

## MENCARI LOKASI FASILITAS KESEHATAN (FASKES) RUJUKAN TERDEKAT BERBASIS VORONOI DIAGRAM

**Slamet Sudaryanto N\*, Sudaryanto**

Fakultas Ilmu Komputer, Universitas Dian Nuswantoro Semarang

Jl. Imam Bonjol No. 2015-207, Kota Semarang 50131

\*Email: slametalica301@dsn.dinus.ac.id

### Abstrak

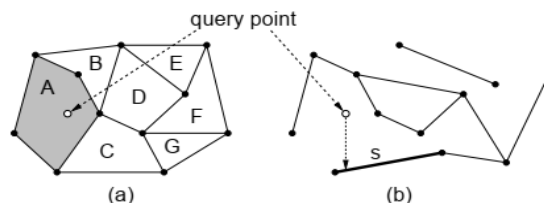
*Istilah lokasi mengacu pada posisi suatu titik lokasi (point location) yang relatif terhadap subdivisi geometris atau sekumpulan objek geometris yang terputus-putus. Contoh yang paling terkenal adalah masalah lokasi titik, di mana pembagian ruang menjadi daerah-daerah yang terpisah diberikan, dan masalahnya adalah mengidentifikasi daerah mana yang berisi titik kueri tertentu. Masalah ini banyak digunakan di bidang-bidang seperti grafik komputer, sistem informasi geografis, dan robotika. Lokasi titik adalah juga digunakan sebagai metode untuk pencarian kedekatan, bila diterapkan bersama dengan diagram voronoi. Pada tulisan paper ini akan membahas metode identifikasi lokasi titik fasilitas kesehatan (faskes) rujukan seperti rumah sakit, poliklinik, puskesmas terdekat. Seringkali pengguna mencari lokasi faskes rujukan terdekat akan memakan waktu yang lama, terutama ketika ada ribuan pengguna yang menggunakan aplikasi secara bersamaan. Data obyek faskes sebagai obyek titik pencarian selalu tumbuh, begitu juga perangkat pencarian bergerak sebagai mobile navigation. Sehingga untuk memenuhi kemudahan pencarian data terutama untuk pencari data melalui perangkat mobile diperlukan pengelolaan data dengan baik dan benar. Studi ini ditunjukkan dengan menerapkan beberapa metode pencarian terutama untuk pencarian berbasis voronoi pada mobile navigator. Terdapat metode yang digunakan untuk pencarian yaitu K-d Trees, namun performa yang dihasilkan masih belum memuaskan. Metode lain yang diusulkan adalah algoritme Voronoi Continuous K Nearest Neighbor (VCKNN) yang menggunakan diagram Voronoi untuk membantu pencarian nodes sebagai obyek pada data spasial. Penggunaan algoritme VCKNN merupakan pengembangan dari algoritma Continuous K Nearest Neighbor (KNN). VCKNN dalam melakukan pencarian lokasi data di dalam poligon Voronoi memiliki cara pembagian node (split nodes) seperti dalam metode K-d Trees. Pada penelitian ini akan dilakukan perbandingan antara algoritme VCKNN dan K-d Trees dalam mengelola pencarian lokasi obyek tertentu.*

**Kata kunci:** point location, voronoi diagram, k-d Trees, KNN, VCKNN

### 1. PENDAHULUAN

Masalah lokasi titik planar adalah salah satu masalah query yang paling mendasar dalam geometri komputasi. Pertimbangkan grafik garis lurus planar  $S$ . Ini adalah grafik tidak berarah, digambar pada bidang, yang ujung-ujungnya adalah segmen garis lurus yang memiliki bagian dalam yang saling berpasangan. Tepi  $s$  membagi bidang menjadi (mungkin tak terbatas) daerah poligonal, yang disebut permukaan. Selanjutnya, struktur seperti itu akan disebut sebagai subdivisi poligonal. Sepanjang, kami membiarkan  $n$  menunjukkan kompleksitas kombinatorial  $S$ , yaitu, jumlah total simpul, tepi dan permukaan (Edelsbrunner, 1987).

