

Penerapan *Modified K-Nearest Neighbor* dengan Algoritma Genetika Pada Prediksi PM10 di Pekanbaru

Fitri Insani^{1*}, Syarifatun Nissa²

^{1,2}Program Studi Teknik Informatika,, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sultan Syarif Kasim Riau

Jalan HR. Soebrantas Panam Km. 15 No. 155, Tuah Madani, Kec. Tampan, Kabupaten Kampar, Riau 28293

*email: fitri.insani@uin-suska.ac.id

(Naskah masuk: 14 Januari 2021; diterima untuk diterbitkan: 14 April 2021)

ABSTRAK – Pencemaran udara merupakan suatu kondisi udara yang tercemar oleh bahan-bahan, zat-zat, atau partikel kimia dan juga bahan biologi lainnya yang membahayakan kesehatan makhluk hidup. Salah satu zat yang menyebabkan pencemaran udara adalah PM10. PM10 (particulate matter) atau lebih dikenal dengan partikel debu adalah partikel udara dalam wujud padat yang berdiameter kurang dari 10 μm . Dampak PM10 dalam udara dapat dirasakan langsung oleh masyarakat, seperti gangguan pernafasan. Penelitian prediksi PM10 sebelumnya sudah banyak dilakukan sebelumnya. Salah satu metode yang dapat memprediksi PM10 yaitu Modified K-Nearest Neighbor (MKNN) . Namun, MKNN mempunyai kekurangan yaitu k bias dan komputasi yang kompleks. Berdasarkan kekurangan metode MKNN, maka dilakukan perbaikan dengan mengoptimasi nilai k menggunakan Algoritma Genetika. Data PM10 yang digunakan merupakan data PM10 per 30 menit pada bulan Juli sampai bulan Desember tahun 2015 yang diambil dari laboratorium udara kota Pekanbaru. Data ini kemudian diubah menjadi deret waktu dengan 48 variabel masukan dan 1 variabel keluaran. Hasil dari penelitian ini menunjukkan bahwa metode MKNN dapat memprediksi PM10 dengan error terendah yaitu 8,957 dan metode Algoritma Genetika dapat mencari nilai k optimal pada MKNN dengan k optimal yaitu 3.

Kata Kunci – Algoritma genetika; deret waktu; modified k-nearest neighbor; PM10; prediksi.

The Implementation of Modified k-Nearest Neighbor with Genetic Algorithm on PM10 Prediction at Pekanbaru

ABSTRACT – Air pollution is an air condition that is polluted by materials, substances, or chemical particles and also other biological materials that harm the health of living things. One of the substances that causes air pollution is PM₁₀. PM₁₀ (particulate matter) or better known as dust particles are air particles in solid form with diameter less than 10 μm . The impact of PM₁₀ in the air can be directly felt by people, such as respiratory disorders. There were many researches about PM₁₀ prediction had been done. One method that can predict PM10 is Modified K-Nearest Neighbor (MKNN). However, MKNN method has disadvantages, namely k-bias and complex computing. Based on the disadvantages of MKNN method, improvements were made by optimizing the value of k using Genetic Algorithm. The PM10 data used are PM10 data each 30 minutes from July to December 2015 which were taken from the Pekanbaru city air laboratory. The data was converted into a time series with 48 input variables and 1 output variable. The result of this study shows that MKNN can predict PM10 with the lowest error that is 8.957 and the genetic algorithm can find the optimal-k value in MKNN with optimal-k that is 3.

Keywords - Modified K-Nearest Neighbor; Genetic Algorithm; PM10, Prediction; Time Series.

1. PENDAHULUAN

Particulate matter (PM10) atau lebih dikenal dengan partikel debu adalah partikel udara dalam wujud padat yang berdiameter kurang dari 10 μm .

Kadar PM10 diudara sering berubah-ubah. Berdasarkan data yang didapat dari laboratorium udara di Pekanbaru PM10 diukur setiap 30 menit, sedangkan menurut data dari Badan Meteorologi,

Klimatologi, dan Geofisika (BMKG) PM10 diukur setiap 1 jam. Dampak PM10 dalam udara dapat dirasakan langsung oleh masyarakat, seperti gangguan pernafasan pada manusia dan terhambatnya proses fotosintesis pada tumbuhan [1].

Penelitian prediksi udara sudah banyak dilakukan oleh peneliti sebelumnya [1]-[10]. Selain menggunakan beberapa metode yang sudah digunakan peneliti sebelumnya, prediksi udara juga dapat diukur dengan metode *modified k-nearest neighbor* (MKNN). MKNN merupakan metode yang dikembangkan dari metode *k-nearest neighbor* (KNN), namun dalam algoritma MKNN menambahkan proses baru untuk melakukan prediksi yaitu, perhitungan nilai validitas untuk mempertimbangkan validitas antar data latih dan pehitungan nilai *weighted voting* untuk menghitung bobot [11]. Penelitian dengan menggunakan metode MKNN sudah banyak dilakukan sebelumnya [11]-[15]. Namun ternyata MKNN juga memiliki kekurangan yaitu k bias dan komputasi yang kompleks [16]. Pada penelitian selanjutnya bisa dilakukan dengan hibridisasi MKNN dan algoritma heuristik seperti Algoritma Genetika yang terbukti efektif untuk permasalahan yang kompleks [17].

