])
ax.set_ylabel('Aktual', fontsize=14)
ax.set_yticklabels(['Negative', 'Netral', 'Positif'])
plt.show()

# prediksi kelas pada data test
y_pred = np.argmax(model2.predict(X_test), axis=-1)

# ubah one-hot encoded true label menjadi label kelas integer
y_test_int = np.argmax(y_test, axis=1)

# hitung metrics
print(classification_report(y_test_int, y_pred))

# hitung confusion matrix
conf_mat = confusion_matrix(y_test_int, y_pred)
print(conf_mat)

# Hitung akurasi, presisi, recall, dan F1-score
accuracy = accuracy_score(y_test_int, y_pred)
precision = precision_score(y_test_int, y_pred, average='macro')
recall = recall_score(y_test_int, y_pred, average='macro')
f1 = f1_score(y_test_int, y_pred, average='macro')

# Print hasil
print('Accuracy: {:.2f}%'.format(accuracy * 100))
print('Precision: {:.2f}%'.format(precision * 100))
print('Recall: {:.2f}%'.format(recall * 100))
print('F1-score: {:.2f}%'.format(f1 * 100))

#prediksi dan hasil
sentiment = ['Neutral', 'Negative', 'Positive']

text_input = input('Masukkan Teks untuk diprediksi:')
sequence = tokenizer.texts_to_sequences([text_input])
test = pad_sequences(sequence, maxlen=max_len)
sentiment[np.around(model2.predict(test),
decimals=0).argmax(axis=1)[0]]

```