

Pemrosesan Query Multi Terms Pada Mesin Pencari Web Semantik

Urip T. Setijohatmo, Setiadi Rachmat

Jurusan Teknik Komputer dan Informatika, Politeknik Negeri Bandung, Bandung 40012

E-mail : {urip, setiadi}@jtk.polban.ac.id

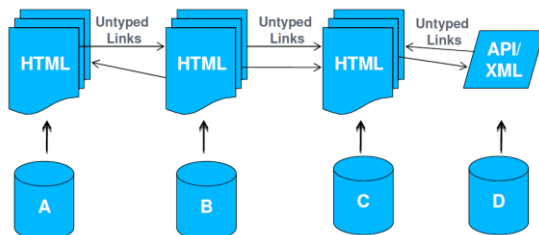
ABSTRAK

Semantik web menurut ilmuwan IT merupakan pengembangan dari web 2.0 yang sekarang sedang digunakan, dan adalah generasi berikutnya dari teknologi web. Berbagai penelitian tentang ontology, description logic, representasi dan teknologi penyimpanan dan retrieval telah dilakukan. Penggunaan web sebagai sumber pengetahuan menjadikan mesin pencari mendapatkan perhatian para peneliti IT. Komponen mesin pencari yang penting dalam menentukan urutan hasil pencarian adalah komponen ranking. Penelitian ini berfokus pada mekanisme retrieval khususnya pemrosesan query, dan merupakan lanjutan penelitian sebelumnya oleh Hogan dkk, dimana Hogan mengusulkan penanganan term tunggal query menggunakan metode perankingan ReconRank. Metode ReconRank bekerja dengan cara membuat topical subgraph berdasarkan keyword yang dimasukkan user, dari topical subgraph tersebut dibuat graph resources dan graph contexts. Kedua graph tersebut kemudian disatukan sehingga membentuk sebuah graph ReCon, lalu setiap vertex pada graph ReCon dihitung nilai rankingnya. Penelitian ini memperluas query menangani multi term. Metode yang digunakan adalah membentuk topical subgraph menghubungkan literal-literal yang match dengan multi term yang dimasukkan user. Atas dasar topical subgraph ini selanjutnya diekstrak path-path menghubungkan topical subgraph dan dibobotkan dengan ReconRank untuk path-path berukuran panjang sama, sedangkan untuk path ukuran berbeda diperingkat dimana path berukuran lebih pendek memiliki peringkat lebih tinggi. Semua path tersebut selanjutnya dibentuk perintah SPARQL (code generator).

Kata Kunci : pemrosesan query, ontology, graph, ReconRank, SPARQL.

1. Pendahuluan

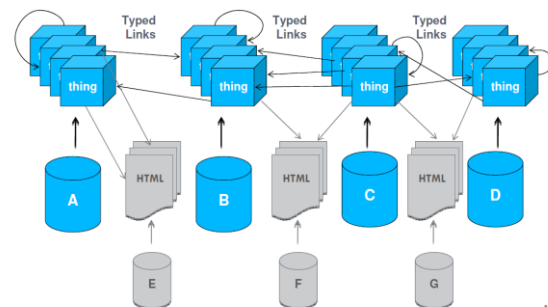
Penggunaan web sebagai sumber pengetahuan dan data sudah merupakan kebutuhan bagi hampir semua orang dan penggunaannya yang semakin meningkat membuat beberapa peneliti bidang IT melakukan pengembangan terhadap teknologi yang ada sekarang (web 2.0).



Gambar 1. Web 2.0

Web saat ini merupakan web dokumen dimana web dokumen dirancang untuk konsumsi manusia dimana objek utama adalah dokumen dan link antara dokumen. Salah satu pencapaian dari penelitian untuk mengembangkan web 2.0 tersebut

telah diperkenalkan konsep dan teknologi web berikutnya yaitu web semantik (web 3.0) seperti disebutkan salah satunya pada [1].