Masalah lokasi titik adalah untuk melakukan praproses subdivisi poligonal  $S$  di bidang menjadi struktur data sehingga, dengan titik kueri  $q$ , bidang poligonal dari subdivisi yang berisi  $q$  dapat dilaporkan dengan cepat. Masalah ini adalah generalisasi alami dari masalah pencarian biner dalam ruang 1 dimensi, di mana bidang subdivisi sesuai dengan interval antara nilai kunci 1 dimensi. Dengan analogi kasus 1-dimensi, tujuannya adalah untuk melakukan praproses subdivisi menjadi struktur data berukuran  $O(n)$  sehingga pertanyaan lokasi titik dapat dijawab dalam waktu  $O(\log n)$ .



**Gambar 1. Ilustrasi (a) lokasi titik pencarian dan (b) kueri pemotretan sinar vertikal**

Diberikan titik query  $q$ , masalahnya adalah menentukan ruas garis dari  $S$  yang terletak vertikal di bawah  $q$ . Pada Gambar 1(b), segmen garis  $s$  akan dilaporkan sebagai wilayah atau bidang yang dilalui atau dilewati. Diagram voronoi adalah metode untuk membagi wilayah atau ruang diantara sekumpulan titik-titik yang menetapkan area untuk masing-masing. Properti utama dari wilayah ini adalah bahwa setiap area atau wilayah mewakili ruang dimana titik-titik tersebut adalah tetangga terdekat (nearest neighbor). Dengan demikian, setiap titik pada wilayah tersebut telah dipasangkan dengan *site* yang terdekat (Berg et al., 2008). Studi ini mengusulkan metode baru berdasarkan versi diskrit dari diagram voronoi yang adaptif yang memungkinkan untuk membagi ruang dua dimensi ke dalam zona ukuran tertentu, dengan mempertimbangkan posisi dan berat masing-masing rasio pelayanan kesehatan. Metode yang sudah dikembangkan dalam pencarian titik lokasi pada bidang satu dimensi yaitu  $K$ -d Trees, namun performa yang dihasilkan masih belum memuaskan. Pendekatan *Continuous K Nearest Neighbor* (CKNN) juga menarik minat beberapa peneliti. Untuk menemukan split node, semua pendekatan CKNN yang ada membagi jalur kueri menjadi segmen, menemukan hasil KNN untuk dua node akhir dari setiap segmen, dan kemudian, untuk setiap segmen, menemukan node split. Satu segmen jalan dimulai dari persimpangan dan berakhir di persimpangan lain. Untuk setiap segmen, proses KNN dipanggil untuk menemukan node split untuk setiap segmen. Zona terhubung secara geografis menggunakan metrik berdasarkan jalur terpendek (Yan, H., & Weibel, R., 2008).

## 2. DIAGRAM VORONOI

Diagram Voronoi menunjukkan pembagian suatu wilayah tertentu menjadi beberapa bagian yang disebut *cell*, di mana masing-masing bagian berisi satu titik lokasi (*site*). Setiap titik di dalam suatu *cell* memiliki jarak lebih dekat ke *site* yang berada di dalam *cell* tersebut dibandingkan dengan *site* lainnya pada wilayah tersebut. Dengan demikian, setiap titik pada wilayah tersebut telah dipasangkan dengan *site* yang terdekat (Berg et al., 2008) di suatu wilayah lebih dekat ke *site* yang sesuai dengan wilayahnya daripada *site* yang lain. Konsep utama dari diagram voronoi ditemukan oleh Reitsma, R. and Trubin, S., (2007), “Diberikan sejumlah titik-titik yang berbeda dalam ruang Euclidean 2-D, diagram Voronoi dari set titik adalah kumpulan daerah yang membagi bidang dan semua lokasi di satu wilayah (kecuali batas wilayah) lebih dekat ke titik yang berhubungan daripada ke titik lain”. Sedangkan secara konsep matematika ditemukan oleh Novaes (2007): “Diberikan satu himpunan titik yang berbeda  $P = \{P_1, P_2, \dots, P_m\}$  didalam ruang kontinu (bidang), maka akan berusaha untuk melibatkan semua titik lain dari ruang dengan himpunan tertutup titik  $m$  anggota terdekat dari himpunan  $P$  adalah himpunan generator diagram voronoi, dengan  $m \geq 2$ ”. Misalkan  $p = (p_1, \dots, p_n)$  dan  $q = (q_1, \dots, q_n)$  adalah dua titik pada  $R^n$ . Pembagi dari  $p$  dan  $q$  terdiri dari titik-titik  $x = (x_1, \dots, x_n)$  dimana

$$\|p - x\| = \|q - x\| \iff \|p - x\|^2 = \|q - x\|^2 \iff \|p\|^2 - \|q\|^2 = 2(p - q)^T x$$