Berdasarkan beberapa kekurangan metode MKNN, maka dilakukan perbaikan dengan mengoptimasi nilai k menggunakan Algoritma Genetika. Algoritma Genetika dipilih karena sudah terbukti dapat digunakan untuk melakukan optimasi pada nilai k untuk MKNN [18]. Selain itu Algoritma Genetika dipilih karena Algoritma Genetika yang terbukti efektif untuk permasalahan yang kompleks dalam waktu yang relatif cepat [19].

Penelitian menggabungkan metode MKNN dan Algoritma Genetika sudah dilakukan oleh beberapa peneliti sebelumnya [16], [18], [20]. Berdasarkan hasil beberapa penelitian diatas Algoritma Genetika terbukti dapat mengoptimasi MKNN sehingga didapatkan nilai K optimal.

Pada penelitian ini akan dibangun suatu aplikasi matlab yang dapat memprediksi PM10 dengan kombinasi MKNN dan Algoritma Genetika. Metode Algoritma Genetika dalam penelitian ini digunakan untuk mendapatkan nilai K optimal dari metode MKNN, sedangkan metode MKNN digunakan untuk mencari hasil prediksi PM10. Data PM10 yang digunakan dalam penelitian ini merupakan data deret waktu PM10 per 30 menit yang didapat dari laboratorium udara kota Pekanbaru.

Lebih lanjut hasil penelitian dapat membantu memberikan informasi kadar PM10 30 menit kemudian dan penelitian ini membuktikan bahwa Algoritma Genetika mampu menghasilkan nilai k optimal untuk metode MKNN.

2. METODE DAN BAHAN

Tahapan proses prediksi menggunakan metode optimasi MKNN menggunakan Algoritma Genetika ini dibagi menjadi tiga proses yaitu analisa kebutuhan data, proses pelatihan dan proses pengujian.

1. Kebutuhan Data

Data yang dibutuhkan dalam penelitian ini adalah data deret waktu. Data yang digunakan merupakan data kadar PM10 dari tanggal 1 Juli 2015 pukul 00.30 hingga data kadar PM10 dari tanggal 31 Desember 2015 pukul 24.00. Data kadar PM10 didapat dari laboratorium udara kota Pekanbaru dapat dilihat pada Tabel 1.

Tabel 1. Data kadar PM10

No	Tanggal	Jam	Kadar PM10
1	1 Juli 2015	00:30	31
2	1 Juli 2015	01:00	34
3	1 Juli 2015	01:30	38
4	1 Juli 2015	02:00	40
5	1 Juli 2015	02:30	43
...
8640	31 Desember 2015	24:00	23

Selanjutnya data tersebut diubah menjadi data deret waktu dengan 48 variabel masukan ($t-48, t-47, t-46, t-45, \dots, t-1$) dan 1 variabel keluaran (t).

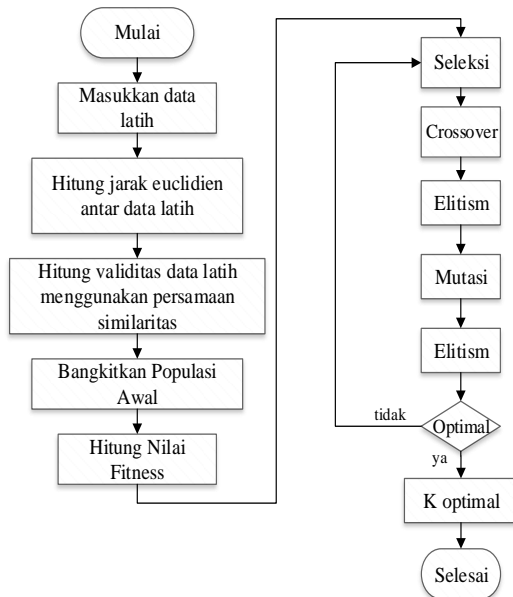
Tabel 2. Data deret waktu kadar PM10 di kota Pekanbaru

No	$t-48$	$t-47$	$t-46$	$t-45$...	$t-1$	T
1	31	34	38	40	...	23	24
2	34	38	40	43	...	24	35
3	38	40	43	52	...	35	46
4	40	43	52	61	...	46	39
5	43	52	61	60	...	39	31
...
2812	9	8	9	12	...	24	24

Data pada Tabel 2 selanjutnya akan dibagi menjadi 2 bagian yaitu data latih dan data uji. Persentase data latih dan data uji yang digunakan dalam penelitian ini yaitu 70%:30%, 80%:20%, dan 90%:10%.

2. Proses Pelatihan

Ada beberapa tahapan dalam proses pelatihan. Gambar proses pelatihan dapat dilihat pada Gambar 1.