Gambar 2. Web Semantik

Web semantik adalah web of data. Hal yang menjadi fokus utama adalah “things” (description of things). Web semantik menggambarkan link antar “things”. Web semantik didesain untuk mesin dan manusia. Berbagai penelitian web semantik mulai dari konsep ontology, description logic sebagai basis matematik logika, representasi serta teknologi penyimpanan dan retrieval telah dilakukan. Penelitian ini berfokus pada mekanisme retrieval khususnya pemrosesan query, dan merupakan lanjutan penelitian sebelumnya yang telah dilakukan oleh Hogan dkk pada [5], dimana Hogan mengusulkan penanganan term tunggal query menggunakan metode perankingan ReconRank. Penelitian ini memperluas penanganan term dengan mempertimbangkan multi term query bersifat

independen. Sebagai ilustrasi term tunggal adalah “komputer” dan multi term misalkan “petani komputer”. Sedangkan independen berarti masing-masing n term mempunyai arti berbeda tentang n objek, seperti “petani” dengan “komputer”. Sebaliknya dengan “blue sea” yang merupakan dependen atau satu makna tentang satu objek.

Kontribusi penelitian ini adalah tiga hal yaitu, pertama membentuk topical subgraph untuk multi term dan kedua menyesuaikan algoritme pemeringkatan dan membentuk algoritme detil yang tidak ditemukan pada [5] serta ketiga code generator SPARQL.

2. Tinjauan Pustaka

Pada bagian ini akan dijabarkan pustaka-pustaka yang relevan, dikelompokkan sebagai mesin pencari dan metode pemeringkatan hasil pencarian.

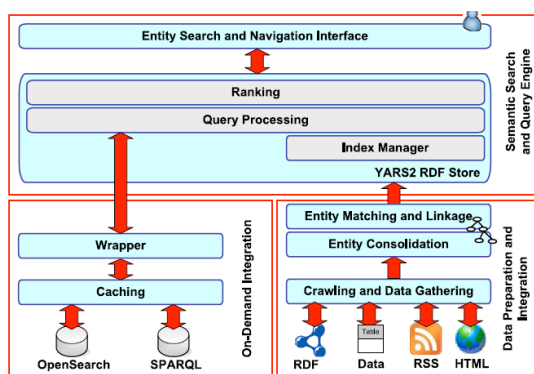
- Mesin Pencari (search engine)

Pada saat ini mesin pencari yang populer adalah Google, Altavista dan Bing untuk disebut beberapa, dan mereka adalah dari jenis web 2.0. Selain kemampuannya yang sudah baik, namun sering juga pengguna merasakan informasi yang dicarinya tidak sesuai. Lebih jauh lagi metode yang digunakan dapat memungkinkan pemeringkatan hasil pencarian termanipulasi sedemikian sehingga mendahulukan situs-situs tertentu. Juga mesin pencari tersebut tidak dapat menjawab pencarian seperti mencari lagu yang dinyanyikan John Lennon dan istrinya atau mencari teman dari seseorang [2][3]. Sedangkan untuk web semantik beberapa wakilnya untuk disebutkan beberapa adalah Swoogle, SWSE, dan Watson.

Komponen utama mesin pencari baik web 2.0 maupun web semantik adalah Crawler, Extractor, Ranker, dan Query Processor [4]. Gambar 3 memperlihatkan arsitektur mesin pencari web semantik SWSE menggunakan arsitektur Yars2. Materi mesin pencari ini digunakan oleh penulis sebagai landasan arsitektur untuk implementasi hasil penelitian dan melakukan eksperimen query.

- Metode Pemeringkatan Hasil Pencarian

Pada pemeringkatan hasil pencarian Google menggunakan metode PageRank, sedangkan Hogan pada [5] mengusulkan menggunakan ReConRank, yaitu sebuah algoritma yang dikembangkan dari



Gambar 3. Arsitektur Yars2

PageRank untuk menentukan urutan resources dalam file RDF beserta context-nya untuk meningkatkan kualitas pengurutan. ReConRank merupakan gabungan dari ResourceRank dan ContextRank.

