

**LAPORAN KEMAJUAN
PENELITIAN DASAR UNGGULAN PERGURUA TINGGI (PDUPT)**



Pengembangan Metode Pencarian Probabilistic Threshold k Aggregate Nearest
Neighbor (PTANN)
Berdasarkan Data Dinamis Pada Diagram Voronoi

Tahuk ke 2 dari rencana 2 tahun

Slamet Sudaryanto Nurhendratno, ST.M.Kom (0607087101/Ketua)
Sudaryanto, M.Kom (0621076701 /Anggota)

Dibiayai Oleh :

Direktorat Riset dan Pengabdian Masyarakat
Direktorat Jendral Penguatan Riset dan Pengembangan
Kementerian Riset, Teknologi dan Pendidikan Tinggi
Sesuai dengan Kontrak Penelitian Tahun Anggaran 2020
Nomor : 014/L6/PG/SP2H/PL.II/2020

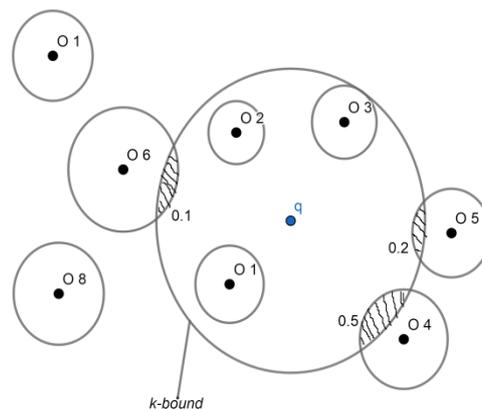
**UNIVERSITAS DIAN NUSWANTORO SEMARANG
SEPTEMBER - 2020**

Pengisian poin C sampai dengan poin H mengikuti template berikut dan tidak dibatasi jumlah kata atau halaman namun disarankan ringkas mungkin. Dilarang menghapus/memodifikasi template ataupun menghapus penjelasan di setiap poin.

C. HASIL PELAKSANAAN PENELITIAN: Tuliskan secara ringkas hasil pelaksanaan penelitian yang telah dicapai sesuai tahun pelaksanaan penelitian. Penyajian meliputi data, hasil analisis, dan capaian luaran (wajib dan atau tambahan). Seluruh hasil atau capaian yang dilaporkan harus berkaitan dengan tahapan pelaksanaan penelitian sebagaimana direncanakan pada proposal. Penyajian data dapat berupa gambar, tabel, grafik, dan sejenisnya, serta analisis didukung dengan sumber pustaka primer yang relevan dan terkini.

1. Hasil Analisa dan Evaluasi Metode Pencarian Probabilistic Threshold k-Aggregate Nearest Neighbor (PT-ANN).

Dalam metode pencarian obyek yang muncul seperti layanan berbasis lokasi, nilai - nilai item basis data secara alami tidak tepat karena selalu berubah dengan tidak pasti. Untuk database yang tidak pasti ini, kueri penting adalah Probabilistik k-Nearest-Neighbor Query (k-PNN), yang menghitung probabilitas set objek k untuk menjadi yang terdekat dengan titik kueri yang diberikan. Evaluasi kueri ini dapat menjadi komputasi yang komplks dan konskwensinya biaya I/O nya mahal, karena di sana adalah sejumlah besar set objek k, dan integrasi numerik diperlukan. Seringkali seorang pengguna mungkin tidak peduli dengan nilai probabilitas yang tepat. Misalnya, ia mungkin hanya perlu jawaban yang memiliki kepercayaan diri cukup tinggi. Oleh karena itu kami mengusulkan Probabilistic Threshold k-Nearest-Neighbor Query (T-k-PNN), yang mengembalikan set objek k yang memenuhi kueri dengan probabilitas lebih tinggi dari beberapa ambang batas T.



Gambar 1: Probabilistik k-NN Query (k-PNN) dengan k=3

Tiga langkah yang kami usulkan untuk ditangani permintaan (queries) ini secara efisien :

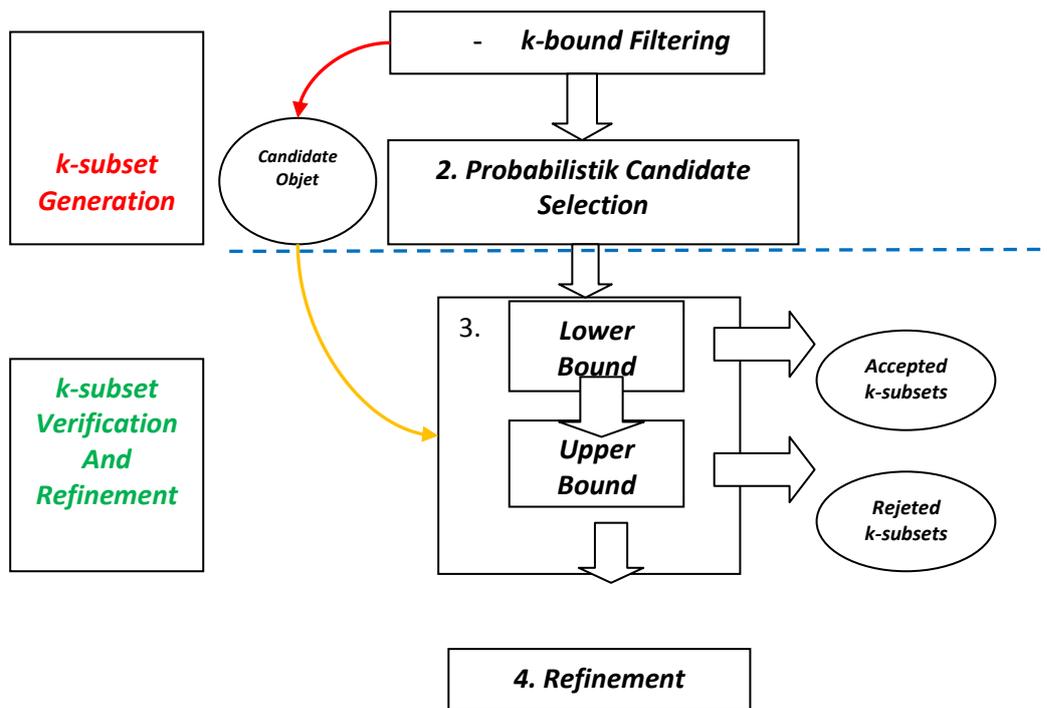
Pada tahap pertama, objek yang tidak dapat membentuk jawaban disaring atau di filter (k-bound filtering) dengan bantuan indeks spasial.

Langkah kedua, yang disebut seleksi kandidat probabilistik (Probabilistik Candidate Selection), memangkas sejumlah set kandidat secara signifikan untuk diperiksa.

Set yang tersisa dikirim untuk verifikasi, yang memperoleh kemungkinan jawaban di batas bawah dan atas, sehingga set kandidat dapat dengan cepat diputuskan apakah harus dimasukkan dalam jawaban.

Kami juga memeriksa struktur data yang efisien secara spasial yang mendukung metode ini. Solusi kami dapat diterapkan pada data yang tidak pasti dengan fungsi kepadatan probabilitas arbitrer.

Langkah-langkah tersebut dapat kami gambarkan dalam frame work sebagai berikut :



Gambar 2. Hasil Framework Probability Threshold k-Nearest-Neighbor Query (T-K-PNN)

Kami menyajikan tiga metode untuk memproses permintaan T -kPNN secara efisien, metode pertama, disebut penyaringan k-bound filtering, secara efektif menghapus semua objek yang tidak memiliki kesempatan untuk menjadi jawaban kueri. Hasil analisa dari gambar 1 (Probabilistik k-NN Query (k-PNN) dengan k=3), yang menunjukkan "k-bound" (sebagai lingkaran titik-titik yang berpusat pada q) yang benar-benar menutup tiga objek o1, o2 dan o3. Jari-jari dari 3-bound didefinisikan sebagai yang ketiga minimum jarak maksimal objek dari q (dalam hal ini contoh, jarak maksimum o3 dari q). Dengan k-bound, objek o7 dan o8 dapat segera dipangkas, karena tidak ada kesempatan untuk mengambil alih salah satu objek o1, o2 atau o3 untuk menjadi bagian dari jawaban pertanyaan 3-PNN. Secara umum, dengan k-bound, banyak objek dapat dihapus, dan seperti yang kita tunjukkan di paper, itu penggunaan dapat dengan mudah ditingkatkan ke indeks spasial (mis., R-tree). Untuk kenyamanan, kita menyebut objek yang tidak dipangkas oleh k-bound filtering (mis., yang tumpang tindih dengan k-bound) kandidat objek.

Setelah pemfilteran k-bound, kita masih perlu mempertimbangkan k-subsets bagian dari objek kandidat. Pada Gambar 1, misalnya, $C36 = 20$ set kardinalitas 3 mungkin perlu dipertimbangkan. Untuk mengurangi lebih lanjut ruang pencarian, kami mengusulkan metode kedua, yaitu Seleksi Kandidat Probabilistik atau Probabilistic Candidate Selection (PCS), yang dapat mendeteksi secara efisien k-subset (mis., subset basis data D dengan kardinalitas k) yang probabilitas kualifikasi kurang dari T, juga disebut tidak memenuhi syarat k-himpunan bagian. Sedangkan penyaringan k-bound memanfaatkan informasi jarak untuk pemangkas (pruning), PCS memanfaatkan informasi probabilitas data tidak pasti untuk menghapus himpunan bagian yang tidak memenuhi syarat. Alasan di balik PCS adalah bahwa mengingat kemungkinan objek kandidat itu terletak di dalam k-bound

(disebut probabilitas cutoff), probabilitas kualifikasi sub-k harus lebih rendah daripada produk dari probabilitas cutoff dari himpunan bagiannya. Pada Gambar 1, misalnya, probabilitas cutoff dari o4, o5, dan o6 masing-masing 0,5, 0,2 dan 0,1 (ditampilkan sebagai area yang diarsir). Probabilitas kualifikasi 3-subset {o2, o4, o5} harus lebih rendah dari produk cutoff probabilitas {o4, o5}, atau $0,5 \times 0,2 = 0,1$. Jika $T = 0,2$, maka {o2, o4, o5} dapat dipangkas. Berdasarkan fakta yang bermanfaat ini, algoritma PCS membangun k-subset dengan menumbuhkan daftar i-subsets sehubungan dengan i (di mana $i = 1, 2, \dots, k$). Pada setiap iterasi i, the produk dari probabilitas cutoff dari setiap i-subsets diperiksa apakah kurang dari T, dan i -subset dipangkas jika itu benar. Selain itu, kami juga merancang teknik, yang disebut "pemangkasan benih" ("seed-pruning"), untuk lebih lanjut meningkatkan kinerja PCS dengan menghapus k-subset yaang tidak memenuhi syarat dengan "seed" - objek yang berada di dalamnya k-bound. Apalagi metode kompresi data yang efisien itu menekan jumlah overhead penyimpanan perantara yang diperlukan oleh PCS disajikan. Eksperimen kami desain untuk menunjukkan bahwa PCS mengurangi a porsi signifikan k-himpunan bagian yang akan diperiksa dalam rangka melakukan kueri.