Gambar 1. Proses pelatihan

Tahap 1: Masukkan data latih

Tahap 2: Hitung jarak *eucliden* antar data latih dengan menggunakan rumus persamaan 1.

$$D = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \quad (1)$$

dimana x adalah sampel data, y adalah data uji dan D adalah jarak.

Tahap 3: Hitung nilai validitas antar data latih menggunakan persamaan similaritas. Rumus validitas data latih dan similaritas dapat dilihat pada persamaan 2 dan 3.

$$S(a, b) = \begin{cases} 1 & a = b \\ 0 & a \neq b \end{cases} \quad (2)$$

dimana S adalah nilai similaritas antar data latih, a dan b adalah kelas antar data latih.

$$\text{Validitas}(x) = \frac{1}{H} \sum_{i=1}^H S(\text{lbl}(x), \text{lbl}(Ni(x))) \quad (3)$$

dimana H adalah jumlah titik terdekat dan (x) adalah label kelas titik terdekat.

Tahap 4: Representasi kromosom. Untuk menentukan kromosom maka harus ditentukan populasi dari kromosom. Misal populasi yang diinginkan adalah 3. Maka secara *random* akan dibangkitkan kromosom (kemungkinan solusi) sebanyak 3 buah, dengan ketentuan, nilai $k < \text{data latih}$. Misal hasil kromosom didapatkan 3, 9, 4. Selanjutnya nilai kromosom tersebut diubah menjadi

biner, misal menjadi 0011 untuk kromosom 3, 0100 untuk kromosom 4 dan 1001 untuk kromosom 9.

Tahap 5: Menghitung nilai fitness. Rumus nilai fitness pada penelitian ini dapat dilihat pada persamaan 4.

$$\text{Nilai fitness} = \frac{\text{jumlah validitas data latih}(k)}{\text{jumlah data latih}} \quad (4)$$

Tahap 6: Seleksi dengan *roulette wheel*. Tahapan dalam seleksi *roulette wheel* yaitu:

Menghitung jumlah nilai *fitness* dari semua individu dalam suatu populasi dengan rumus pada persamaan 5.

$$\text{total fitness} = \sum_{i=1}^n \text{fitness} \quad (5)$$

Setelah itu hitung nilai probabilitas tiap individu dengan rumus pada persamaan 6.

$$\text{Probabilitas}(n) = \frac{\text{fitness}}{\text{total fitness}} \quad (6)$$

Hitung probabilitas kumulatif tiap individu dengan rumus pada persamaan 7.

$$\text{Probabilitas kumulatif}(n) = \sum_{i=1}^n \text{probabilitas}(n) \quad (7)$$

Tahap 7: *Crossover one point*. Pada proses ini dilakukan penukaran nilai gen induk pertama dan induk kedua dari posisi awal sampai dengan posisi akhir untuk diperoleh anak pertama dan anak kedua.

Tahap 8: *Elitism*. Proses *elitism* tahap ini dilakukan dengan cara menggabungkan individu hasil *crossover* dengan individu pada populasi awal. Dari hasil penggabungan ini kemudian akan dibagi dua kelas yaitu individu terbaik dan individu terburuk. Kelas individu terbaik akan disimpan sedangkan kelas individu terburuk akan dimutasi.

Tahap 9: Mutasi. Pada kromosom biner, mutasi dilakukan dengan mengubah gen biner 0 menjadi 1 dan 1 menjadi 0. Untuk mendapatkan gen yang akan dimutasi haruslah ada probabilitas mutasi. Probabilitas mutasi yang digunakan pada penelitian ini adalah 0,1.

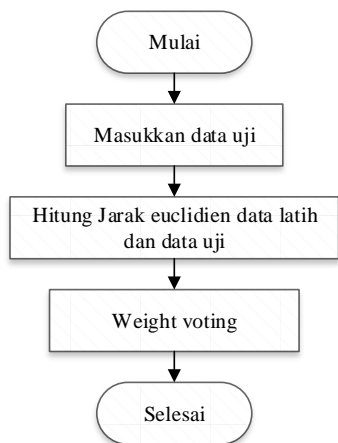
Tahap 10: *Elitism*. Proses *elitism* pada tahap ini yaitu populasi hasil mutasi akan digabungkan dengan kelas individu terbaik, kemudian dari penggabungan tersebut akan diambil individu-individu dengan nilai tertinggi sejumlah populasi.

Tahap 11: Didapat individu terbaru dari nilai fitness terbaik. Selanjutnya proses Algoritma Genetika akan diulang mulai dari seleksi hingga *elitism* untuk

mendapatkan nilai optimal sampai generasi yang sudah ditentukan.

3. Proses Pengujian

Langkah langkah-langkah proses pengujian menggunakan metode optimasi MKNN dengan Algoritma Genetika dapat lihat pada Gambar 2.



Gambar 2. Proses pengujian

Tahap 1. Masukkan data uji.