Pendekatan yang digunakan berfokus pada topical subgraph. Topical subgraph adalah subgraph yang merepresentasikan hasil query term yang sesuai dengan keyword yang dimasukkan user. Pengurutan dilakukan hanya pada topical subgraph tersebut. Hasilnya adalah daftar resources yang diurutkan berdasarkan popularitas mereka dalam topical subgraph. Mesin pencari web semantik yang telah menggunakan metode pemeringkatan ReConRank adalah SWSE [6]. Algoritme ReConRank akan dijabarkan perilaku detil pada Hasil dan Pembahasan. Subjek kajian Metode ReConRank ini digunakan penulis untuk dijadikan basis mengembangkan dari term tunggal menjadi *multi term*.

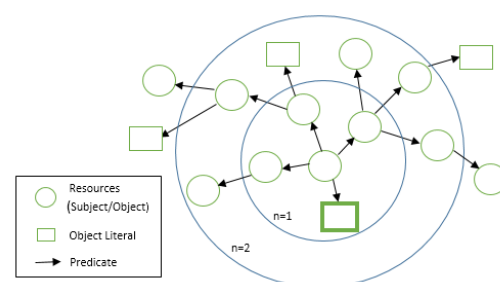
3. Hasil Dan Pembahasan

3.1. Analisis Algoritme ReConRank

Algoritma ReConRank adalah urutan langkah-langkah untuk menghitung nilai ranking dari vertex yang ada pada graph ReCon, dan graph ReCon dibentuk dari topical subgraph.

Topical subgraph dibuat berdasarkan pada keyword yang dimasukkan oleh user. Topical subgraph dibentuk dengan cara mencari object literal yang cocok dengan keyword dan dapatkan subject dari object tersebut. Setelah subject didapatkan, ambil semua predicate dan object terkait subject tersebut sebanyak satu hop ($n=1$). Semakin kecil nilai hop maka ukuran topical subgraph makin kecil, hal itu dapat meningkatkan nilai presisi hasil pencarian. Gambar 4 mengilustrasi topical subgraph, dimana kotak bergaris tebal hijau merupakan object literal yang cocok dengan keyword dan lingkaran biru merupakan topical subgraph untuk $n=1$ dan $n=2$.

Berikut adalah detil algoritma ReConRank.



Gambar 4. Topical Subgraph-Illustrasi

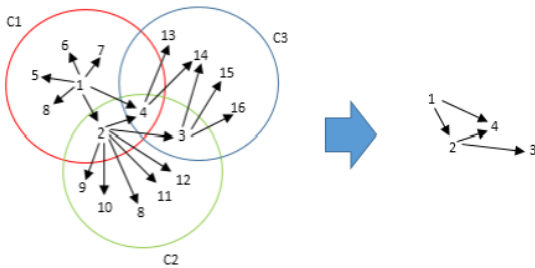
1. Proses “Create topical subgraph”

Masukan dari proses ini adalah keyword dari user dan data yang ada pada quad store. Keluarannya adalah topical subgraph, dimana vertex adalah subject atau object dan edge adalah predicate dan context. Topical subgraph dibentuk dengan cara mencari object literal yang cocok dengan keyword dan dapatkan subject dari object tersebut. Setelah subject didapatkan, ambil semua predicate dan object terkait subject tersebut sebanyak satu hop ($n=1$). Proses pembentukan topical

subgraph dari Hogan [5] tidak akan digunakan karena yang akan dikembangkan berkenaan dengan multi term.

2. Proses “Create Resources Graph”

Graph resources merupakan subgraph dari topical subgraph, dimana vertex-nya adalah resources dan edge-nya adalah predicate antar resources tersebut. Resources yang terpilih untuk berada pada graph resources ini adalah resources yang pernah berada diposisi sebagai subject. Berdasarkan [5] ada empat resources yang pernah berperan sebagai subject, jadi graph resources yang terbentuk terdiri dari keempat resources beserta predicate antar resources tersebut (Gambar 5-sebelah kanan).