Metode ketiga, yang disebut verifikasi (verification), berguna untuk menangani k-subset yang tidak disaring (filtered) oleh dua metode sebelumnya. Teknik ini menentukan apakah k-subset adalah jawaban kueri, dengan memanfaatkan ketidakpastian pdf objek yang dikembalikan oleh penyaringan k-bound. Kami mengusulkan dua jenis verifikasi: batas bawah (lower-bound) dan batas atas (upper-bound) verifikasi, yang dengan cepat menghitung batas bawah dan atas probabilitas kualifikasi k-subset. Batas ini kemudian bisa digunakan untuk menentukan bagaimana k-subset harus ditangani. Sebagai contoh, k-subset dapat dihapus jika probabilitas batas atasnya lebih kecil dari T; itu harus dimasukkan dalam jawaban pertanyaan (query answer) jika itu probabilitas batas bawah lebih tinggi dari T. Kami akan menunjukkan desain rinci, analisis kompleksitas, serta bukti kebenaran untuk metode ini.

Gambar 2 menggambarkan kerangka solusi kami, yang terdiri dari empat langkah. Pertama, pemfilteran k-bound (k-bound Filtering) menghapus objek yang tidak boleh menjadi bagian dari k-tetangga terdekat q (k-nearest neighbor of q). Semua objek kandidat kemudian diteruskan ke PCS, yang memperoleh k-subset berdasarkan informasi probabilitas cutoff. Selanjutnya, batas bawah (atas) dari probabilitas kualifikasi dari k-subsets digunakan untuk menerima dan menolak k-subsets. Yang masih belum dapat ditentukan dikirim untuk disempurnakan, yang probabilitas pastinya dihitung. Meskipun perbaikan (refinement) mahal, itu dapat memanfaatkan informasi yang dihasilkan selama verifikasi. Jadi ini masih lebih cepat daripada menghitung probabilitas kualifikasi k-subset secara langsung.

Untuk meringkas, kami mengusulkan solusi yang efisien secara komputasi dan I / O untuk mengevaluasi permintaan T-k-PNN. Solusi kami mengurangi overhead I / O dengan menggunakan indeks spasial (mis., R-tree) untuk memangkas sejumlah besar objek. Untuk mengurangi overhead komputasi yang besar, kami mengusulkan PCS untuk mengurangi jumlah subset k yang akan diperiksa, serta verifikasi / penyempurnaan untuk menghindari perhitungan probabilitas yang tepat. Kami mengusulkan kerangka kerja untuk menghubungkan teknik-teknik ini untuk memberikan solusi yang efisien. Kami selanjutnya menyelidiki struktur data yang efisien penyimpanan untuk mendukung solusi kami. Eksperimen kami menunjukkan bahwa pendekatan kami dapat secara signifikan meningkatkan kinerja evaluasi kueri. Misalnya, pada $T = 0,1$ dan $k = 5$ waktu yang diperlukan oleh metode kami hanya 1,6% dari waktu yang dibutuhkan dengan menghitung probabilitas kualifikasi secara langsung.

2. Hasil Identifikasi Fungsi (algoritma) Probabilistic Threshold k-Aggregate Nearest Neighbor (PT k-ANN).

Berdasarkan Analisa dan evaluasi Evaluasi Metode Pencarian Probabilistic Threshold k-Aggregate Nearest Neighbor (PT k-ANN), maka didapatkan beberapa kebutuhan algoritmas sebagai berikut :

Algoritma k-bound Filtering

Algoritma PCS (Probabilistic Candidate Selection)

Algoritma distance of seed (seed_pruning)

Algoritma lower and upper bound verification

Algoritma perbaikan (refinement)

Identifikasi algoritma ini belum dikaitkan dengan kebutuhan fungsi voronoi diagram sebagai sebagai metode atau alat untuk implementasi klasifikasi pada database spasial.

3. Hasil Konsep Rancangan Metode k-bound Filtering pada T-k-PNN

Kami sekarang menyajikan semantik dari kueri T-k-PNN, Kemudian kami menjelaskan solusi sederhana untuk kueri ini. Misalkan $p(S) \in [0, 1]$ menjadi probabilitas bahwa elemen-elemen dari sub-k S adalah k tetangga terdekat dari titik kueri q (yaitu, probabilitas kualifikasi). Kemudian, T-k-PNN dapat didiskripsikan beberapa hal yang berkaitan dengan ambang batas (threshold) probabilitik.

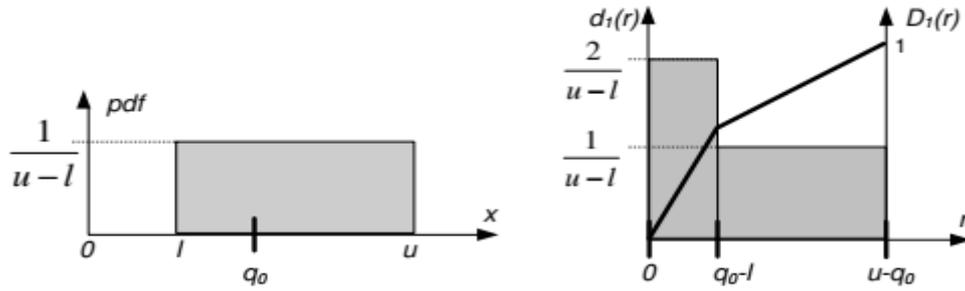
Ambang Batas Probabilitas k-NN Query (T - k-PNN) mengembalikan himpunan S, sedemikian sehingga $\{S \mid S \subseteq D \wedge |S| = k\}$ dan $p(S) \geq T$, di mana $T \in (0,1]$. Kami menyebut T parameter ambang, sebuah k-subset S diizinkan dikembalikan sebagai jawaban query jika probabilitas kualifikasinya tidak kurang dari T. Dibandingkan dengan k-PNN, kueri ini tidak mengembalikan probabilitas kualifikasi aktual S ke pengguna. Juga, kami selanjutnya dapat menggunakan kendala lain (mis., jumlah maksimum jawaban) untuk membatasi jumlah himpunan bagian-k yang dikembalikan kepada pengguna. Tabel dibawah ini merangkum simbol yang digunakan dalam definisi T-k-PNN.

Tabel 1. Simbul dan diskripsi metode T-k-PNN

Symbol	Maksud
D	Database Tidak Pasti (Uncertain Database)
O _i	Objek kei tidak pasti dari D (i=1,..., D)
r _i	O _i -q
d _i (r)	Pdf dari r _i (jarak pdf-probability density function)
D _i (r)	Cdf of r _i (jarak cdf-continous density function)
q	Query point
T	Probability Threshold
S	{o _i o _i ∈ D}
P(S)	Probabilitas Kualifikasi dari S (Qualification prob. of S)

Kami gunakan simbol dan diskripsi diatas sebagai solusi sederhana untuk menjawab T-k-PNN, yang membentuk dasar untuk pemecahan masalah dan usulan metode lebih lanjut. Metode ini memanfaatkan distribusi probabilitas setiap jarak objek dari q. Secara formal, bahwa $r_i \in \mathfrak{R}$ menjadi jarak absolut dari objek yang tidak pasti dari q. Yaitu, $r_i = |o_i - q|$. Kami berasumsi bahwa r_i mengambil nilai $r \in \mathfrak{R}$. Kemudian, jarak pdf dan cdf dari o_i didiskripsikan sebagai simulai sebagai berikut :

“ Diberikan objek yang tidak pasti, misal jaraknya pdf, dilambangkan dengan di (r), adalah pdf dari Ri, misl jaraknya cdf, dilambangkan oleh Di (r), adalah cdf dari Ri”



Gambar 3. Jarak pdf dan cdf dari obyek tidak pasti o1 (a) dan dan jarak distribusi r (b)

Kami menggunakan contoh untuk menggambarkan dua fungsi ini pada ilustrasi Gambar 3 diatas (gambar 3 kiri) menunjukkan objek tak pasti satu dimensi, o1, yang memiliki seragam nilai pdf $1/u-l$ di wilayah ketidakpastian $[l, u]$. Titik kueri, q_0 , juga ditampilkan. Pada gambar 3 diatas (gambar (a) 3 kanan), jarak pdf (diarsir) dari $r_1 = |o_1 - q_0|$, mulai dari nol hingga $u - q_0$, diilustrasikan. Jarak pdf dalam $[0, q_0 - l]$ adalah jumlah dari pdf di kedua sisi q_0 , yaitu sama dengan $2/u-l$. Dalam $[q_0 - l, u - q_0]$, jarak pdf dari r_1 adalah $1/u-l$. Sedangkan jarak cdf dari r_1 , digambarkan sebagai garis padat, ditemukan dengan mengintegrasikan jarak pdf.

Berdasarkan ketidakpastian pdf dan cdf dari masing-masing objek, probabilitas kualifikasi dari k-subset S (mis., $P(S)$) dapat dihitung probabilitasnya. Probabilitas, $p(S)$, kemudian digunakan untuk membandingkan terhadap T; jika $p(S) \geq T$, maka S menjadi jawaban. Sekarang bisa kita lihat bagaimana $p(S)$ dihitung:

$$p(S) = \sum_{o_i \in S} \int_0^{+\infty} d_i(r) \prod_{o_j \in S \wedge o_j \neq o_i} D_j(r) \prod_{o_h \in D-S} (1 - D_h(r)) dr$$

Dengan pemahaman perhatikan bahwa untuk S menjadi jawaban permintaan, jarak dari objek apa pun o_h (di mana $o_h \in D - S$) dari q harus lebih besar dari o_i (di mana $o_i \in S$). Sekarang, pada jarak r , the pdf objek $o_i \in S$ memiliki jarak terpendek ke-k dari q adalah produk dari faktor-faktor berikut:

- pdf yang memiliki jarak r dari q , yaitu, $d_i(r)$;
- probabilitas bahwa semua objek di S selain o_i memiliki lebih pendek jarak dari r , mis., $\prod_{o_j \in S \wedge o_j \neq o_i} D_j(r)$; dan
- probabilitas bahwa objek dalam $D - S$ memiliki jarak yang lebih jauh dari r , mis., $\prod_{o_h \in D - S} (1 - D_h(r))$

Fungsi integrasi dalam persamaan tersebut pada dasarnya adalah produk dari tiga faktor di atas. Dengan mengintegrasikan fungsi ini ke atas $(0, +\infty)$, kami memperoleh probabilitas bahwa S berisi k tetangga terdekat dengan o_i sebagai tetangga terdekat k-th. Akhirnya, dengan merangkum nilai probabilitas ini untuk semua objek $o_i \in S$, dapat hasil evaluasi persamaan tersebut tidak efisien, dengan beberapa ketidak efisienan tersebut sebagai berikut :

Pertama, jarak pdf dan cdf dari setiap objek harus dihitung.