Karena  $p$  dan  $q$  berbeda, ini adalah persamaan hyperplane, dilambangkan dengan  $H(p_i, p_j)$  ruang paruh tertutup yang dibatasi oleh garis bagi  $p_i$  dan  $p_j$  yang berisi  $p_i$ . Dalam  $R^2$ ,  $H(p_i, p_j)$  adalah setengah bidang. Diagram voronoi menunjukkan pembagian suatu wilayah tertentu menjadi beberapa bagian yang disebut *cell*, dimana masing-masing bagian berisi satu titik lokasi yang disebut *site* (Tiede, D. & Strobl, J., 2006). Didefinisikan  $V(p_i)$ , sel Voronoi untuk  $p_i$ , sebagai himpunan titik  $q$  yang lebih dekat ke  $p_i$  dibandingkan ke *site* lainnya. Sehingga, sel voronoi untuk  $p_i$  didefinisikan dengan persamaan :

$$V(p_i) = \{q \mid \text{jarak}(p_i, q) < \text{jarak}(p_j, q), \text{ untuk } j \neq i\} \text{ atau } P_i = \{x \mid d(x, p_i) \leq d(x, p_j)\}$$

$$V(pi) = \bigcap_{j \neq i} H(pi, pj)$$

### 3. PENCARIAN TITIK LOKASI BERDASARKAN k-NN

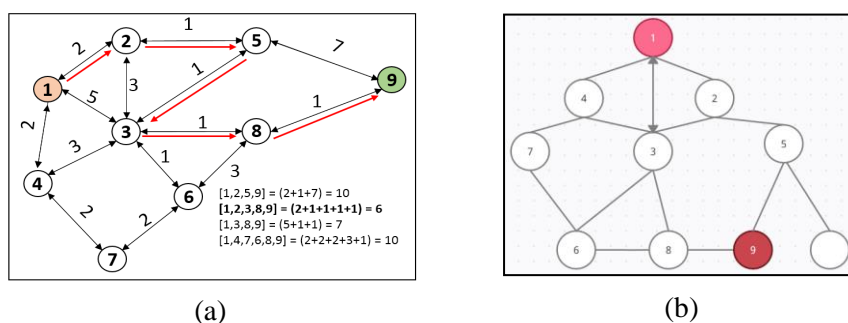
Menurut definisi, diagram Voronoi membagi ruang menjadi sel-sel menurut situs mana yang paling dekat. Jadi, untuk menentukan lokasi terdekat, cukup menghitung diagram Voronoi dan menghasilkan struktur data lokasi titik untuk diagram Voronoi. Dengan cara ini, kueri tetangga terdekat direduksi menjadi kueri lokasi titik. Ini menyediakan metode waktu kueri  $O(n)$  ruang dan  $O(\log n)$  yang optimal untuk menjawab kueri lokasi titik di bidang. Sayangnya, solusi ini tidak dapat digeneralisasi dengan baik ke dimensi yang lebih tinggi. Kompleksitas kombinatorial kasus terburuk dari diagram Voronoi dalam dimensi  $d$  tumbuh sebagai  $O(n^{\lfloor d/2 \rfloor})$ , dan struktur data lokasi titik optimal tidak diketahui ada di dimensi yang lebih tinggi. Ada banyak cara untuk mendefinisikan pengertian persamaan matematika sebagai formula model. Karena fokus dari paper ini adalah pada pendekatan geometris, kita akan mengasumsikan bahwa kedekatan didefinisikan dalam bentuk jarak Euclidean yang telah diketahui. Sebagian besar hasil yang akan disajikan di bawah ini dapat digeneralisasikan ke metrik *Minkowski*, di mana jarak antara dua titik  $p$  dan  $q$  didefinisikan dalam persamaan dibawah ini.