Gambar 5. Pembentukan Resources Graph

```
Function CreateGraphResources (topicalSubGraph:
Graph):Graph
begin
  Type Edge = record
    v : String;
    w : String;
  End;
  graphResource : Graph
  E, e : set of namedEdge
  v1, v2 : Vertex
  Hv : set of String
  edges : set of Edge
  Hv = e.v ∈ EtopicalSubgraph
  foreach v1 ∈ Hv do
    foreach v2 ∈ Hv do
      if v1 ≠ v2 then
        if v1 and v2 ∈ EtopicalSubgraph.v)
        then begin
          edges.v = v1
          edges.w = v2
          graphResource = graphResource U
            edges
        endif
      endfor
    endfor
  return graphResource
end
```

3. Proses “Create Context Graph”

Graph context merupakan subgraph dari topical subgraph, dimana vertexnya adalah context dan edgenya adalah hubungan antara context tersebut. Berbeda dengan graph resources yang hubungan antar vertexnya eksplisit terlihat, hubungan antar context pada graph context tidak selalu eksplisit. Jika itu terjadi dibuatlah sebuah *virtual predicate*. Berdasarkan Gambar 6, ada resources yang berada dalam dua context yang berbeda. Adanya irisan tersebut menentukan adanya *virtual predicate* antar kedua context tersebut. Resources nomor 2 dan 4 berada didalam C1 dan C2. Berdasarkan pada studi terhadap [5], maka ada tiga kondisi yang

menunjukkan adanya hubungan antara dua context, E(c1, c2), yaitu :

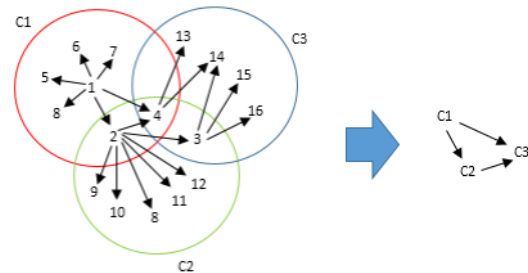
Kondisi pertama, jika ada *object* pada context 1 yang menjadi *subject* pada context 2:

s1, p1, o1, c1 dengan o1, p2, o2, c2

Kondisi kedua, jika ada dua *subject* pada context yang berbeda berhubungan dengan *object* yang sama: s1, p1, o1, c1 dengan s2, p2, o1, c2

Kondisi ketiga, jika ada satu *subject* pada context yang berbeda berhubungan dengan *object* yang sama: s1, p1, o1, c1 dengan s1, p2, o2, c2

Gambar 6 adalah hasil pembentukan graph context, C(V,E), dimana V adalah context dan E adalah virtual predicate.



Gambar 6. Pembentukan Graph Konteks

```
Function CreateGraphKonteks (topicalSubGraph:
Graph):Graph
BEGIN
  type Edge = record
    v, w : String;
  End;
  graphContext :Graph = null
  edges : set of Edge
  v1,v2,w1,w2,n1,n2,c1,c2 : String
  Hv, Hw, Hc, Hn : array of string
  //ambil semua anggota v, w, n, dan c pada
    EtopicalSubgraph
  Hv = e.v ∈ EtopicalSubgraph
  Hw = e.w ∈ EtopicalSubgraph
  Hn = e.n ∈ EtopicalSubgraph
  Hc = e.c ∈ EtopicalSubgraph
  foreach v1 ∈ Hv do
    foreach v2 ∈ Hv do
      foreach w1 ∈ Hw do
        foreach w2 ∈ Hw do
          foreach n1 ∈ Hv do
            foreach n2 ∈ Hv do
              foreach c1 ∈ Hw do
                foreach c2 ∈ Hw do
                  if ((v1,n1,w1,c1∈EtopicalSubgraph) ∧
                    (v1,n2,w2,c2∈ EtopicalSubgraph)) ∨
                    ((v1,n1,w1,c1 ∈ EtopicalSubgraph) ∧
                    (v2,n2,w1,c2∈ EtopicalSubgraph)) ∨
                    ((v1,n1,w1,c1 ∈ EtopicalSubgraph) ∧
                    (w1,n2,w2,c2∈ EtopicalSubgraph))
                  then begin
                    edges.v = c1
                    edges.w = c2
                    graphContext = graphContext U
                      edges
                  endif
                endfor
              endfor
            endfor
          endfor
        endfor
      endfor
    endfor
  endfor
```

```

    endfor
  endfor
endfor
return graphResource
end

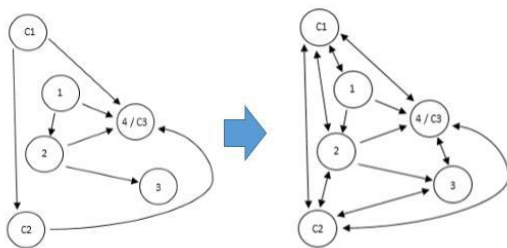
```