Tahap 2. Hitung jarak *eucliden* dari data latih ke data uji dengan rumus seperti persamaan 1.

Tahap 3. Hitung bobot (*weight voting*) menggunakan rumus seperti persamaan 8.

$$Weight\ voting\ (i) = Validitas\ (x) \times \frac{1}{de + \alpha} \quad (8)$$

dimana Validitas (x) adalah nilai validasi data, de adalah jarak *eucliden*, dan α adalah nilai regulator *smoothing* (pemulusan).

Setelah didapat nilai *weight voting*, urutkan nilai *weight voting* dari yang terbesar hingga terkecil.

Ambil urutan nilai *weight voting* sebanyak nilai k. Bila sebanyak nilai k terdapat lebih 1 nilai *weight voting* yang memiliki kesamaan kelas maka nilai *weight voting* dijumlah terlebih dahulu, setelah itu tiap nilai *weight voting* yang kelasnya berbeda dibandingkan. Dan ambil nilai terbesar sebagai kelas hasil prediksi data uji.

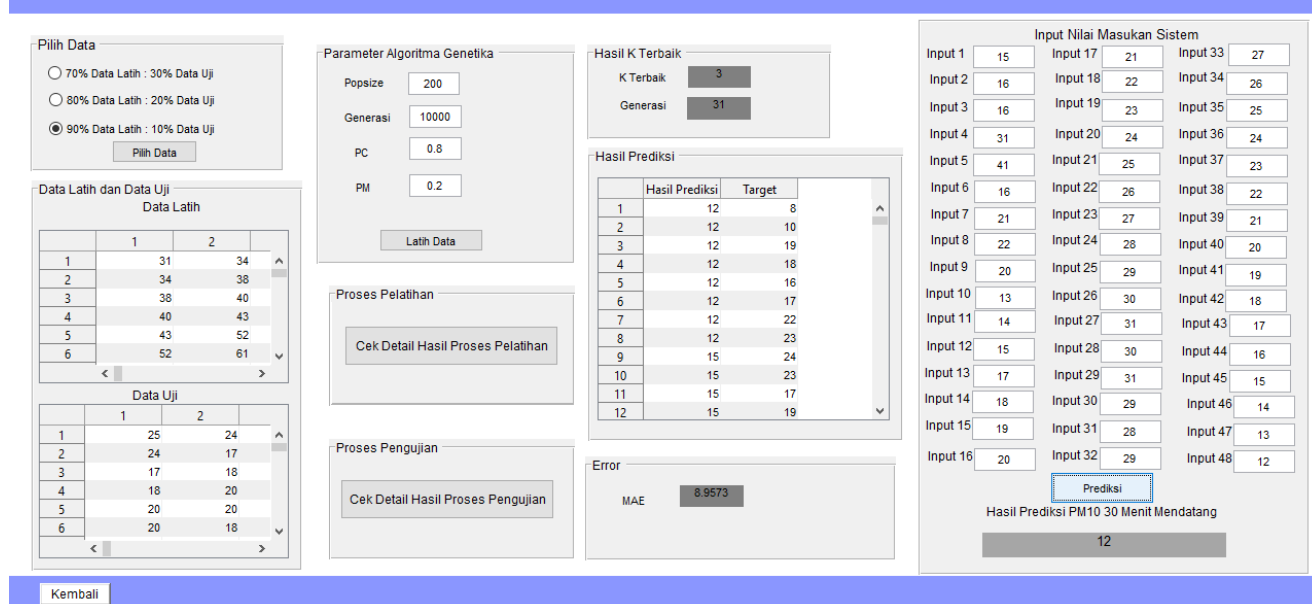
3. HASIL DAN PEMBAHASAN

3.1 Penerapan Menggunakan Aplikasi Matlab

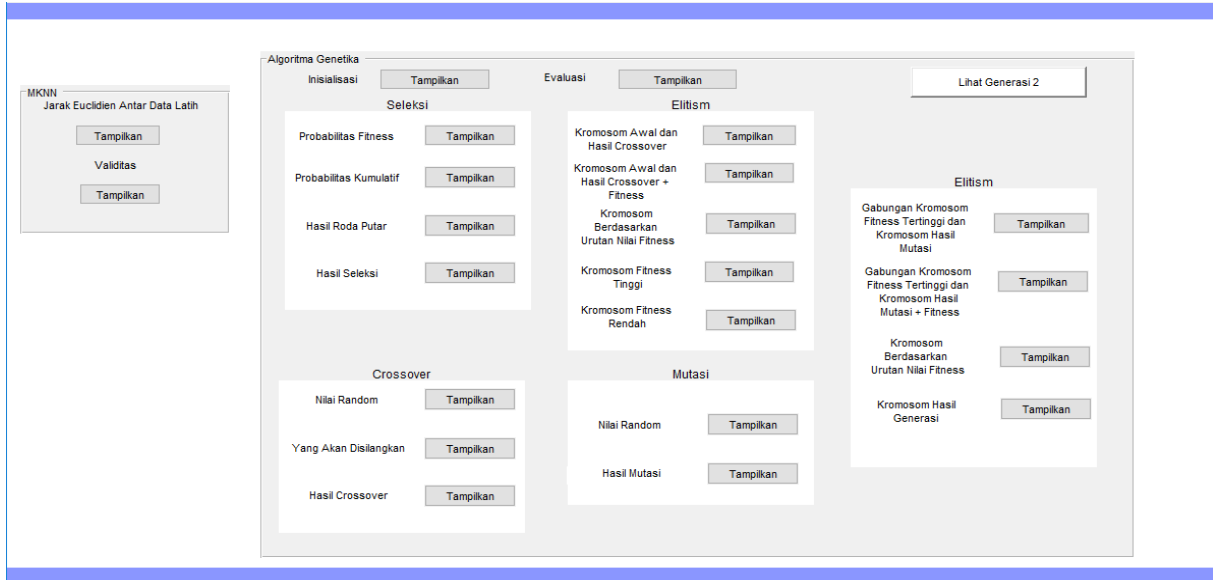
Implementasi penelitian ini yaitu dengan menggunakan aplikasi matlab. Penerapan menggunakan aplikasi matlab dimulai dengan menampilkan halaman utama seperti yang ditunjukkan pada Gambar 3.

Pada halaman menu utama ini berisi inti dari proses prediksi PM10 dengan menggunakan metode MKNN dan Algoritma Genetika. Pada menu ini user akan memilih perbandingan data latih dan data uji, lalu data latih dan data uji muncul. Setelah data latih dan data uji muncul user akan memasukkan parameter yang terdiri data *popsi*, generasi, PC dan PM. Selanjutnya user dapat melihat hasil detail dari proses pelatihan dan pengujian. Halaman proses pelatihan dapat dilihat pada Gambar 4 dan proses pengujian dapat dilihat pada Gambar 5.