Kedua, Persamaan 1 melibatkan integrasi numerik yang mahal, dan harus dilakukan dalam rentang yang besar.

Ketiga, probabilitas setiap k-subset S harus dihitung, dan jumlah himpunan-himpunan ini adalah eksponensial.

Namun, kami menemukan bahwa T-k-PNN dapat ditangani dengan cara yang lebih baik, dengan menyajikan metode baru yang memiliki beberapa fungsi secara khusus. Di bagian bawah, kami akan melakukan :

Mengeksploitasi tingkat wilayah ketidakpastian objek dan ambang probabilitas untuk secara signifikan memangkas jumlah sub-k yang akan diperiksa.

Kemudian, di bagian berikutnya, kami memperoleh batas bawah dan atas probabilitas kualifikasi k-subset.

Sehingga keputusan apakah sub-k harus diterima sebagai jawaban kueri dapat dibuat tanpa menghitung probabilitas aktualnya.

4. Hasil Wilayah atau Area Ketidak Pastian dan Ambang Probabilitas (k-bound filtering)

Diberikan titik query (query point), solusi naif adalah untuk menghitung semua kemungkinan kombinasi set ukuran k dari database D dan hitung probabilitasnya sesuai dengan diskripsi masalah. Jika probabilitas lebih besar dari T, himpunan akan dikembalikan sebagai hasilnya. Jelas, ini adalah pendekatan yang tidak efisien sehubungan dengan perhitungan dan biaya I / O. Bahkan, mengingat T-k-PNN, pertama-tama kita dapat memanfaatkan jarak informasi objek untuk memangkas mereka yang tidak memenuhi syarat jawaban. Secara khusus, kami mengusulkan filter yang efisien berdasarkan minimum k-th dari jarak maksimal f_k , disebut k-bound filter, untuk menghapus objek yang memiliki probabilitas nol untuk menjadi jawaban T-k-PNN. Dasar pemikiran di balik filter terikat-k dinyatakan dalam mengikuti diskripsi permintaan sebagai berikut :

Diberikan objek $o_i \in D$, titik kueri q , jika $\min(r_i) > f_k$, maka o_i tidak akan muncul di kumpulan data jawaban (data set) mana pun, di mana f_k adalah kth minimum jarak maksimal (k-bound) di antara semua r_j untuk $(j = 1, \dots, n)$, dan r_i adalah jarak antara q dan o_i . Kinerja penyaringan k-bound dapat dipercepat dengan bantuan struktur indeks spasial. Dalam karya ini kami mengindeks wilayah ketidakpastian dari setiap objek data dalam R-tree, di mana penyaringan k-terikat dapat dilakukan. Alasan kami memilih R-tree adalah karena popularitasnya. Namun struktur indeks spasial lainnya juga dapat digunakan. R-tree secara rekursif mengelompokkan objek data yang tidak pasti dengan persegi panjang pembatas minimum (MBR) hingga satu simpul akhir (root) diperoleh. Proses penyaringan melalui R-tree dirinci dalam usulan algoritma 1 (k-bound_Processing). Algoritma mempertahankan heap minimum H yang berisi entri formulir (v, key) , di mana kuncinya adalah $\min(\text{dist}(q, v))$. H pertama kali diinisialisasi (baris 1). Set objek kandidat dikosongkan dan f_k ditetapkan sebagai tak terhingga (baris 2 dan 3). Kemudian, simpul akar dimuat dan disimpan dalam H (baris 4). Setiap kali kami mengeluarkan entri (v, kunci) dari heap H (baris 6), dan memeriksa apakah kunci lebih kecil dari f_k (baris 7). Jika jawabannya negatif, entri ini dibuang (Lemma 4.1). Jika tidak, kami kemudian memeriksa apakah v adalah simpul daun (baris 8). Jika jawabannya ya, kami menyisipkan objek kandidat yang didirikan ini ke C (baris 9). Jika v adalah simpul perantara, untuk setiap entri v_i dalam v , kami menghitung jarak minimumnya dari q (baris 11 dan 12). Jika jarak minimum ini juga lebih kecil dari f_k , kami memasukkan v_i ke heap H dan memperbarui f_k jika perlu (baris 13 hingga 15). Proses ini berulang sampai antrian kosong.

```

input : R-tree I constructed over D
input : q:(query point)
input : k
output: Kandidat Kumpulan Obyek C dan fk
1.   inisialisasi min-heap H menerima entri dalam bentuk (v, key);
2.   C ← ∅;
3.   fk ← +∞;
4.   insert (root(I),0) into heap H;
5.   while H is not empty do
6.   (v, key) ← de-heap H;
7.   if key < fk then
8.   if v is an uncertain object then
9.   insert v into C;
10.  else
11.  for each entry vi contained in v do
12.  compute key ← min(dist(q, vi));
13.  if key < fk then
14.  insert(vi, key) to H;
15.  update fk according to key;
16.  return C, fk

```

Gambar 4. Rancangan Algoritma metode fungsi k-bound Processing

$\prod_{i \in S} p(r_i \leq f_k)$, yaitu produk dari probabilitas cutoff obyek anggotanya. Karena probabilitas cutoff selalu lebih kecil dari atau sama dengan 1, $UB\ Prob(S')$ memberikan batas atas $\prod_{i \in S} p(r_i \leq f_k)$.

5. Hasil Desain Metode Probabilitas Pemilihan Candidate (Calon)

Pemilihan Calon Probabilistik Setelah pemfilteran k-bound, anggap kita memperoleh objek $m \in [k, n]$, yaitu $C = \{o_1, \dots, o_m\}$, masih mungkin ada C_m kemungkinan jawaban k-subset. Secara langsung menghitung set jawaban ini akan menghasilkan biaya eksponensial dalam memori dan komputasi. Bahkan, tidak perlu untuk menghasilkan semua himpunan k-subsets. Pada bagian ini, kami mengusulkan metode pembuatan set kandidat berdasarkan informasi probabilitas, yaitu pemilihan kandidat probabilistik (Probabilistic Candidate Selection-PCS). Secara khusus, kami menggunakan probabilitas dari objek yang terletak di dalam k-bound, yang disebut probabilitas cutoff (CP) (ditunjukkan pada Gambar 5 (a)) dan fakta bahwa probabilitas kualifikasi subset k-harus lebih kecil dari produk dari probabilitas cutoff anggotanya. Diskripsi persamaannya adalah:

Fakta “S mengandung k terdekat tetangga” mengharuskan semua objek anggota S memiliki jarak dari q tidak lebih besar dari f_k . Itu berarti $p(S)$ tidak boleh lebih besar dari $\prod_{i \in S} p(r_i \leq f_k)$, yaitu produk dari probabilitas cutoff obyek anggotanya. Karena probabilitas cutoff selalu lebih kecil dari atau sama dengan 1, $UB\ Prob(S')$ memberikan batas atas $\prod_{i \in S} p(r_i \leq f_k)$.

```

input : C = {o1, ..., om}, q,k, T
output: Ck: the set of candidate object sets
1.   C1 ← {{o1}, ..., {om}};
2.   for i ← 1 to k - 1 do
3.     Ci+1 ← ∅;
4.     for each S ∈ Ci do
5.       z ← maxoi∈S(i) ;
6.       for j ← z + 1 to c do
7.         if oj ∈/ S then
8.           S' ← S ∪ {oj};
9.           if S' ∈/ Ci+1 then
10.            if UBProb(S') ≥ T then
11.              Ci+1.add(S')
12.            else
13.              break;
14.    return Ck

```

Gambar 5. Rancangan Algoritma metode Probabilistik Candidate Selection

Berdasarkan probabilitas cutoff dari setiap objek kandidat dalam k-terikat, algoritma PCS membangun k-subset dengan menumbuhkan daftar subset i sehubungan dengan i (di mana $i = 1$ hingga $k - 1$). Langkah-langkah PCS tercantum dalam Algoritma 2. Pertama, algoritma menghasilkan 1-subset berdasarkan kandidat set C (baris 1). Kemudian, subset $(i + 1)$ - dihasilkan oleh penggabungan i-subset dan C (baris 2 hingga 13). Nilai rob UBP (S') dapat diperoleh dengan Lemma 4.2 (baris 10). Semua himpunan bagian dengan rob UBP (S') lebih kecil dari ambang akan dipangkas (baris 11). Oleh karena itu, banyak himpunan bagian menengah dipangkas, dan jumlah himpunan bagian-k akan sangat berkurang. Ketika kita memperluas S ke S' dengan menambahkan o_j , dan menemukan bahwa S' harus dipangkas, maka tidak perlu memeriksa ekstensi dengan $o_j + 1, \dots, o_m$ (baris 13), karena objek data diurutkan dalam urutan probabilitas cutoff mereka.

Gambar 5 (a) -5 (c) menunjukkan contoh menghasilkan kandidat k-subset berdasarkan probabilitas cutoff dari masing-masing objek kandidat dalam k-bound. Gambar 5 (a) daftar probabilitas cutoff (CP) dari setiap objek. Kita dapat dengan aman menghapus objek kandidat o_6 karena CP-nya kurang dari ambang $T = 0,2$. Kemudian, di babak kedua, seperti yang ditunjukkan pada Gambar 5 (b), subset $\{o_4, o_5\}$ dapat dihapus. Demikian pula, pada ronde ketiga (Gambar 5 (c)) subset kandidat $\{o_1, o_4, o_5\}$, $\{o_2, o_4, o_5\}$ dan $\{o_3, o_4, o_5\}$ dapat dihapus dengan aman.