4. Proses “Create ReCon Graph”

Pada proses ini dibentuk sebuah graph ReCon yang merupakan gabungan dari graph resources dan graph context, dimana vertex-nya adalah resources dan context, edge-nya adalah hubungan antar resources, antara resources dan context, dan hubungan antar context. Untuk menentukan hubungan antara kedua graph tersebut, dibuat *implied link* dengan kondisi sebagai berikut:

- Kondisi 1: hubungan antara context dan resource yang dikandungnya.
- Kondisi 2: hubungan antara resource dan context yang mengandungnya.
- Kondisi 3: hubungan antara context dengan context, misalkan sebuah resource pada context A memiliki link ke context B, maka dibuat link antara context A dengan context B.

Langkah yang harus dilakukan adalah menggabungkan graph resources dan context. Setelah itu tentukan hubungan antara resources dan context yang ada. Pertama semua elemen graph context dimasukkan kedalam graph ReCon, setelah itu baru semua element graph resource dimasukkan sekaligus tentukan resource tersebut memiliki hubungan dengan context mana saja. Lalu buat edge antara resource dan context tersebut. Untuk mengetahui hubungan resource dengan context dapat menggunakan graph yang awal yaitu topical subgraph. Pada Gambar 7 terdapat topical subgraph. Untuk kondisi yang pertama dapat diketahui bahwa resource no 1,2, dan 4 memiliki hubungan dengan C1, resource 2 dan 4 memiliki hubungan dengan C2, resource 3 dan 4 memiliki hubungan dengan C3. Kondisi kedua sama dengan kondisi pertama tapi dengan arah sebaliknya, C1 ada hubungan dengan resource 1,2,dan 4. Contoh untuk kondisi ketiga: resource 2 merupakan anggota dari C2, resource 2 berhubungan dengan C1, maka C2 berhubungan dengan C1.



Gambar 7. Pembentukan Subgraph ReCon

```

Function createReconGraph(resourceGraph: Graph,
contextGraph: Graph): Graph
begin
  type Edge = record
    v, w : String;
  end;
  reconGraph : Graph = null

```

```

  v : vertex
  edges : set of Edge
  c1,c2 : String
Algoritma :
  reconGraph = reconGraph U contextGraph U
    resourceGraph
  foreach v ∈ EresourceGraph do
    if v ∈ EtopicalSubgraph.v then begin
      c1 = EtopicalSubgraph.c
      edges.v = v
      edges.w = c1
      reconGraph = reconGraph U edges
      edges.v = c1
      edges.w = v
      reconGraph = reconGraph U edges
    foreach v2 ∈ EtopicalSubgraph.v
      c2 = EtopicalSubgraph.c
      if v = v2 then
        if c1 != c2 then
          edges.v = c1
          edges.w = c2
          reconGraph = reconGraph U edges
          edges.v = c2
          edges.w = c1
        endif
      endif
    endfor
  endfor
  return reconGraph
end

```

5. Proses “Calculate Ranking”

Proses ini menghitung nilai ranking untuk setiap vertex pada graph. Nilai ranking setiap vertex pada graph didapatkan dengan cara menghitung nilai *eigenvector*. Langkah yang harus dilakukan untuk menghitung nilai tersebut adalah

- Representasikan graph dalam bentuk matrix.
- Hitung nilai eigenvector awal.
- Hitung nilai dasar untuk setiap node.
- Hitung nilai eigenvector dengan cara mengalikan nilai eigenvector sebelumnya dengan matrix.
- Hitung L1 norm nya dengan cara mengurangi nilai norma eigenvector pada iterasi sekarang dengan nilai norma eigenvector pada iterasi sebelumnya. Lakukan pengulangan terhadap langkah c-d-e sampai pada kondisi jika nilai L1 norm tersebut sudah mencapai toleransi tertentu (misal 0.001). Maka eigenvector yang merupakan nilai ranking telah ditemukan.