Pada halaman proses pelatihan akan muncul beberapa proses dan jika dipilih tampilkan maka akan muncul tabel hasil masing-masing proses. Pada kategori *crossover* ada 3 tahap yaitu nilai random, kromosom yang akan disilangkan dan hasil *crossover*. Nilai random *crossover* dapat dilihat pada Gambar 6 dan kromosom yang akan disilangkan dapat dilihat pada Gambar 7. Gambar 8 merupakan salah satu hasil dari proses pengujian yaitu jarak eucliden antar data latih.



Gambar 3. Halaman utama



Gambar 4. Halaman proses pelatihan



Gambar 5. Halaman Proses Pengujian

	1
1	0.4229
2	0.0942
3	0.5985
4	0.4709
5	0.6959
6	0.6999
7	0.6385
8	0.0336
9	0.0688
10	0.3196
11	0.5309
12	0.6544
13	0.4076
14	0.8200
15	0.7184
16	0.9686
17	0.5313
18	0.3251
19	0.1056
20	0.6110
21	0.7788
22	0.4235
23	0.0908
24	0.2665
25	0.1537
26	0.2810
27	0.4401
28	0.5271
29	0.4574
30	0.8764

Gambar 6. Nilai random crossover

	1	2	3	4	5	6	7	8	9	10	11	12
1	1	0	0	1	0	0	0	0	0	1	1	1
2	0	0	0	0	1	1	1	1	1	0	1	1
3	0	0	1	0	1	1	0	0	0	0	0	1
4	1	0	0	1	0	1	1	1	1	0	1	1
5	1	0	0	1	1	0	0	0	1	0	1	0
6	1	0	0	1	1	0	0	1	1	0	0	0
7	0	1	0	0	1	1	0	0	1	1	0	1
8	0	1	1	1	1	1	1	0	1	0	0	1
9	0	1	0	0	0	1	1	0	1	1	0	0
10	1	0	0	1	0	1	1	1	1	1	0	0
11	0	1	1	0	0	1	1	1	1	1	0	0
12	0	0	0	0	0	1	0	1	1	0	1	1
13	1	0	0	0	0	1	1	0	0	1	0	1
14	0	1	1	1	0	1	1	1	1	1	1	0
15	0	0	1	1	1	1	1	0	0	0	0	1
16	0	1	1	0	0	1	1	1	1	0	1	1
17	0	0	0	1	1	0	1	1	0	0	1	0
18	0	1	1	0	1	1	1	1	1	0	1	1
19	0	0	0	0	0	1	0	1	0	0	0	1
20	0	0	0	0	0	1	1	1	0	1	0	1
21	0	0	0	0	1	1	1	1	1	1	1	0
22	0	1	1	0	1	1	0	1	1	1	1	0
23	0	0	1	1	0	0	1	0	0	0	1	1
24	1	0	0	1	0	1	1	0	0	1	0	1
25	0	0	0	0	0	1	0	1	1	0	0	0
26	0	1	0	0	0	1	0	1	0	1	1	1
27	0	0	1	1	1	1	0	0	0	1	1	0
28	0	0	0	1	1	1	0	1	1	0	0	1
29	0	1	1	1	0	0	0	0	0	0	1	1
30	0	0	1	0	1	0	1	1	0	1	1	1

Gambar 7. Kromosom yang akan disilangkan

Gambar 8. Jarak euclidien antar data latih

3.2 Pengujian Sistem

Bagian ini menerangkan hasil pengujian GMKNN dan MKNN tanpa optimasi. Pengujian dilakukan dengan 3 pembagian data latih dan data uji yaitu 70%:30%, 80%:20% dan 90%:10%.