Kami mengamati bahwa pengguna mungkin tidak selalu tertarik untuk mendapatkan nilai probabilitas yang tepat. Dia mungkin hanya membutuhkan jawaban dengan keyakinan yang memenuhi beberapa kondisi yang telah ditentukan. Misalnya, pengguna hanya mungkin memerlukan jawaban dengan keyakinan lebih tinggi dari beberapa nilai tetap. Pada Gambar 2, misalnya, jika jawaban dengan probabilitas minimal 20% diperlukan, maka himpunan $\{o_1, o_2, o_3\}$ dan $\{o_1, o_2, o_4\}$ akan menjadi satu-satunya jawaban. Kami menyebut varian k-PNN dengan batasan ambang probabilitas, T (mis., 20%), sebagai Ambang Batas Probabilitas k-Nearest-Neighbor Query (atau singkatnya T-k-PNN). Batasan ambang batas memungkinkan pengguna untuk mengontrol kepercayaan yang diinginkan diperlukan dalam jawaban kueri. Pada Gambar 2, misalnya, 0,2-3-PNN mengembalikan $\{o_1, o_2, o_3\}$ dan $\{o_1, o_2, o_4\}$ sebagai jawaban pertanyaan. Jawaban kueri semacam itu juga memungkinkan pengguna untuk mengekstrak beberapa informasi yang berguna (mis., o_1 dan o_2 muncul di kedua 3-subsets dalam contoh ini). Perhatikan bahwa dengan nilai T yang moderat, jumlah himpunan k-subsets kembali cukup kecil dalam praktik. Misalnya, dalam percobaan kami, pada $T = 0,1$, rata-rata dua k-subset dikembalikan.

6. Hasil Desain Metode Probabilitas Pemilihan Candidate

Selain itu, kami menyajikan tiga metode untuk memproses permintaan T -kPNN secara efisien. Metode pertama, disebut penyaringan k-bound filtering, secara efektif menghapus semua objek yang tidak memiliki kesempatan untuk menjadi jawaban kueri. Mari kita perhatikan Gambar 2 lagi, yang menunjukkan "k-bound" (sebagai lingkaran putus-putus yang berpusat pada q) yang benar-benar menutup tiga objek o1, o2 dan o3. Jari-jari dari 3-bound didefinisikan sebagai minimum ketiga dari jarak maksimal objek dari q (dalam contoh ini, jarak maksimum o3 dari q). Dengan k-bound, objek o7 dan o8 dapat segera dipangkas, karena mereka tidak memiliki kesempatan untuk menyalip salah satu objek o1, o2 atau o3 untuk menjadi bagian dari jawaban untuk kueri 3-PNN. Secara umum, dengan k-bound, banyak objek dapat dihapus, dan seperti yang kami tunjukkan dalam experiment penelitian ini, penggunaannya dapat dengan mudah ditingkatkan ke indeks spasial (mis., R-tree). Untuk kenyamanan, kami menyebut objek yang tidak dipangkas oleh pemfilteran k-bound (mis., Yang tumpang tindih dengan k-terikat) objek kandidat.

Setelah pemfilteran k-bound, kita masih perlu mempertimbangkan k-subsets bagian dari objek kandidat. Pada Gambar 2, misalnya, $C36 = 20$ set kardinalitas 3 mungkin perlu dipertimbangkan. Untuk lebih mengurangi ruang pencarian, kami mengusulkan metode kedua, yaitu Probabilistic Candidate Selection (atau PCS), yang dapat secara efisien mendeteksi k-subset (yaitu, himpunan bagian dari database D dengan kardinalitas k) yang probabilitas kualifikasi kurang dari T, juga disebut k-subset wajar tanpa pengecualian. Sementara pemfilteran k-bound menggunakan informasi jarak untuk pemangkas, PCS menggunakan informasi probabilitas dari data yang tidak pasti untuk menghapus sub-k yang tidak memenuhi syarat. Alasan di balik PCS adalah bahwa mengingat probabilitas objek kandidat yang terletak di dalam k-bound (disebut probabilitas cutoff), probabilitas kualifikasi k-subsets harus lebih rendah daripada produk probabilitas cutoff subsetnya. Pada Gambar 2, misalnya, probabilitas cutoff dari o4, o5, dan o6 masing-masing adalah 0,5, 0,2 dan 0,1 (ditampilkan sebagai area berarsir). Probabilitas kualifikasi 3-subset {o2, o4, o5} harus lebih rendah dari produk dari probabilitas cutoff {o4, o5}, atau $0,5 \times 0,2 = 0,1$. Jika $T = 0,2$, maka {o2, o4, o5} dapat dipangkas. Berdasarkan fakta yang bermanfaat ini, algoritma PCS membangun k-subsets bagian dengan menumbuhkan daftar i-subsets sehubungan dengan i (di mana $i = 1, 2, \dots, k$). Pada setiap iterasi i, produk dari probabilitas cutoff dari setiap subset i diperiksa apakah itu lebih kecil dari T, dan subset i dipangkas jika itu benar. Selain itu, kami juga merancang teknik, yang disebut "pemangkas seeds", untuk lebih meningkatkan kinerja PCS dengan menghapus himpunan bagian yang tidak memenuhi syarat dengan "seeds" - objek yang terletak di dalam k-bound. Selain itu, metode kompresi data yang efisien yang menekan jumlah overhead penyimpanan menengah yang diperlukan oleh PCS disajikan. Eksperimen kami menunjukkan bahwa PCS mengurangi porsi k-subsets yang signifikan untuk diperiksa.

1-subset	CP
{o1}	1
{o2}	1
{o3}	1
{o4}	0,5
{o5}	0,2
{o6}	0,1

a. Round 1 (cadidat o6 di catoff, karena $CP < T$)

2-subset	CP
{o1,o2}	1
{o1,o3}	1
{o1,o4}	0,5

{o1,o5}	0,2
{o2,o3}	1
{o2,o4}	0,5
{o2,o5}	0,2
{o3,o4}	0,5
{o3,o5}	0,2
{o4,o5}	0,1

b. Round 2 (cadidat o4,o5 di catoff, karena $CP < T$)

2-subset	CP
{o1,o2,o3}	1
{o1,o2,o4}	0,5
{o1,o2,o5}	0,2
{o1,o3,o4}	0,5
{o1,o3,o5}	0,2
{o1,o4,o5}	0,1
{o2,o3,o4}	0,5
{o2,o3,o5}	0,2
{o2,o4,o5}	0,1
{o3,o4,o5}	0,1

c. Round 3 (cadidat o4,o5 di catoff, karena $CP < T$)

Gambar 6. Ilustrasi CP Subset Canidat

Size-1 set	CP
{o1}	1
{o2}	1
{o3}	1
{o4}	0,5
{o5}	0,2

a. Round 1 (Size-1 set)

Size-1 set	CP
{o1,o5}	1
{o2,o5}	1
{o3,o5}	0,5

b. Round 2 (Size-2 set)

Size-3 set	CP
{o1,o2,o5}	1
{o1,o3,o5}	1
{o2,o3,o5}	1

c. Round 3 (Size-3 set)

Gambar 7. Ilustrasi Compressed Subset Canidat CP

```

input : seeds, S and  $f_1, \dots, f_k$ .
output: A boolean value indicating whether S is a possible
result.
1.   for each  $o_j \in S$  do
2.   if  $\min(r_j) < f_1$  then
3.    $\gamma \leftarrow 0$ ;
4.   else
5.    $\gamma \leftarrow$  the largest  $i$  satisfying  $\min(r_j) \geq f_i$ ;
6.   if  $\gamma > 0$  then
7.   if not {seeds[1], ..., seeds[ $\gamma$ ]}  $\subseteq S$  then
8.   return False
9.   return True

```

Gambar 8. Rancangan Algoritma Metode Cutoff-Seed

Namun, kami menemukan bahwa T-k-PNN dapat ditangani dengan cara yang lebih baik. Secara khusus, nanti, di Bagian 4, kami mengeksploitasi tingkat wilayah ketidakpastian objek dan ambang probabilitas untuk secara signifikan memangkas jumlah sub-k yang akan diperiksa. Kemudian, di Bagian 5, kami memperoleh batas bawah dan atas probabilitas kualifikasi k-subset, sehingga keputusan apakah sub-k harus diterima sebagai jawaban kueri dapat dibuat tanpa menghitung probabilitas aktualnya.

Kami mengamati bahwa pengguna mungkin tidak selalu tertarik untuk mendapatkan nilai probabilitas yang tepat. Dia mungkin hanya membutuhkan jawaban dengan keyakinan yang memenuhi beberapa kondisi yang telah ditentukan. Misalnya, pengguna mungkin hanya memerlukan jawaban dengan keyakinan lebih tinggi daripada beberapa yang diperbaiki nilai. Pada Gambar 2, misalnya, jika jawaban dengan probabilitas minimal 20% diperlukan, maka set {o1, o2, o3} dan {o1, o2, o4} akan menjadi satu-satunya jawaban. Kami menyebut varian k-PNN dengan batasan ambang probabilitas, T (mis., 20%), sebagai Ambang Batas Probabilitas k-Quest-Neighbor Query (atau singkatnya T-k-PNN). Ambang batas kendala memungkinkan pengguna untuk mengontrol kepercayaan yang diinginkan diperlukan dalam jawaban kueri. Pada Gambar 2, misalnya, 0,2-3-PNN kembali {o1, o2, o3} dan {o1, o2, o4} sebagai jawaban pertanyaan. Jawaban kueri semacam itu juga memungkinkan pengguna untuk mengekstrak beberapa informasi yang bermanfaat (mis., O1 dan o2 muncul di 3-himpunan bagian dalam contoh ini). Perhatikan itu dengan nilai T yang moderat, jumlah k-himpunan bagian yang dikembalikan cukup kecil dalam praktik. Misalnya, dalam percobaan kami, pada $T = 0,1$, rata-rata dua k-subset dikembalikan.

7. Hasil Desain Konsep Metode Verifikasi dan Perbaikan.

Basarkan k-himpunan bagian yang dihasilkan oleh PCS, kami sekarang menyajikan efisiensi teknik untuk menangani himpunan bagian-k. Kami membahas teknik untuk menurunkan batas atas dan bawah probabilitas kualifikasi k-himpunan bagian dalam Bagian 5.1. Kami kemudian mempelajari bagaimana teknik ini memfasilitasi perhitungan probabilitas.

Proses Verifikasi, Kami sekarang mensimulasikan bagaimana partisi dapat digunakan untuk memperoleh batas bawah dan atas dari masing-masing probabilitas kualifikasi k-subset secara efisien (mis., $P(S)$). Biarkan $[p(S) .l, p(S) .u]$ menjadi batas bawah dan atas dari $p(S)$. Misalkan X adalah struktur data yang menyimpan informasi partisi, dan Q adalah himpunan bagian-k yang dihasilkan dari algoritma PCS. Dengan input ini, algoritme verifikasi (Algoritma 4 menilai apakah k-subset S harus dianggap sebagai jawaban kueri. Ini menghasilkan set jawaban A (yaitu, k-subset yang memenuhi kueri) dan penyempurnaan set U (yaitu, k- himpunan bagian yang perlu diselidiki lebih lanjut). Desain langkah-langkah tersebut adalah :

Langkah 1 menginisialisasi dua set, A dan U , untuk mengosongkan set.