$$\text{dist}_{m(p,q)} = \left( \sum_{k=i=1}^d |p_i - q_i|^2 \right)^{1/n}$$

Kasus  $m = 2$  adalah jarak Euclidean, kasus  $m = 1$  adalah Jarak Manhattan, dan kasus pembatas  $m = 1$  adalah jarak maksimal. Dalam aplikasi geometris tipikal, dimensi  $d$  diasumsikan sebagai konstanta tetap dan dalam ruang metrik yang kompleks atau nongeometris (P. Indyk, 2004). Sudah tentu, bahwa tanpa preprocessing apapun, masalah pencarian tetangga terdekat tidak dapat diselesaikan dalam waktu  $O(n)$  melalui pencarian brute-force sederhana (Friedman et al, 2005).

#### 3.1. Algoritma VCKNN

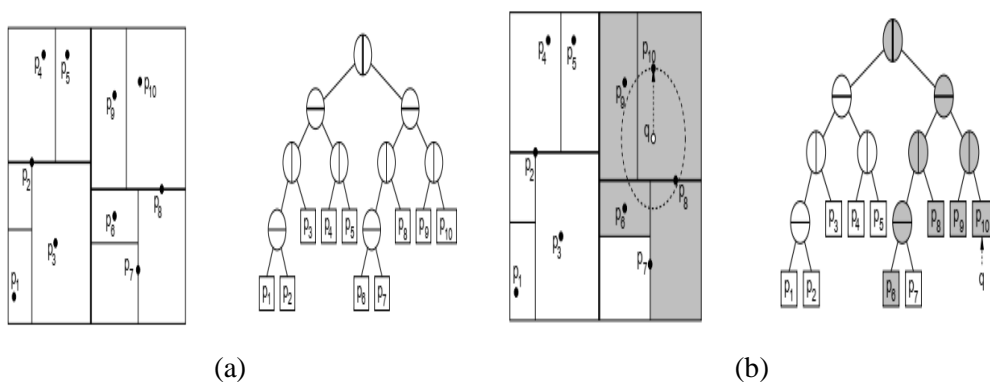
Dalam VCKNN sebagai pengembangan metode CKNN berupaya mengeliminasi kelemahan sebagai solusi untuk mencari titik lokasi, VCKNN berbasis voronoi diagram juga memberikani visibilitas point of interest (POI) atau titik minat mana yang bergerak keluar atau masuk daftar dan di posisi mana node akan menjadi node terpisah (split node). Model algoritma pada pendekatan VCKNN yang didukung dengan diagram network voronoi adalah meliputi preposisi dan bukti (Stergiopoulos et al, 2015). Preposisi merupakan langkah pertama sebagai generator dari poligon Voronoi yang mencakup titik *query* harus menjadi tetangga terdekat dari titik *query*. Sedangkan bukti merupakan keterangan yang dengan sendirinya karena poligon mendefinisikan daerah dimana ada suatu titik di daerah tersebut yang mendekati generator poligon dibandingkan dengan *generator* lainnya. Langkah selanjutnya pembagian node (split nodes) dalam diagram network Voronoi ditentukan oleh semua titik perbatasan yang berpotongan dengan jalur *query* dan tepi generator adalah split nodes. Jelas bahwa, ketika jalur *query* mencapai tepi *generator*, NN pertama akan berubah karena jarak ke *generator* tepi bersama memiliki nilai yang sama.



Gambar 2. Dijkstra Tree (a) split nodes (b)