Pengujian MKNN dengan optimasi Algoritma Genetika menggunakan parameter perpaduan probabilitas *crossover* (PC) dan

probabilitas mutasi (PM). Pengujian ini dilakukan untuk mendapatkan hasil Mean Absolute Error (MAE), generasi Algoritma Genetika dan waktu. Waktu yang digunakan untuk menghitung dimulai dari proses pelatihan sampai didapatkan hasil prediksi. Tabel 3 sampai Tabel 5 menunjukkan hasil pengujian MKNN dan Algoritma Genetika.

Tabel 3. Hasil pengujian MKNN dengan Algoritma Genetika dengan data latih 90%, data uji 10%

No	Parameter	Pop size = 100			Pop size = 200			Pop size = 300		
		MAE	Generasi	Waktu(s)	MAE	Generasi	Waktu(s)	MAE	Generasi	Waktu(s)
1	PC = 0,9 PM = 0,1	8,957	128	200,897	8,957	167	345,654	8,957	582	876,114
2	PC = 0,8 PM = 0,2	8,957	113	181,651	8,957	38	55,873	8,957	52	79,567
3	PC = 0,7 PM = 0,3	8,957	45	63,912	8,957	58	70,873	8,957	48	64,872
4	PC = 0,6 PM = 0,4	8,957	284	497,128	8,957	45	69,760	8,957	141	187,345
5	PC = 0,5 PM = 0,5	8,957	53	86,531	8,957	59	77,898	8,957	140	176,315
6	PC = 0,4 PM = 0,6	8,957	395	653,634	8,957	90	150,986	8,957	471	998,520
7	PC = 0,3 PM = 0,7	8,957	1264	2341,653	8,957	499	778,456	8,957	167	331,829
8	PC = 0,2 PM = 0,8	8,957	147	289,771	8,957	88	120,845	8,957	141	298,920

Berdasarkan hasil dari Tabel 3 dengan data latih 90%, didapatkan hasil pengujian terbaik pada popsize=200, PC= 0,8 dan PM=0,2 dengan MAE= 8,957 yang terdapat pada generasi ke 38 dengan waktu pengujian 55,873 detik. Pengujian dengan data latih 80% pada Tabel 4, didapatkan hasil terbaik pada popsize=300, PC= 0,8 dan PM=0,2 dengan

MAE= 11,25 yang terdapat pada generasi ke 63 dengan waktu pengujian 123,102 detik. Berdasarkan pengujian pada Tabel 5 dengan data latih 70%, hasil terbaik didapatkan pada popsize=300, PC= 0,2 dan PM=0,8 dengan MAE= 9,71 yang terdapat pada generasi ke 70 dengan waktu pengujian 136,493 detik

Tabel 4. Hasil pengujian MKNN dengan Algoritma Genetika dengan data latih 80%, data uji 20%

No	Parameter	Pop size = 100			Pop size = 200			Pop size = 300		
		MAE	Generasi	Waktu(s)	MAE	Generasi	Waktu(s)	MAE	Generasi	Wakt(s)
1	PC = 0,9 PM = 0,1	11,25	293	517,145	11,25	427	905,240	11,25	389	760,106
2	PC = 0,8 PM = 0,2	11,25	337	594,805	11,25	135	286,200	11,25	63	123,102
3	PC = 0,7 PM = 0,3	11,25	475	838,375	11,25	371	786,520	11,25	364	711,256
4	PC = 0,6 PM = 0,4	11,25	380	670,700	11,25	72	152,640	11,25	466	910,564
5	PC = 0,5 PM = 0,5	11,25	250	441,250	11,25	334	708,080	11,25	384	750,336
6	PC = 0,4 PM = 0,6	11,25	134	236,510	11,25	342	725,040	11,25	466	910,564
7	PC = 0,3 PM = 0,7	11,25	460	811,900	11,25	258	546,960	11,25	205	400,570
8	PC = 0,2 PM = 0,8	11,25	479	845,435	11,25	64	125,056	11,25	165	322,410

Tabel 5. Hasil pengujian MKNN dengan Algoritma Genetika dengan data latih 70%, data uji 30%