Langkah 2 Mengulaing langkah 1 Untuk setiap k-subset $S \in Q$,

Langkah 3 Langkah 3 menggunakan subrutin UB untuk menemukan batas atas probabilitas kualifikasi S (mis., $p(S) .u$). Jika $p(S) .u$ lebih kecil dari T , maka S tidak dapat memenuhi permintaan dan dipangkas.

Angkah 4 dan 5 Selanjutnya, subrutin LB dipanggil untuk menemukan $p(S) .l$.

Langkah 6 Jika nilai ini tidak kurang dari T , lalu S dimasukkan ke set jawaban.

Langkaah 7 Jika tidak, S dimasukkan ke dalam set U untuk diproses lebih lanjut.

Langkah 8 mengembalikan set A dan U .

Ini karena istilah $\prod_{i \in S} D_i (e_j + 1)$ adalah probabilitas itu semua objek di S memiliki jarak dari q tidak lebih besar dari titik akhir $e_j + 1$. Dengan mengurangi $\prod_{i \in S} D_i (e_j)$ darinya, kita memperoleh probabilitas bahwa setidaknya satu objek terletak di dalam $P_j = [e_j, e_j + 1]$. Ini juga kemungkinan jarak maksimum semua objek di S (mis., $m(S)$) berada dalam P_j , seperti yang ditunjukkan pada Persamaan 4. Berikut ini menjelaskan rumus untuk $p_j(S) .l$ dan $p_j(S) .u$. Mengingat bahwa $m(S) \in P_j$, semakin rendah dan atas batas probabilitas kualifikasi k-subset, S , adalah:

```

input : Partition info. X, set Q of k-subsets
output: set A of answers, set U of k-subsets to be refined
1.   A ← ∅; U ← ∅;
2.   for each S ∈ Q do
3.     if UB(X, S) ≥ T then
4.     if LB (X, S) ≥ T then
5.       insert S into A;
6.     else
7.       insert S into U;
8. return A,

```

Gambar 9. Algoritma Metode Verifikasi

8. Hasil Ujicoba (Experiment) Sementara

Eksperimen 1: Dalam makalah ini kami membandingkan Algoritma yang penulis kembangkan yaitu Partition_PANN dengan algoritma PTKANN, dan Algoritma PANN, algoritma ini memanfaatkan ide utama (Papadias D, et al, 2005). Memanggil algoritme berulang kali dalam (Papadias D. et al, 2005) untuk k kali untuk mengeksekusi query PTKANN. Dalam percobaan ini, biaya waktu yang diukur adalah biaya rata-rata 100 kueri dengan dataset yang sama tetapi dengan titik kueri yang berbeda dihasilkan secara acak. Hasil kami dirangkum dalam Tabel 1.

Tabel 1 menunjukkan bahwa waktu pemrosesan kueri Algoritma Partition_PANN kurang dari dua pendekatan lainnya. Sangat mudah untuk melihat bahwa waktu pemrosesan query dari Algoritma Partition_PANN tumbuh sangat lambat, terutama ketika $n < 4.000$ yang tidak seperti dua pendekatan lainnya meningkat tajam seperti n , ukuran dataset, meningkat dari Tabel 1. Hasil percobaan menunjukkan bahwa Algoritma Partition_PANN untuk kueri 1NN mengungguli metode sebelumnya dalam waktu proses. Keunggulan ini menonjol, terutama pada dataset besar. Keunggulan ini berasal

dari bahwa struktur data yang sangat baik VR-tree digunakan yang membuat peran diagram Voronoi dibawa ke dalam waktu proses penuh dalam permintaan INN.

Table 1.

Execution time of Partition_PANN versus PtkANN and PANN for different data size

Time(sec)	1.000	2.000	3.000	4.000	5.000	6.000
PtkANN	0,0091	0,0109	0,0201	0,0211	0,0262	0,0298
PANN	0,0062	0,0081	0,0090	0,0103	0,0156	0,0177
Partition_PANN	0,0035	0,0039	0,0043	0,0080	0,0088	0,0098

Eksperimen 2: Untuk setiap nilai k, kami melakukan 100 kNN kueri dengan titik kueri yang berbeda tetapi pada dataset yang sama yang dipilih secara acak yang terdiri dari 6.000 objek, menggunakan Partition_PANN, PANN dan PtkANN, masing-masing dengan k bervariasi dari 2 hingga 200. Kami menyajikan waktu eksekusi rata-rata dari 100 run dari kNN query untuk setiap nilai k dan untuk setiap algoritma, seperti yang ditunjukkan pada Tabel 2.

Dari Tabel 2, mudah untuk melihat bahwa waktu PtkANN and PANN adalah antara 2,01 dan 2,2 kali sebanyak Algoritma Partition_PANN. Kita dapat melihat bahwa waktu eksekusi Algoritma Partition_PANN tumbuh sangat lambat ketika k bertambah ketika k <20.

Table 2.

Execution time of Partition_PANN versus PtkANN and PANN for different k point

Time(sec)	4 k Point	10 k Point	20 k Point	40 k Point	80 k Point	100 k Point
PtkANN	0,0097	0,0111	0,0151	0,0201	0,0232	0,0268
PANN	0,0086	0,0104	0,0133	0,0160	0,0206	0,0254
Partition_PANN	0,0040	0,0051	0,0063	0,0103	0,0167	0,0191

Dari eksperimen terlihat bahwa kinerja Algoritma Partition_PANN akan lebih baik dan lebih sederhana karena k meningkat ketika k > 40. Namun demikian performa tersebut jauh lebih cepat daripada dua metode lainnya. Fenomena ini muncul karena fakta bahwa ukuran kandidat yang ditetapkan $AG_i(p) \cup, \dots, UAG_i(p)$ yang digunakan untuk mencari (i + 1) - tetangga terdekat tumbuh dengan cepat untuk nilai i yang lebih besar, di mana p adalah tetangga terdekat dari titik permintaan. Hasil percobaan menunjukkan bahwa Algoritma Partition_PANN mengungguli metode sebelumnya dalam waktu proses, terutama untuk nilai yang lebih kecil dari k sehubungan dengan ukuran dataset.

Dari Tabel 3, Pengaruh pruning pada f=sum bahwa waktu eksekusi dari Algoritma Partition_PANN selalu lebih baik dua pendekatan lainnya. Kita juga dapat melihat bahwa perbedaan waktu eksekusi Partition_PANN, PtkANN and PANN menjadi semakin lebar seiring dengan meningkatnya ukuran dataset (dari 4 k samapi 1.024 k). Hasil percobaan menunjukkan bahwa Algoritma Partition_PANN lebih baik performanya dari metode sebelumnya dalam waktu eksekusi.

Table 3.

CPU cost by fixing A_Q to 6 % of the data space and the number k of retrieved ANNs (f=sum)

Time(sec)	4 k	16 k	64 k	256 k	1.024 k
PtkANN	0,0139	0,0411	0,0610	0,0921	0,1210
PANN	0,0096	0,0181	0,0195	0,0600	0,0969
Partition_PANN	0,0044	0,0049	0,0054	0,0080	0,0108

Eksperimen 3: Kami membandingkan kinerja Partition_PANN dengan dua metoda lainnya dapat diterapkan pada kasus bahwa lintasan pruning $f=\text{sum}$ adalah segmen garis. Karenanya lintasan pruning yang dipertimbangkan dibatasi untuk segmen garis dalam percobaan kami. Dalam percobaan ini, menggunakan rata-rata 6% untuk setiap area query (A_Q), panjang segmen garis kueri, kami melakukan 40 kueri dengan segmen garis kueri yang berbeda pada dataset yang sama yang dipilih secara acak yang terdiri dari 100 node.

Dari Tabel 4, Pengaruh pruning pada $f=\text{max}$ bahwa waktu eksekusi dari Algoritma Partition_PANN selalu lebih baik dua pendekatan lainnya. Kita juga dapat melihat bahwa perbedaan waktu eksekusi Partition_PANN, PTKANN and PANN menjadi semakin lebar seiring dengan meningkatnya ukuran dataset (dari 4 k samapi 1.024 k). Hasil percobaan menunjukkan bahwa Algoritma Partition_PANN lebih baik performanya dari metode sebelumnya dalam waktu eksekusi.

Table 4.

CPU cost by fixing A_Q to 6 % of the data space
and the number k of retrieved ANNs ($f=\text{max}$)

Time(sec)	4 k	16 k	64 k	256 k	1.024 k
PTkANN	0,0189	0,0461	0,0677	0,0952	0,1265
PANN	0,0109	0,0192	0,0231	0,0620	0,0981
Partition_PANN	0,0051	0,0068	0,0075	0,0080	0,0113

Eksperimen 4: Kami membandingkan kinerja Partition_PANN dengan dua metoda lainnya dapat diterapkan pada kasus bahwa lintasan pruning $f=\text{max}$ adalah segmen garis. Karenanya lintasan pruning yang dipertimbangkan dibatasi untuk segmen garis dalam percobaan kami. Dalam percobaan ini, menggunakan rata-rata 6% untuk setiap area query (A_Q), panjang segmen garis kueri, kami melakukan 40 kueri dengan segmen garis kueri yang berbeda pada dataset yang sama yang dipilih secara acak yang terdiri dari 100 node.

Dari Tabel 4, Pengaruh pruning pada $f=\text{main}$ bahwa waktu eksekusi dari Algoritma Partition_PANN selalu lebih baik dua pendekatan lainnya. Kita juga dapat melihat bahwa perbedaan waktu eksekusi Partition_PANN, PTKANN and PANN menjadi semakin lebar seiring dengan meningkatnya ukuran dataset (dari 4 k samapi 1.024 k). Hasil percobaan menunjukkan bahwa Algoritma Partition_PANN lebih baik performanya dari metode sebelumnya dalam waktu eksekusi

Table 5.

CPU cost by fixing AQ to 6 % of the data space
and the number k of retrieved ANNs ($f=\text{min}$)

Time(sec)	4 k	16 k	64 k	256 k	1.024 k
PTkANN	0,0119	0,0128	0,0151	0,0192	0,0215
PANN	0,0088	0,0101	0,0135	0,0168	0,0196
Partition_PANN	0,0038	0,0044	0,0051	0,0078	0,0101

Eksperimen 4: Kami membandingkan kinerja Partition_PANN dengan dua metoda lainnya dapat diterapkan pada kasus bahwa lintasan pruning $f=\text{max}$ adalah segmen garis. Karenanya lintasan pruning yang dipertimbangkan dibatasi untuk segmen garis dalam percobaan kami. Dalam percobaan ini, menggunakan rata-rata 6% untuk setiap area query (A_Q), panjang segmen garis kueri, kami melakukan 40 kueri dengan segmen garis kueri yang berbeda pada dataset yang sama yang dipilih secara acak yang terdiri dari 100 node.