### 3.2. Algoritma K-d Tree

Untuk kelas pendekatan yang paling populer untuk pencarian tetangga terdekat atau k-nearest neighbor (k-NN) akan melibatkan semacam subdivisi spasial hierarkis. Misalkan  $S$  menyatakan himpunan  $n$  titik di  $R^d$  yang pertanyaannya harus dijawab (Skiena, S.S., 2008). Dalam pendekatan seperti itu, seluruh ruang dibagi lagi menjadi wilayah yang lebih kecil secara berurutan, dan hierarki yang dihasilkan diwakili oleh pohon berakar (k-d Tree). Setiap simpul pohon mewakili wilayah ruang, yang disebut sel. Secara implisit, setiap node mewakili subset dari titik-titik  $S$  yang terletak di dalam selnya. Akar pohon dikaitkan dengan seluruh ruang dan seluruh himpunan titik  $S$ . Untuk beberapa simpul *split nodes*, jika jumlah titik  $S$  yang terkait dengan  $u$  kurang dari suatu konstanta, maka simpul ini dinyatakan sebagai sehelai daun dari pohon. Jika tidak, sel yang terkait dengan  $u$  secara rekursif dibagi menjadi sub-sel yang lebih kecil (mungkin tumpang tindih) menurut beberapa aturan pemisahan. Kemudian titik-titik terkait dari  $S$  didistribusikan di antara anak-anak ini menurut sub-sel mana mereka berada.

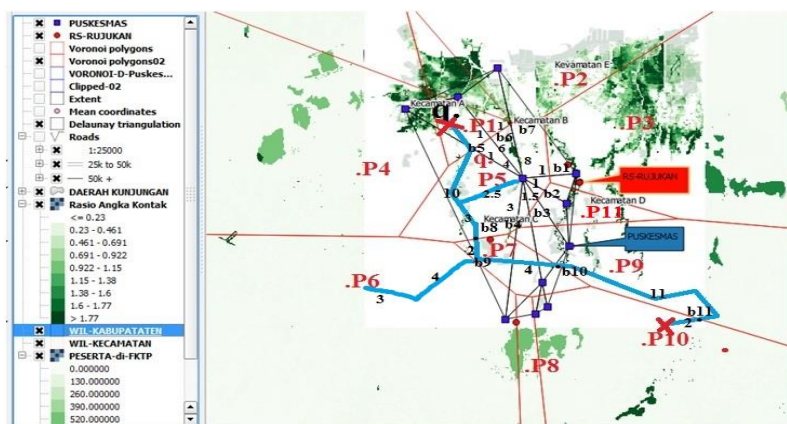


Gambar 3. (a) k-d Tree dari sekumpulan titik pada data spasial terkait (b) Pencarian tetangga terdekat di k-d Tree, node yang diarsir yang dikunjungi

## 4. HASIL DAN PEMBAHASAN

### 4.1. Analisis yang dilakukan terhadap algoritme VCNN dan K-D Tree

Adalah dengan menggunakan data eksperimen terhadap pencarian dari titik  $q$  pada  $P5$  sebagai titik awal dan titik  $x$  pada  $P10$  sebagai titik tujuan seperti pada Gambar 4 merupakan hasil ilustrasi terhadap wilayah yang memiliki berbagai jalur jalan dan faskes kesehatan berupa puskesmas dan rs rujukan. Mencari jumlah jarak yang paling kecil dengan menggabungkan nilai yang memiliki alternative jalur yang sama arahnya dengan tujuan . Pada Gambar 4, titik  $q$  diilustrasikan dengan *node 1* sedangkan titik  $x$  diilustrasikan dengan tujuan pencarian lokasi.



Gambar 4. Penerapan VCKNN Berbasis Voronoi Diagram

Bagian ini tidak hanya menjelaskan pembahasan proses VCKNN, tetapi juga membandingkan dengan K-d Trees pada penerpan diagram voronoi dalam rangka mencari titik lokasi. Kami akan membuat fungsi perhitungan sepotong-sepotong (segmen) dan menggambar garis di bagan agar

mudah dipahami. Gambar 4 menunjukkan sebuah contoh eksperimen. Kuerinya adalah untuk menemukan titik lokasi di sepanjang jalur kueri (P5 dan P10), ditampilkan sebagai garis biru tebal, yang dimulai dari q pada p5 dan berakhir di P10. Border V (P5) yaitu b1, b2, b3, b4, b5, b6, b7 dan jalur dari P1 ke titik perbatasan juga ditampilkan. Pada gambar tersebut ditemukan candidate interest point (CS) dari polygon P5 adalah {P1, P2, P3, P4, P6, P7, P11}. Dengan ditemukannya CS tersebut kemudian dilakukan pencarian dari titik lokasi q yang berada pada V(P5) menuju titik lokasi pada V(P10). Pencarian dari titik q menuju tepi polygon tetangga terdekat dari P5.