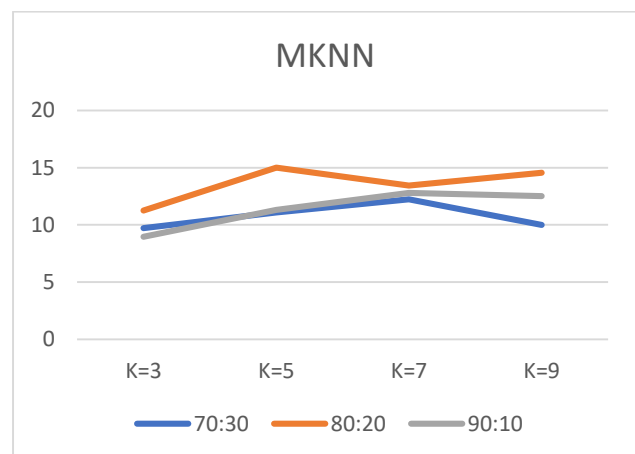
No	Parameter	Pop size = 100			Pop size = 200			Pop size = 300		
		MAE	Generasi	Waktu(s)	MAE	Generasi	Waktu(s)	MAE	Generasi	Wakt(s)
1	PC = 0,9 PM = 0,1	9,71	190	402,800	9,71	407	793,609	9,71	334	651,267
2	PC = 0,8 PM = 0,2	9,71	105	222,600	9,71	138	269,086	9,71	105	204,739
3	PC = 0,7 PM = 0,3	9,71	284	602,080	9,71	76	148,192	9,71	405	789,709
4	PC = 0,6 PM = 0,4	9,71	321	680,520	9,71	430	838,457	9,71	340	662,966
5	PC = 0,5 PM = 0,5	9,71	488	1034,560	9,71	435	848,207	9,71	485	945,706
6	PC = 0,4 PM = 0,6	9,71	236	500,320	9,71	303	590,819	9,71	471	998,520
7	PC = 0,3 PM = 0,7	9,71	433	844,307	9,71	320	623,968	9,71	167	331,829
8	PC = 0,2 PM = 0,8	9,71	451	879,405	9,71	70	136,493	9,71	141	298,920

Berdasarkan hasil pengujian GMKNN pada penelitian ini didapat K optimal adalah K=3. Hal ini dibuktikan dengan melakukan pengujian metode MKNN tanpa optimasi yang mana didapatkan hasil error terendah yaitu pada nilai K=3. Grafik untuk pengujian MKNN tanpa optimasi dapat dilihat pada Gambar 9.

Setelah dilakukan penelitian ternyata prediksi menggunakan metode MKNN dengan Algoritma Genetika menggunakan data deret waktu ini memiliki kelemahan yaitu pada tahap validitas data terdapat banyak hasil 0 karena target yang tidak sesuai sehingga berpengaruh pada proses *weight voting*.

Saran untuk penelitian selanjutnya yang ingin menggunakan metode MKNN dengan Algoritma

Genetika ini sebaiknya menggunakan target berbentuk kategori bukan numerik.



Gambar 9. Grafik error pengujian metode MKNN

4. KESIMPULAN

Kesimpulan dari pengujian penggunaan metode MKNN dengan Algoritma Genetika untuk prediksi PM10, yaitu:

1. Berdasarkan pengujian MKNN dan pengujian MKNN menggunakan Algoritma Genetika, dapat disimpulkan bahwa Algoritma Genetika berhasil mendapatkan nilai K terbaik.
2. Penerapan metode Algoritma Genetika pada MKNN memberikan hasil K optimal yaitu 3.
3. Hasil pengujian terbaik dari pengujian pembagian data = 90%:10%, popsize = 200, PC = 0,8 dan PM = 0,2 dengan MAE= 8,957 pada generasi ke 38 dengan waktu 55,873.
4. Perbandingan nilai PC dan PM terbaik berdasarkan jumlah generasi Algoritma Genetika yaitu PC=0,3 PM=0,7.
5. Penelitian ini menghasilkan hasil yang kurang baik karena metode MKNN tidak cocok menggunakan data deret waktu. Pada tahap validitas data terdapat banyak hasil 0 karena target yang tidak sesuai sehingga berpengaruh pada proses *weight voting*.

UCAPAN TERIMA KASIH

Terima kasih kami ucapkan kepada laboratorium udara kota Pekanbaru yang mendukung dan membantu dalam pelaksanaan penelitian ini.