9. Hasil Luaran Wajib dan Luaran Tambahan

Hasil luaran ini merupakan bagian yang takterpisahkan dari pengembangan metode pencarian berdasarkan identifikasi kesamaan data dalam konsep algoritma K-NN (*k*-nearest neighbours). Algoritma *k*-nearest neighbor adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Data pembelajaran diproyeksikan ke ruang berdimensi banyak, dimana masing-masing dimensi merepresentasikan fitur dari data. Hasil luaran wajib dan tambahan yang dihasilkan pada tahun pertama (ke-2) masih mengacu pada obyek statis, pada tahun ke 2 akan dikembangkan model pencarian pada data dinamis dan bersifat probabilistik. Adapun hasil luaran yang sudah tercapa tersebut adalah sebagai berikut :

- a. Luaran Wajib. Luaran ini berupa Jurnal internasional yang sudah disubmit pada jurnal "Journal of Computer Science" pada tanggal 11 September 2020 (<https://thescipub.com/es/index.php>). Judul pada jurnal tersebut adalah "Voronoi Partition to Support Data Search in Uncertain Database with *k*-Bound Filtering". Luaran ini menitik beratkan optimasi dengan mengembangkan algoritma *k*-NN menjadi PNN untuk pencarian pada data sebagai obyek yang tersebar (data terdistribusi) .
- b. Luaran tambahan. Buku ajar tentang pencarian data dengan konsep K-NN menggunakan pendekatan Voronoi Diagram. Buku tersebut masih dalam tahap editing. Luaran tambahan yang lain adalah hak cipta dari buku ajar tersebut. Sehingga untuk luaran hak cipta masih berupa draf sambil menunggu buku ajarnya terbit.

D. STATUS LUARAN: Tuliskan jenis, identitas dan status ketercapaian setiap luaran wajib dan luaran tambahan (jika ada) yang dijanjikan. Jenis luaran dapat berupa publikasi, perolehan kekayaan intelektual, hasil pengujian atau luaran lainnya yang telah dijanjikan pada proposal. Uraian status luaran harus didukung dengan bukti kemajuan ketercapaian luaran sesuai dengan luaran yang dijanjikan. Lengkapi isian jenis luaran yang dijanjikan serta unggah bukti dokumen ketercapaian luaran wajib dan luaran tambahan melalui Simlitabmas.

Ada dua kelompok luaran yang dihasilkan dari penelitian tahun kedua (Ke-2) ini, yaitu berupa luaran wajib dan luaran tambahan. Status luaran wajib dan luaran tambahan yang dijanjikan tersebut berupa :

- a. Luaran wajib. Luaran ini berupa Jurnal internasional yang sudah terbita pada jurnal Journal of Computer Science (<https://thescipub.com/jcs>) dengan nomor ISSN: 1549-3636 (Print) dan ISSN: 1552-6607 (Online). Judul pada jurnal tersebut adalah " *Voronoi Partition to Support Data Search in Uncertain Database with k-Bound Filtering*". Luaran ini menitik beratkan optimasi algoritma *k*-NN untuk pencarian pada data sebagai obyek yang tersebar (data terdistribusi). Untuk melakukan optimasi pada pencarian ini dilakukan dengan memadukan algoritma K-NN dengan melakukan perpaduan partisi voronoi dan *k*-bound filtering. Tujuan dari model pencarian ini adalah untuk mendukung pengembangan atau memodifikasi algoritma yang sudah ada agar dapat digunakan untuk membantu pencarian obyek yang tersebar (terdistribusi) contohnya adalah data kesehatan DBD pada area atau wilayah yang tersebar (terdistribusi) secara geografis yang berbeda.
- b. Luaran tambahan. Buku ajar tentang pencarian data dengan konsep K-NN menggunakan pendekatan Voronoi Diagram. Buku tersebut masih dalam tahap editing. Luaran tambahan yang lain adalah hak cipta dari buku ajar tersebut. Sehingga untuk luaran hak cipta masih berupa draf sambil menunggu buku ajarnya terbit.

E. PERAN MITRA: Tuliskan realisasi kerjasama dan kontribusi Mitra baik *in-kind* maupun *in-cash* (untuk Penelitian Terapan, Penelitian Pengembangan, PTUPT, PPUPT serta KRUPPT). Bukti pendukung realisasi kerjasama dan realisasi kontribusi mitra dilaporkan sesuai dengan kondisi yang sebenarnya. Bukti dokumen realisasi kerjasama dengan Mitra unggah melalui Simlitabmas.

F. KENDALA PELAKSANAAN PENELITIAN: Tuliskan kesulitan atau hambatan yang dihadapi selama melakukan penelitian dan mencapai luaran yang dijanjikan, termasuk penjelasan jika pelaksanaan penelitian dan luaran penelitian tidak sesuai dengan yang direncanakan atau dijanjikan.

Selama pelaksanaan penelitian tahun ke dua (tahun ke -2) hambatan yang sering kami temukan adalah berkaitan dengan administrasi penelitian, pengurusan hakiserta teknik penulisan jurnal internasional. Hal tersebut karena harus merubah dan menyusun jadwal pertemuan dengan beberapa pihak secara langsung tetapi harus menyesuaikan dengan online (daring) karena masa pandemic. Selain itu hanya ada beberapa hambatan teknis yang sebenarnya relatif lebih mudah dalam menyelesaikan, tetapi sulit untuk diadministrasikan. Secara garis besar hambatan atau kesulitan tersebut dapat kami jelaskan sebagai berikut :

a. Hambatan Administrasi Penelitian. Karena dalam penelitian ini kami dituntut harus tertib administrasi sebagai bentuk pertanggung jawaban dan akuntabilitas pelaksanaan penelitian, maka waktu kami sangat disibukan dengan berbagai hal yang berkaitan dengan urusan administrasi atau pelaporan, sehingga waktu kami banyak yang tersita untuk pembuatan laporan yang harus sesuai dengan tuntutan pelaporan penelitian. Sehingga kami berharap ada bentuk atau model pelaporan yang lebih simpel dan praktis sehingga waktu kami sebagai peneliti baik ketua ataupun anggota dapat lebih fokus pada substansi penelitian. Model pembiayaan karena dana hibah belum cair atau turun sampai dengan bulan September 2020 maka mencari solusi pembiayaan mandiri untuk dilakukan ijin penggantian reimbursement. Dengan demikian pelaksanaan didahulukan pembiayaan akan dibuat penggantian pembiayaan dilakukan beberapa waktu kemudian. Kemudian juga ada komponen pembiayaan baru dalam rangka kegiatan video conference (Vicon) dan kegiatan lain yang dilakukan secara daring.

b. Hambatan Jurnal internasional. Dalam penulisan jurnal internasional (sebagai hasil luaran) terutama jurnal yang bereputasi kebanyakan memiliki teknik dan standar penulisan yang "rigid" sehingga proses review dan refisi memakan waktu yang relatif lama, karena jurnal yang bereputasi biasanya penerbitanya hanya 2 kali dalam satu tahun bahkan ada yang satu kali dalam satu tahun. Sebagian besar jurnal yang bereputasi melakukan review dalam bentuk blind review sehingga sulit di konfirmasi sejauh mana proses dan waktu yang dibutuhkan untuk melakukan review. Sehingga perlu diasiasi atau diantisipasi karena hal ini yang membutuhkan waktu proses yang lama. Kami menyiasati dengan menulis jurnal diawal waktu penelitian sehingga punya waktu yang agak lama untuk melakukan penulisan, submit , review dan refisi. Hal ini penulis lakukan dengan pertimbangan keterbatasan waktu (jika normal sesuai time line penelitian dilakukan menjelang akhir penelitian), tetapi dari segi logika penelitian seperti terjadi anomali karena mengedepankan publikasi dibandingkan proses riset itu sendiri. Dan yang perlu menjadi perhatian adalah biaya penerbitan jurnal bereputasi yang relatif sangat mahal untuk ukuran dosen pada umumnya maupun peneliti pada khususnya.

c. Hambatan Teknis. Beberapa kegiatan dilakukan secara daring sehingga membutuhkan pelaksanaan teknis tambahan bagi para pendukung penelitian seperti analis, programmer dan penguji model. Bagi peneliti dibidang komputer dan informatika yang menghasilkan model software, komponen terbesar biaya penelitian bukan bahan habis pakai atau bahan penelitian yang sifatnya fisik atau material. Semua uji coba penelitian sifatnya adalah konsep rekayasa dan algoritma. Sehingga membutuhkan tenaga atau profesional untuk penunjang teknis penelitian baik itu analis sistem, programer atau pun database desain yang memiliki cukup waktu yang lama untuk menghasilkan rekayasa dan program algoritma dalam bentuk software. Secara waktu (ketersediaan waktu) pekerjaan teknis tersebut tidak mungkin dapat di handle oleh ketua peneliti maupu anggota, sehingga dibutuhkan honor untuk para penunjang teknis tersebut, sedangkan komponen honor pada saat ini ditiadakan oleh aturan penelitian terbaru yang mengacu pada peraturan menteri keuangan.

G. RENCANA TAHAPAN SELANJUTNYA: Tuliskan dan uraikan rencana penelitian di tahun berikutnya berdasarkan indikator luaran yang telah dicapai, rencana realisasi luaran wajib yang dijanjikan dan tambahan (jika ada) di tahun berikutnya serta *roadmap* penelitian keseluruhan. Pada bagian ini

diperbolehkan untuk melengkapi penjelasan dari setiap tahapan dalam metoda yang akan direncanakan termasuk jadwal berkaitan dengan strategi untuk mencapai luaran seperti yang telah dijanjikan dalam proposal. Jika diperlukan, penjelasan dapat juga dilengkapi dengan gambar, tabel, diagram, serta pustaka yang relevan. Jika laporan kemajuan merupakan laporan pelaksanaan tahun terakhir, pada bagian ini dapat dituliskan rencana penyelesaian target yang belum tercapai.

Rencana dan tahapan penelitian tahun ke-2 merupakan tindak lanjut dari hasil yang sudah dicapai dari penelitian tahun ke-1 sebelumnya (2019). Penjelasan dibawah ini menggambarkan tahapan penelitian yang sudah dikerjakan dan belum dikerjakan (sebagai rencana tahapan berikutnya) dari kegiatan penelitian tahun ke-2.

Tahapan-tahapan sebagai bagian metode Penelitian Dasar Unggulan Perguruan Tinggi (PDUPT) ini akan menjelaskan tentang bagaimana pendekatan atau metode dalam pengembangan algoritma pencarian untuk data spasial menjadi sistem pencarian yang efektif dengan instrument perbandingan adalah waktu query, penggunaan I/O device dan biaya komunikasi.