#### 4.4.1. Analisis Algoritme VCKNN

Sejumlah node split pertama adalah node persimpangan antara poligon Voronoi dan jalur bergerak sebagai batas atau borders (Solis, et all, 2009). Dalam hal ini, split nodes = {b8, b9, b10, b11}. Split nodes b8 adalah titik batas antara poligon Voronoi V (P5) dan V (P7), split node b9 adalah titik batas antara V (P6) dan V (P7), dan split node b10 adalah titik perbatasan antara V (P7) dan V (P9), serta split node b11 adalah titik perbatasan antara V (P9) dan V (P10). Berikut ini ada hasil identifikasi kelompok NN dan melalui split nodes V(P)

**Tabel 1. Jumlah split nodes Masing-masing V**

V(P)	Nearest Neighbour	Split Nodes
P5	{P1, P2, P3, P4, P6, P7, P11}	3
P7	{P5, P6, P8, P10, P9}	3
P9	{P5, P7, P10, P11}	2
P10	{P7, P8, P9}	1

Perhatikan bahwa hasil NN pertama adalah P5 dengan rentang jarak 0.0 dan 9.5, NN kedua P7 dengan rentang jarak 9.5 dan 15.5, NN ketiga P9 dengan rentang jarak 15.5 dan 26.5, dan NN keempat P10 dengan rentang jarak 26.5 dan 28.5. Singkatnya, kita dapat menulis hasil NN pertama sebagai suatu pencarian lokasi maka NN pertama sebagai adalah {P5(0.0–9.5), P7(9.5–15.5), P9(15.5–26.5), P10(26.5–28.5)}. Semua rentang jarak adalah jarak dari titik kueri awal. Artinya, ketika titik kueri q bergerak dari 0 ke 10, P5 adalah NN pertama, dan ketika q bergerak dari 10 ke 12, P7 akan menjadi NN selanjutnya, dan seterusnya sampai titik lokasi P10 ditemukan. Kemudian untuk V (P5), V (P7), dan V (P9), serta V(P10) lakukan langkah berikut. Ambil V (P5) sebagai langkah NN pertama :

**Tabel 2. Pergerakan Setiap Titik Lokasi Perbatasan Awal Pada P5**

Pergerakan jarak q (km)	b1	b2	b3	b4	b5	b6	b7	b8
0.0	5.0	5.0	5.5	7.0	1.0	10	12	9.5
4.0	1.0	1.0	1.5	3.0	5.0	6.0	8.0	5.5
6.5	3.5	3.5	4.0	5.5	10	11	12	3.0
9.5	6.5	6.5	7.0	8.5	13	14	15	0

Jalur yang didapat dalam pencarian adalah q, (P5, P7, P9, P10) melalui Split Nodes = {b8, b9, b10, b11}. Split nodes yang digunakan hanya pada jalur peta yang memiliki kedekatan dengan V(P) berikutnya. gunakan semua untuk dibandingkan untuk mendapatkan jarak terdekat berdasarkan semua segmen.

#### 4.4.2. Analisis Algoritme k-d Tree

Sejumlah node split pertama adalah semua segmen sejumlah jalur persimpangan pada poligon Voronoi dan jalur bergerak sebagai batas atau borders (Ricca et all, 2008). Dalam hal ini, split nodes = {b1, b3, b2, b8, b9, b10, b11}. Split nodes b1 adalah titik batas antara V (P5) dan V (P3), Split nodes b2 adalah titik batas antara V (P5) dan V (P11), Split nodes b3 adalah titik batas antara V (P5) dan V (P11), Split nodes b8 adalah titik batas antara poligon Voronoi V (P5) dan V (P7), split node b9 adalah titik batas antara V (P6) dan V (P7), dan split node b10 adalah titik perbatasan antara

V (P7 ) dan V (P9), serta split node b11 adalah titik perbatasan antara V (P9 ) dan V (P10). Berikut ini ada hasil identifikasi kelompok NN dan melalui split nodes V(P)

**Table 3. Jumlah splite nodes Masing-masing V**

V(P)	Nearest Neighbour	Split Nodes
P3	{P2, P5, P11}	2
P5	{P1, P2, P3,P4,P6,P7, P11}	7
P7	{P5, P6, P8, P10, P9}	6
P9	{P5, P7, P10, P11}	7
P10	{P7, P8, P9}	4
P11	{P3, P5, P9}	3