DAFTAR PUSTAKA

- [1] C. Bernadeta and A. Suharsono, "Peramalan Kandungan Particulate Matter (PM10) dalam Udara Ambien Kota Surabaya Menggunakan Double Seasonal ARIMA (DSARIMA)," *J. Sains Dan Seni ITS*, vol. 4, no. 2, 2015.
- [2] V. Yadav and S. Nath, "Daily Prediction of PM10 using Radial Basis Function and Generalized Regression Neural Network," *IEEE Int. Conf. 2018 Recent Adv. Eng. Technol. Comput. Sci. RAETCS 2018*, pp. 1-5, 2018.
- [3] Z. Kang and Z. Qu, "Application of BP neural network optimized by genetic simulated annealing algorithm to prediction of air quality index in Lanzhou," *2017 2nd IEEE Int. Conf. Comput. Intell. Appl. ICCIA 2017*, vol. 2017-Janua, pp. 155-160, 2017.
- [4] M. M. Dedovic, S. Avdakovic, I. Turkovic, N. Dautbasic, and T. Konjic, "Forecasting PM10 concentrations using neural networks and system for improving air quality," *2016 11th Int. Symp. Telecommun. BIHTEL 2016*, 2016.
- [5] A. R. D. M. A. A. BUDI WARSITO, "Pemodelan General Regression Neural Network Untuk Prediksi Tingkat Pencemaran Udara Kota Semarang," *Med. Stat.*, vol. 1, no. 1, pp. 43-51, 2008.
- [6] N. F. Arifien, S. Arifin, B. L. Widjiantoro, and A. S. Aisjah, "Prediksi Kadar Polutan Dengan Menggunakan Jaringan Syaraf Tiruan (Jst) Untuk Pemantauan Kualitas Udara Di Kota Surabaya," *Semin. Nas. Tek. Kim. Soeardjo Brotohardjono IX*, pp. 1-11, 2012.
- [7] M. Venkatadri, "A Novel Air Quality Prediction Model Using Artificial Neural Networks," *Int. J. Eng. Res. Technol.*, vol. 3, no. 2, pp. 2973-2978, 2014.
- [8] E. G. Dragomir, "Air Quality Index Prediction using K-Nearest Neighbor Technique," *Bulletin of PG University of Ploiesti, Series Mathematics, Informatics, Physics LXII*, vol. 1, no. 1, pp. 103-108, 2010.
- [9] M. Oprea, M. Popescu, S. F. Mihalache, and E. G. Dragomir, "Data Mining and ANFIS Application to Particulate Matter Air Pollutant Prediction," *9th Int. Conf. Agents Artif. Intell. (ICAART 2017)*, vol. 1, no. Icaart, pp. 551-558, 2017.
- [10] R. E. Putra and T. Indriyani, "Penerapan Aturan Asosiasi Dengan Algoritma Apriori Untuk Analisis Polutan Udara Di Surabaya," *Semin. Nas. Sist. Inf. Indones.*, no. November, pp. 253-258, 2015.
- [11] F. Wafiyah, N. Hidayat, and R. S. Perdana, "Implementasi Algoritma Modified K-Nearest Neighbor (MKNN) untuk Klasifikasi Penyakit Demam," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 1, no. 10, pp. 1210-1219, 2017.
- [12] H. Parvin, H. Alizadeh, and B. Minati, "A Modification on K-Nearest Neighbor Classifier," *Glob. J. Comput. Sci. Technol.*, vol. 10, no. 14, pp. 37-41, 2010.
- [13] Okfalisa, I. Gazalba, Mustakim, and N. G. I. Reza, "Comparative analysis of k-nearest neighbor and modified k-nearest neighbor algorithm for data classification," *Proc. - 2017 2nd Int. Conf. Inf. Technol. Inf. Syst. Electr. Eng. ICITISEE 2017*, vol. 2018-Janua, pp. 294-298, 2018.
- [14] Z. S. Putri, R. Regasari, and M. Putri, "Deteksi Autisme pada Anak Menggunakan Metode Modified K-Nearest Neighbor (MKNN)," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 1, no. 3, pp. 241-248, 2017.
- [15] H. Parvin, H. Alizadeh, and B. Minaei-bidgoli, "MKNN: Modified K-Nearest Neighbor," *Proc. World Congr. Eng. Comput. Sci. 2008*, pp. 22-25, 2008.
- [16] F. D. Astuti, D. E. Ratnawati, and A. W. Widodo, "Deteksi Penyakit Kucing dengan Menggunakan Modified K-Nearest Neighbor

- Teroptimasi (Studi Kasus : Puskesmas Klinik Hewan dan Satwa Sehat Kota Kediri),” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 1, no. 11, pp. 1295–1301, 2017.
- [17] T. H. Simanjuntak and W. F. Mahmudy, “Implementasi Modified K-Nearest Neighbor Dengan Otomatisasi Nilai K Pada Pengklasifikasian Penyakit Tanaman Kedelai,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 1, no. 2, pp. 75–79, 2017.
- [18] S. Mutrofin, A. Izzah, A. Kurniawardhani, and M. Masrur, “Optimasi teknik klasifikasi modified k nearest neighbor menggunakan algoritma genetika,” *J. Gamma*, vol. 10, no. 1, pp. 130–134, 2014.
- [19] R. Sulistiyorini and W. F. Mahmudy, “Penerapan algoritma genetika untuk permasalahan optimasi distribusi barang dua tahap,” *Repository Jurnal Mahasiswa PTIK Universitas Brawijaya*, vol 5, no. 12, pp. 1-12, 2015.
- [20] W. J. Shudiq, “Penerapan K-Nearest Neighbor Berbasis Algoritma Genetika Untuk Klasifikasi Mutu Padi Organik,” *Pros. SNATIF*, pp. 121–126, 2017.