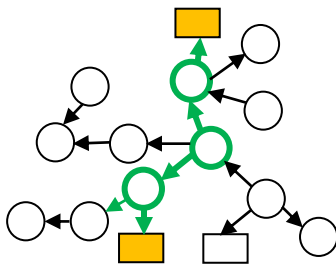
```

Function HitungReconRank(reconGraph: Graph):
double
begin
  double rank[], oldRank[], llresidual,
    min,double M[][];
  M = creatConnecMatrix(reconGraph)
  M = createTransitionalMatrix(M)
  oldR = nilai inlink tiap vertex dibagi
    jumlahSemuaLink
  do begin
    min = (jmlRankingDeadNode /
      jumlahNode) + (jmlRanking
        liveNode * 0.15/jumlahNode)
    rank = oldRank * M * 0.85 + min
    llresidual = |rank - oldRank|
  end while (llresidual>0.001)
  return rank
end

```

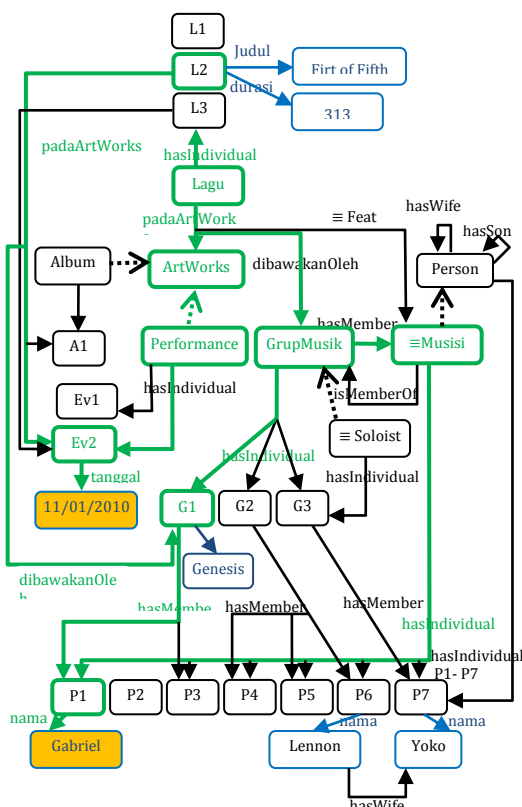
3.2. Perluasan ReConRank-XReconRank

Prinsip kerja XreconRank (perluasan ReconRank) adalah untuk setiap term tunggal dari n term dicari topical subgraphnya dengan cara membentuk path yang menghubungkan n term terpisah tersebut. Prinsip kerja ini di dasari oleh suatu logika bahwa multi term yang dimasukkan bermaksud membentuk suatu ide atau konsep utuh tentang sesuatu dari multi term. Perluasan ReConRank berkisar pada pembentukan topical subgraph dari term tunggal menjadi multi term dan bagaimana pemeringkatannya. Pada Gambar 8 kotak berwarna kuning adalah literal yang match dengan terms query yang dimasukkan. Subgraph yang terbentuk merupakan subgraph yang menghubungkan dua literal ini (warna hijau) membentuk topical subgraph. Subgraph tersebut merupakan jawaban yang dicari berdasarkan ontology yang tersedia. Untuk penjelasan lebih detail berikut diberikan ontology musik yang disederhanakan, diikuti dengan tabel berisi RDF dalam bentuk kalimat tersusun sebagai tripple SPO. Baris berwarna abu merupakan kalimat yang objeknya adalah literal.



Gambar 8. Subgraph Konektivitas Dua Literal

Pada Gambar 9 path berwarna hijau menyatakan konektivitas subgraph mempertemukan objek literal “Gabriel” dan literal “11/01/2010”.



Gambar 9. Topical Subgraph Term “Gabriel” dan “11/01/2010” dari Suatu Ontology Musik

Demikian pula isian pada Tabel-1 berwarna hijau merupakan triple yang mewakili path.