Dalam mempersiapkan pengembangan algoritma pencarian pada data spasial terutama pada model pencarian berbasis lokasi, peneliti menggunakan metode pengembangan software System Development Life Cycle (SDLC), metode ini sangat relevan dan cocok untuk mengembangka system software yang memiliki proses dasar dengan fungsionalitas dan alur data yang jelas seperti PTNN. Pada tahapan metoda ini juga memudahkan dalam merancang secara konseptual menuju perancangan detail, ada beberapa tahap dalam metode ini, yaitu:

1. Tahap Perencanaan, dilakukan oleh ketua dan anggota peneliti
2. Tahap Analisis, dilakukan oleh anggota peneliti dan dan tim penunjang teknis pada Laboratorium RPL dan Data
3. Tahap Perancangan (design) algoritma, dilakukan oleh ketua dan anggota peneliti serta tim teknis serta tim support.
4. Tahap Pembuatan model algoritma pencarian, dilakukan oleh tim teknis, tim support dibawah pengawasan anggota peneliti.
5. Tahap Implementasi, dilakukan oleh tim teknis dibawah pengawasan ketua dan anggota peneliti
6. Tahap Evaluasi dan Monitoring, dilakukan oleh tim pendukung, ketua dan anggota peneliti.

Dari 6 tahapan tersebut yang sudah diselesaikan pada pelaksanaan penelitian tahun ke-2 adalah tahap 1, tahap 2, tahap 3, tahap 4.

Sedangkan pada tahapan khusus dalam melakukan desain dan pengembangan algoritma pada tahapan 5 dan 6 tersebut akan dikerjakan dan diselesaikan pada sisa waktu kegiatan penelitian tahun ke-2 (2020). Tahap 5 dan 6 yang merupakan detail desain tersebut akan di uraikan dan di realisasikan melalui sub tahapan sebagai berikut :

Sub Tahap Pemrosesan

Dalam sub tahapan ini kami menggunakan metode inkremental untuk menghitung kumpulan data query tertutup minimum dan mengusulkan algoritma Min_Circle (Q). Pertama hitung lingkaran tertutup minimum dari titik permintaan $i-1$ pertama, dan tambahkan titik i -th. Jika titik berada di atau di lingkaran, maka kami tidak memprosesnya. Kalau tidak, titik yang memiliki jarak terpanjang dari pusat lingkaran tertutup minimum saat ini diakses dan memperoleh lingkaran tertutup minimum baru. Ulangi prosedur yang disebutkan di atas dan lintasi titik-titik query secara berurutan, sampai semua titik query berada di dalam atau di luar lingkaran. Dan lingkaran ini adalah lingkaran tertutup minimum yang kami hitung.

Sub Tahap Filtering Aggregate Function $f=\text{sum}$

Memproses kumpulan data query oleh algoritma Min_Circle dan mendapatkan titik query tunggal q . Jika q ada di SUV (p_i), tambahkan titik generasi tunggal dan k level titik penghasil yang berdekatan ke dalam antrean Can_visit. Jika q dalam MUV (p_i, p_j), tambahkan beberapa titik generasi dan k level titik pembangkitan yang berdekatan ke dalam antrean Can_visit. Untuk setiap titik data C_i dalam Can_visit, tentukan apakah jumlah titik pembangkitan antara C_i dan q kurang dari k atau tidak.

Penyimpanan $(C_i, \text{distmin}(q, C_i))$ ke H dan disortir $\text{distmin}(q, C_i)$ dalam urutan menaik. Hitung minimal k -th dari distk-max maksimal. Jika H tidak kosong, ambil $\text{distmin}(q, C_i)$ secara bergantian dan bandingkan dengan distk-max . Jika $\text{distmin}(q, C_i)$ kurang dari distk-max , tambahkan ke dalam set kandidat. Kemudian, teruskan mengambil $\text{distmin}(q, C_i)$ dan bandingkan. Jika $\text{distmin}(q, C_i)$ lebih besar dari distk-max , pangkas C_i dan semua titik data di belakangnya. Algoritma Sum_Can berakhir dengan Sum_Candidate sebagai set kandidat

Sub Tahap Filtering Aggregate Function $f=\max$

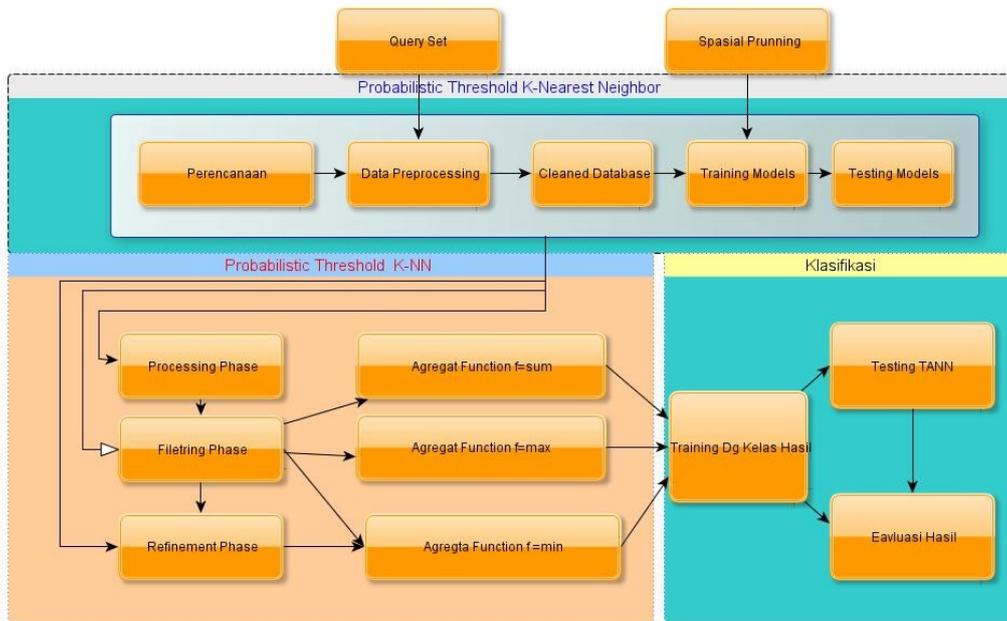
Algoritma menginisialisasi Max_Candidate dan Kunjungan antrian, untuk menyimpan set kandidat dan titik-titik data yang masih harus diakses, masing-masing. Gunakan algoritma Min_Circle untuk memproses kumpulan data query, dan dapatkan titik query tunggal q . Jika q ada di $\text{SUV}(p_i)$, maka tetapkan titik generasi ini sebagai yang terbaik ANN , dan tambahkan titik generasi ini dan titik k level yang berdekatan ke dalam Visit . Jika q berada dalam $\text{MUV}(p_i, p_j)$, maka pilih salah satu dari beberapa titik generasi sebagai yang terbaik ANN , dan tambahkan beberapa titik generasi dan titik-titik k level yang berdekatan ke dalam Visit . Hitung jarak agregat best_ANN dalam kasus $f = \text{jumlah}$, dan tetapkan sebagai best_dist menjadi standar pemrosesan pemangkasan. Untuk setiap titik data p_i dalam Visit , tentukan apakah jumlah titik pembangkitan antara dan q kurang dari k atau tidak, dan hitunglah jarak $\text{dist}(p_i, Q)$ antara p_i dan q . Jika $\text{dist}(p_i, Q)$ lebih besar dari atau sama dengan best_dist , maka p_i dapat dipangkas. Jika $\text{dist}(p_i, Q)$ kurang dari best_dist , tambahkan p_i ke dalam pasangan calon Max_Candidate untuk pengujian lebih lanjut. Algoritma Max_Can berakhir dengan Max_Candidate sebagai set kandidat.

Sub Tahap Filtering Aggregate Function $f=\min$

Algoritma menginisialisasi kandidat mengatur Min_Candidate dan memanggil algoritma 1 untuk mendapatkan pusat lingkaran q dari lingkaran tertutup minimum. Untuk pusat query q , dapatkan lokasi di dalam diagram Voronoi yang tidak pasti. Jika q ada di $\text{SUV}(p_i)$, tambahkan titik generasi tunggal dan k level titik pembangkitan yang berdekatan ke dalam kandidat yang ditetapkan Min_Candidate . Jika q dalam $\text{MUV}(p_i, p_j)$, tambahkan beberapa titik generasi dan k level titik pembangkitan yang berdekatan ke dalam kandidat yang ditetapkan Min_Candidate . Algoritma Min_Can berakhir dengan Min_Candidate sebagai pasangan kandidat.

Sub Tahap Refinement

Dalam tahapan ini, kita memangkas titik-titik data yang tidak bisa menjadi hasil query berdasarkan pada dua fase sebelumnya dan mendapatkan pasangan kandidat. Setelah itu, dalam tahap penyempurnaan, kami menghitung probabilitas kumpulan data titik data k di set kandidat dan membandingkan probabilitas dengan ambang yang ditentukan pengguna T . Akhirnya kami mendapatkan hasil yang ditetapkan. Untuk setiap fungsi agregat, dalam tahap penyempurnaan, pertama-tama kita menginisialisasi hasil yang ditetapkan kANN_Result . Memproses kumpulan data query menurut algoritma 1 dan memperoleh titik pusat q dari dataset query. Kandidat yang menetapkan kANN_Candidate dari kANN diperoleh oleh algoritma yang sesuai dalam fase penyaringan. Untuk titik-titik data di set kandidat, hitung semua himpunan yang tersusun dari titik-titik data k dan hitung probabilitas probabilitas dari mereka yang merupakan hasil yang ditetapkan. Bandingkan probabilitas dengan ambang T yang ditentukan pengguna dan kembalikan hasilnya $\{s, \text{prob}(s)\}$ yang probabilitasnya lebih besar dari ambang.