Perhatikan bahwa hasil NN pertama adalah P3 dengan rentang jarak 0.0 dan 3.0, kedua P5 dengan rentang jarak 3.0 dan 9.5, NN ketiga P7 dengan rentang jarak 9.5 dan 15.5, NN keempat P9 dengan rentang jarak 15.5 dan 26.5, dan NN keelima p10 dengan rentang jarak 26.5 dan 28.5, serta keenam P11 dengan rentang jarak 28.5 dan 31.5. Singkatnya, kita dapat menulis hasil NN pertama sebagai suatu pencarian lokasi maka NN pertama sebagai adalah = {P3(0.0-3.0), P5(3.0-9.5), P7(9.5-15.5), P9(15.5-26.5), P10(26.5-28.5), P11(28.5-31.5)}.

Semua rentang jarak adalah jarak dari titik kueri awal. Artinya, ketika titik kueri q bergerak dari 0 ke 10, P5 adalah NN pertama, dan ketika q bergerak dari 10 ke 12, P7 akan menjadi NN selanjutnya, dan seterusnya sampai titik lokasi P10 ditemukan. Kemudian untuk V (P5), V (P7), dan V (P9), serta V(P10) lakukan langkah berikut. Ambil V (P5) sebagai langkah NN pertama :

**Tabel 4. Pergerakan Setiap Titik Lokasi Perbatasan Awal Pada P5**

Pergerakan jarak q (km)	b1	b2	b3	b4	b5	b6	b7	b8
0.0	5.0	5.0	5.5	7.0	1.0	10	12	9.5
2.5	2.0	3.0	4,5	2.0	10	8.0	6.5	60
4.0	1.0	1.0	1.5	3.0	5.0	6.0	8.0	5.5
6.5	3.5	3.5	4.0	5.5	10	11	12	3.0
9.5	6.5	6.5	7.0	8.5	13	14	15	0

Jalur yang didapat dalam pencarian adalah q, (P3,P5,P7,P9,P10, P11) melalui Split Nodes = {b1, b2,b3,b4,b5, b6, b7,b8, b9, b10, b11}. Seplit nodes yang ada pada titik awal di gunakan semua untuk dibandingkan untuk mendapatkan jarak terdekat berdasarkan semua segmen.

#### 4.4.3. Evaluasi dan Perbandingan Hasil (VKCNN dan K-DTree)

Evaluasi yang dilakukan adalah membandingkan algoritme VCKNN dan K-DD Trees pada aspek, jumlah *split nodes*, *runtime* dan pembagian segmentasi (*segmentation*) dengan melakukan eksperimen sebanyak 100 sampaidengan 500 titik lokasi eksperimen pencarian obyek pada bidang voronoi diagram. Data pada peta digital eksperimen yang memiliki jaringan jalan dan lokasi faskes puskesmas dan rumahsakit rujukan di gabungan dengan data eksperimen pada bidang voronoi diagram. Pencarian antara dua titik ini dibedakan dengan jumlah segmentasi yang dilalui mulai dari titik query pada shell diagram voronoi yang berbeda dengan titik lokasi obyek yang dicari secara random. Pada Tabel 6 dan 7 dibawah ini merupakan hasil dari beberapa percobaan yang dilakukan pada pencarian dengan pendekatan VCKNN dan KD-Tree yang dijadikan acuan dalam melakukan perbandingan pada nilai K=5.

**Tabel 5. Hasil pengujian kinerja menggunakan metode pencarian VCKNN Dengan Nilai K=5**

Jumlah Location Point	Jumlah Splate Nodes	Runtime (ms)	Rata-rata Segment Devision
100	9	0.22	4
200	14	0.30	7
300	16	0.32	8
400	22	0.34	11
500	27	0.37	13

**Tabel 6. Hasil pengujian kinerja menggunakan metode pencarian KD-Trees Dengan Nilai K=5**

Jumlah Location Point	Jumlah Splate Nodes	Runtime (ms)	Rata-rata Segment Devision
100	29	0.90	6
200	31	0.93	8
300	34	0.95	9
400	36	0.96	12
500	37	0.97	15