Tabel-1. Kalimat Triple Tersusun dari Ontology Musik

S	P	O
Album	isA	ArtWorks
padaArtWorks	dibawakanOleh	GrupMusik
padaArtWorks	feat	Musisi
GrupMusik	hasMember	Musisi
Musisi	isMemberOf	GrupMusik
Soloist	isA	GrupMusik
Musisi	isA	Person
Person	hasWife	Person
L2Ev2	dibawakanOleh	G1
L3A1	dibawakanOleh	G3
L3Ev2	dibawakanOleh	G1
L3Ev2	feat	P6
Person	hasSon	Person
P7	hasWife	P6
P6	hasSon	P2
P7	hasSon	P2
L2	padaArtWorks	Ev2
L3	padaArtWorks	A1
L3	padaArtWorks	Ev2
Performance	isA	ArtWorks
G1	hasMember	P1
G1	hasMember	P3
G2	hasMember	P4
G2	hasMember	P5
G2	hasMember	P6
G3	hasMember	P7
Lagu	padaArtWorks	ArtWorks
L1	judul	Yesterday
L1	durasi	270
L2	judul	Firth of Fifth
L2	durasi	360
L3	Judul	Hey Jude
L3	durasi	300
Lagu	hasIndividual	L1
Lagu	hasIndividual	L2
Lagu	hasIndividual	L3
Album	hasIndividual	A1
Album	hasIndividual	A2
Performance	hasIndividual	Ev1
Performance	hasIndividual	Ev2
GrupMusik	hasIndividual	G1
GrupMusik	hasIndividual	G2
GrupMusik	hasIndividual	G3
Person	hasIndividual	P1
Person	hasIndividual	P2
Person	hasIndividual	P3
Person	hasIndividual	P4
Person	hasIndividual	P5
Person	hasIndividual	P6
Person	hasIndividual	P7
Musisi	hasIndividual	P1
Musisi	hasIndividual	P3
Musisi	hasIndividual	P4
Musisi	hasIndividual	P5
Musisi	hasIndividual	P6
Musisi	hasIndividual	P7
Soloist	hasIndividual	G3

padaArtWorks	hasIndividual	L2Ev2
padaArtWorks	hasIndividual	L3A1
padaArtWorks	hasIndividual	L3Ev2
A1	judul	Selling England
Ev1	tanggal	10/11/2009
Ev2	tanggal	11/01/2010
G1	nama	Genesis
G2	nama	Beatles
G3	nama	YokoSolo
P1	nama	Gabriel
P2	nama	SonLennonYoko
P3	nama	Rutherford
P4	nama	Mc Cartney
P5	nama	Ringo
P6	nama	Lennon
P7	nama	Yoko

Permasalahan yang cocok untuk hal ini adalah masalah konektivitas graph dapat diselesaikan dengan traversing DFS, BFS, k-hops atau lainnya seperti random walks. Seperti yang diilustrasikan pada term Query “Gabriel” dan “11/01/2010” bermaksud mencari satu kesatuan ide apa dengan Gabriel pada tanggal tersebut. Sehingga di dalam graph, konsep yang dicari dari setiap term adalah berkaitan. Selanjutnya satu kesatuan topical subgraph ini menjadi masukan algoritme ReConRank untuk pemeringkatan.

Adapun perintah SPARQL untuk menjawab query dengan masukan multi term ini persis mewakili sekumpulan kalimat yang merepresentasikan topical subgraph yang terbentuk (Untuk kasus ontology musik terlihat sebagai kalimat berwarna hijau pada tabel.), maka salah satu hasil code generator adalah:

```

Select ?out ?out2
where
  ?padaArtWorks ?dibawakanOleh ?GrupMusik
  ?GrupMusik ?hasMember ?Musisi
  ?Musisi ?isMemberOf ?GrupMusik
  ?L2Ev2 ?dibawakanOleh ?G1
  ?L2 ?padaArtWorks ?Ev2
  ?Performance ?isA ?ArtWorks
  ?G1 ?hasMember ?P1
  ?G1 ?hasMember ?P3
  ?Lagu ?padaArtWorks ?ArtWorks
  ?L2 ?judul ?out1
  ?L2 ?durasi ?out2
  ?Performance ?hasIndividual ?Ev2
  ?padaArtWorks ?hasIndividual ?L2Ev2
  ?Ev2 ?tanggal ?isi
  ?P1 ?nama ?Gabriel