Gambar 10. Tahapan Penelitian

Tabel 6. Tahapan Penelitian Lanjutan (tahun Ke-2)

No	KEGIATAN	Rencana Tahapan Penelitian Tahun Ke-II (2020)												
		Bulan												
		1	2	3	4	5	6	7	8	9	10	11	12	
1.	Menyusun Tim Desain Model Agregasi Proses Query Probabilistik dan Prose Threshold		■	■	■									
2.	Menyusun Tim Uji Model Proses Agregasi dan Pembentukan Threshold			■	■	■								
3.	Konseptual Sinkronisasi Data Proses Agregasi dan Pemotongan Titik Query Berdasar Threshold			■	■	■	■							
4.	Desain Model Query Probabilistik						■	■	■					
5.	Refisi Model Query Prob dan Integrasi Threshold Data Bergerak						■	■	■	■				
6.	Detail Desain Fungsi Query Probabilistic Agregat NN						■	■	■	■				
7.	Refisi Fungsi Query						■	■	■	■	■			
8.	Desiminasi Hasil, Migrasi Data Bergerak & Implementasi Terbatas TKT Level 3								■	■	■	■	■	■
9.	Pembuatan Buku, Hak Cipta, Publikasi dan Laporan Tahun II								■	■	■	■	■	■

Tabel 2 diatas merupakan berbagai tahapan tindak lanjut dari laporan kemajuan penelitian tahun ke-2 ini disusun sebagai rujukan tahapan pekerjaan akhir tahun terakhir (tahun ke-2), kami membahas terkait fungsi dan voronoi diagram. Di bagian tahapan ini , kami menyajikan semantik formal T-k-PNN, dan kerangka kerja solusi kami. Bagian lain menjelaskan penyaringan dan proses pemilihan kandidat probabilistik. Rincian verifikasi dan penyempurnaan dikembangkan. Kami menyajikan hasil eksperimen di bagian akhir tahapan pada tahun ke 2.

Berdasarkan rujukan hasil dalam penelitian tahun sebelumnya (tahun ke-1) ini maka garis besar rencana kerja tahap selanjutnya (tahun ke 2) yang berkaitan dengan luaran adalah sebagai berikut : Melakukan identifikasi yang terdiri dari berbagai integrasi input data dan fungsi dari berbagai unit pencarian.

Desain konseptual model prototype secara terbatas dilakukan paparan kepada unit fungsi (Fungsi Filtering, Selection, Prunning, Verifikasi dan refinement) dari metode fungsi Probabilistik Threshold K-nearest Neighbour) untuk mendapatkan masukan tentang kelayakan dan penerimaan fungsi inputan model P-T-k-NN) melalui aplikasi data spasial. Masukan yang diharapkan adalah yang berkaitan dengan norma serta fungsi aplikasi, kelengkapan data, kemudahan dan integrasi.

Melakukan desain ujicoba dan fungsi monitoring, yang terdiri dari analisa dan desain kebutuhan data dan fungsi penamingan model sebaran data dalam konsep mencari obyek dalam ruang dua dimensi.

Perbaikan teknis dan berulang (Incremental Refinement) desain data fungsional ungsional, dimana sebagian data akan terintegrasi dalam satu pengolahan dan perbandingan metoda pencarian berbasis K-NN.

Perbaikan desain fungsi pencarian dan sinkronisasi obyek pada ruang 2 dimensi.

Menguji konsep fungsi P-T-K-NN fungsi antar metode lain dengan pengelola pengendalian pencarian obyek tertentu.

Pembuatan laporan kemajuan penelitian tahun ke-2.

H. DAFTAR PUSTAKA: Penyusunan Daftar Pustaka berdasarkan sistem nomor sesuai dengan urutan pengutipan. Hanya pustaka yang disitasi pada laporan kemajuan yang dicantumkan dalam Daftar Pustaka.

1. Amagata Daichi, Sasaki Yuya, Hara Takahiro, et al, (2016) 'Probabilistic nearest query processing on distributed uncertain data, *Distributed and Parallel Databases*', 34(2).
2. Adler, M. and B. Heeringa (2008) 'Search space reductions for nearest-neighbor queries', *Proceedings of the 5th Annual Conference on Theory and Applications of Models of Computation*, April 25-29, Xi'an, China.
3. Beckmann, N., H.P. Kriegel, R. Schneider and B. Seeger (1990) 'The R*-tree: An efficient and robust access method for points and rectangles', *Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data, May 23-25, Atlantic City, New Jersey*.
4. Chazelle, B. and H. Edelsbrunner (1987) 'An improved algorithm for constructing kth-order voronoi diagrams. *IEEE Trans, Comput.*, 36: 1349-1354.
5. Cheema M A, Zhang Wen-Jie, Lin Xue-Min (2012), 'Efficiently processing snapshot and continuous reverse k nearest neighbors queries', *The VLDB Journal*, 2012, 21(5).
6. Chen P, Gu J, Lin X (2013), 'Group nearest neighbor queries over uncertain data in Location Based Services', *International Journal of Hybrid Information Technology*.
7. Cui J, An Z, Guo Y, et al. (2010). 'Efficient nearest neighbor query based on extended B+-tree in high-dimensional space', *Pattern Recognition Letters*.
8. Chen R, Xie X, Yiu M L (2010), 'UV-diagram: A Voronoi diagram for uncertain data', *Proc of IEEE 26th International Conference on Data Engineering, 2010*, 41(3).

9. Elmongui Hicham G, Mokbel Mohamed F, Aref Walid G (2013), 'Continuous aggregate nearest neighbor queries', *GEOINFORMATICA*, 17(1).
10. Fort M, Sellarès J A (2016), 'Efficient multiple bichromatic mutual nearest neighbor query processing', *Journal Information Systems*, (62).
11. Gao Y, Zheng B (2014) 'Continuous obstructed nearest neighbor queries in spatial databases', *Proceedings of the 28th ACM SIGMOD International Conference of Management of Data*. New York. USA.
12. Guttman, A., (1984), 'R-Trees: A dynamic index for geometric data', *Proceedings of the ACM SIGMOD International Conference on Management of Data, June 18-21, USA*.
13. Gu Yu, Liu Guanli, Qi Jianzhong, et al. (2016), 'The moving k diversified nearest neighbor query', *IEEE transactions on knowledge and data engineering*, 28(10).
14. Hristidis, V. And Papakonstantinou, Y. (2004), 'Algorithms And Applications For Answering Ranked Queries Using Ranked Views', *Vldb J*. 13.
15. JH Park, CW Chung, U Kang (2015), 'Reverse nearest neighbor search with a non-spatial aspect, Information Systems', 54 (3).
16. Ji Changqing; Li Zhiyang; Qu Wenyu (2014), 'Scalable nearest neighbor query processing based on Inverted Grid Index', *JOURNAL OF NETWORK AND COMPUTER APPLICATIONS*, 44.
17. Lee, D.T., (1982), 'On k-nearest neighbor Voronoi diagrams in the plane', *IEEE Trans. Comput.*, 31.
18. Lin Yu-Chieh, Yang De-Nian, Shuai Hong-Han (2015), 'Data acquisition for probabilistic nearest neighbor query', *IEEE Transactions on Knowledge and Data Engineering*, 27(2).
19. Lian X, Chen L (2009), 'Efficient processing of probabilistic reverse nearest neighbor queries over uncertain data', *The VLDB Journal*, 18 (3).
20. Li H, Lu H, Huang B (2005), 'Two ellipse-based pruning methods for group nearest neighbor queries', *Proc of the 13 th Annual ACM Int Workshop on Geographic Information System*, New York: AMC, 2005, pp. 192-199.
21. Papadias D, Shen Q, Tao Y (2004), 'Group nearest neighbor queries', *Proc of the 20th Int Conf on Data Engineering*. Los Alamitos: IEEE Computer Society.
22. Papadias D, Tao Y, Mouratidis K (2005), 'Aggregate nearest neighbor queries in spatial databases', *ACM Transactions on Database Systems (TODS)*, 2005, 30(2).
23. Song, M.B., K.J. Park, K.S. Kong and S.K. Lee, (2007), 'Bottom-up nearest neighbor search for R-trees'. *Inform. Proc. Lett.*, 101.
24. Sultana N, Hashem T, Kulik L, (2014), 'Group nearest neighbor queries in the presence of obstacles', *Proc of the 22th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*.
25. Sun Wei-Wei, Chen Chu-Nan, Zhu Liang, et al. (2015), 'On efficient aggregate nearest neighbor query processing in road networks', *Journal of Computer Science and Technology*, 30(4).
26. Song Li, Li-Ping Zhang, Zhong-Xiao Hao (2015), 'Strong neighborhood pair query in dynamic dataset', *Journal of Computer Research and Development*, 52 (3).
27. Son W, Sang W B, Ahn H K (2015), 'Group nearest-neighbor queries in the plane', *Theoretical Computer Science*.

28. Sun Dongpu, Hao Xiaohong, Gao Shuang (2013), 'Probabilistic group nearest neighbor queries based on uncertain voronoi diagram', *Journal of Beijing University of Agriculture*, , 28(4).
29. Tao, Y., D. Papadias and Q. Shen (2002), 'Continuous nearest neighbor search', *Proceedings of 28th International Conference on Very Large Data Bases(VLDB)*, Aug. 20-23, Hong Kong, China.
30. Wang YQ, Zhang R, Xu CF (2014), 'Continuous visible k nearest neighbor query on moving objects', *Information Systems*, 44.
31. Yiu M L, Mamoulis N, Papadias D (2005), 'Aggregate nearest neighbor queries in road networks', *IEEE Transactions on Knowledge and Data Engineering*, 17(6).
32. Yang Shiyu, Cheema Muhammad Aamir, Lin Xuemin, et al. (2017), 'Reverse k nearest neighbors queries and spatial reverse top-k queries', *VLDB JOURNAL*, 26(2).
33. Yiu M, Mamoulis N, Dai X (2009), 'Efficient evaluation of probabilistic Advanced Spatial queries on existentially uncertain data', *Transactions on Knowledge and Data Engineering*, 2009.
34. Zhang Y, Lin X, Zhu G, et al. (2010) 'Efficient rank based k-NN query processing over uncertain data', *Proc of the 26th Int Conf on Data Engineering (ICDE)*, Piscataway, Mar 1-6, 2010. NJ: IEEE.
35. Zhang Bin, Jiang Tao, Bao Zhifeng, Wong Raymond Chi-Wing , Chen Li (2016) 'Monochromatic and bichromatic reverse top-k group nearest neighbor queries', *Expert Systems with Applications*, 53(7).
36. Zhang Li-Ping, Zhao Ji-Qiao, Li Song (2014) 'Research on methods of construction of Voronoi diagram and nearest neighbor query in constrained regions', *Computer Science*, 41(9).
37. Nurhendratno, S. S. *et al.* (2018) 'Query optimization on distributed database dengue fever by minimizing attribute involvement', *Journal of Computer Science*, 14(4). doi: 10.3844/jcssp.2018.466.476.
38. Nurhendratno, S. S., Budiman, F. and Sudaryanto (2017) 'Design model integration and synchronization between surveillance units to support data warehouse epidemiology', *Journal of Theoretical and Applied Information Technology*, 95(3).
39. SUDARYANTO *et al.* (2019) 'Query optimization on distributed health database dbd for supporting data center with materialized view and minimizing attribute involvement', *Journal of Theoretical and Applied Information Technology*, 97(11).