## 5. KESIMPULAN

Dari hasil pengujian dapat disimpulkan bahwa penggunaan metode VCKNN dapat mengurangi waktu yang diperlukan saat melakukan pencarian titik lokasi pada bidang 2 dimensi (memiliki jaringan jalan dan titik lokasi obyek) seperti data lokasi pada peta digital disbanding KD-Tree. Hal ini berarti pendekatan VCNN memiliki response time yang tercepat dalam melakukan inisialisasi pembentukan rute pencarian dibandingkan dengan KD-Tree. Waktu yang diperlukan untuk klasifikasi juga relatif konstan walaupun nilai k berubah-ubah. Implementasi pencarian dengan pendekatan VCNN juga memiliki jumlah splite nodes, dari hasil pengujian didapatkan waktu untuk inisialisasi dan pembentukan jumlah splite nodes berbanding lurus dengan waktu runtime. Semakin sedikit splite nodes yang terbentuk semakin kecil waktu runtimenya. Penelitian ini dapat dilanjutkan atau dikembangkan dengan menguji waktu yang diperlukan untuk klasifikasi pada truktur data VCKNN maupun KD-Tree.

## DAFTAR PUSTAKA

- Berg, M.d., Cheong, O., Kreveld, M.v. and Overmars, M., (2008), Computational Geometry: Algorithms and Applications. 3rd ed. Berlin: Springer.
- H. Edelsbrunner. Algorithms in Combinatorial Geometry, volume 10 of EATCS Monographs on Theoretical Computer Science. Springer-Verlag, Heidelberg, West Germany, 1987.
- J. H. Friedman, F. Baskett, and L. J. Shustek. An algorithm for finding nearest neighbors. IEEE Trans. Comput., C-24(10), 2005.
- Novaes, A. G. (2007). Resolução de Problemas de Transporte com Diagramas de Voronoi, XXI ANPET, Panorama Nacional da Pesquisa em Transportes. Associação Nacional de Pesquisa e Ensino em Transportes, Rio de Janeiro. Brazil.
- P. Indyk. Nearest neighbors in high-dimensional spaces. In Jacob E. Goodman and Joseph O'Rourke, editors, Handbook of Discrete and Computational Geometry. CRC Press LLC, Boca Raton, FL, 2004. (To appear).
- Reitsma, R. and Trubin, S., (2007), Information space partitioning using adaptive Voronoi diagrams. Information Visualization, 6, 123-138.
- Ricca, F, Scozzari, A., and Simeone, B., (2008), Drawing political districts by weighted Voronoi regions and local search. Mathematical and Computer Modelling, 48, 1468-1477.
- Solis, N., Rios-Mercado, R.Z., and Alvarez, A.M.,(2009), *Modelando sistemas territoriales con programacion entera*. Ingenierias, 12 (44), 7-15.

- Skiena, S.S., (2008), *The Algorithm Design Manual*. 2nd ed. London: Springer.
- Stergiopoulos, Y, Thanou, M, Tzes, (2015), A. *Distributed collaborative coverage-control schemes for non-convex domains*. *IEEE Trans. Autom. Control* 2015, 60, 2422–2427.
- Tiede, D. & Strobl, J., (2006), Polygon-based regionalisation in a GIS environment [online]. *In: E. Buhmann, S. Ervin, L. Jorgensen, and J. Strobl, eds. Trends in knowledge-based landscape modeling*. Heidelberg: Wichmann, 54-59. Available from: [http://www.masterla.de/conf/pdf/conf2006/23Tiede\\_L.pdf](http://www.masterla.de/conf/pdf/conf2006/23Tiede_L.pdf). [diakses 02 Mei 2019].
- Yan, H., & Weibel, R., (2008), *An algorithm for point cluster generalization based on the Voronoi diagram*. *Computers and GeoSciences*, vol. 34, no. 8, pp. 939–954.
- Yongxi, G., Guicai, L., Yuan, T., & Yaoyu, L., (2012), *A vector-based algorithm to generate and update multiplicatively weighted Voronoi diagrams for points, polylines, and polygons*. *Computers & Geosciences*, pp. 118-125.