```

Filter

(?isi = “11/01/2010” and ?Gabriel = “Gabriel”)

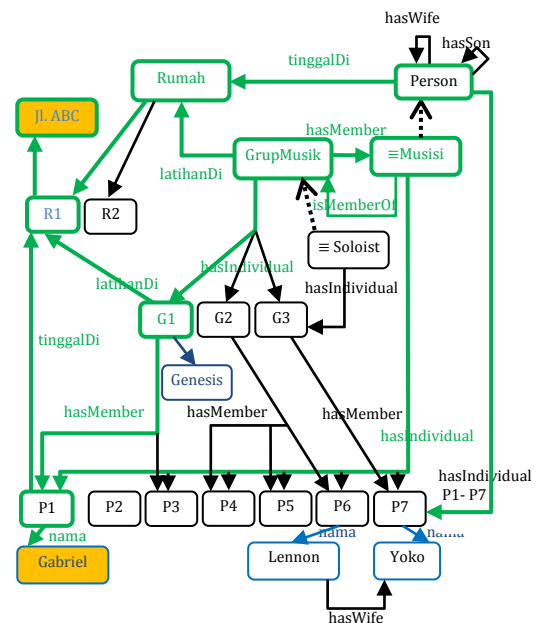
Pada suatu topical subgraph terdapat banyak path yang menghubungkan literal2 yang match dengan multi term masukan user. Maka diperlukan pembobotan dan pemeringkatan path sehingga jawaban ke user dipersembahkan dengan urutan tertentu. Berikut adalah analisa pemeringkatan menggunakan model ontology musik yang sedikit diperluas mewakili tempat tinggal dan latihan grup musik, dimana di sini hanya ditampilkan perluasannya saja.

Anggap term masukan adalah literal “Gabriel” dan “Jl. ABC”. Hasilnya adalah topical subgraph seperti Gambar 10.

Untuk mencapai dari “Gabriel ke “Jl. ABC” terdapat dua path:

- Gabriel-nama-P1-hasMember-G1-latihanDi-R1-Jl.ABC
- Gabriel-nama-P1-memilikiRumah-Jl.ABC

Yang lebih bermakna langsung adalah ketika path yang dilewati lebih pendek, karena memiliki arti yang langsung dibandingkan ketika melewati node-node lain yang lebih panjang. Melewati path lebih panjang berarti memiliki semantik meliputi juga konsep-konsep lain. Pada kasus di atas, yang memiliki arti langsung adalah Gabriel tinggal di jl ABC, sedangkan yang lebih panjang berarti Gabriel merupakan anggota Musisi G1 yang berlatih di Jl. ABC.



Gambar 10. Topical Subgraph term "Gabriel Jl.ABC"

Karena diperlukan dua jawaban tersebut untuk diberikan ke user, maka tidak akan dihilangkan salah satu, tetapi hanya diperingkat dimulai path yang terpendek diikuti dengan yang lebih panjang, dst. Dengan demikian algoritme untuk permasalahan ini terdiri dari:

- pembentukan topical subgraph dari literal masukan multi term dari user
- pembobotan dan pemeringkatan

path terpendek yang menghubungkan literal masukan user adalah berbobot tertinggi. Jika terdapat lebih dari satu path panjang sama, gunakan pembobotan ReconRank.

- generate SPARQL untuk setiap path tersebut.

4. KESIMPULAN

Penelitian ini merupakan perbaikan dari penelitian sebelumnya dari Hogan [5] dengan mengembangkan topical subgraph untuk menangani multi term dan penyesuaian pemeringkatan ReconRank. Pencapaian penelitian saat ini adalah telah menghasilkan desain mekanisme pembentukan topical subgraph untuk multi term dan detail algoritme ReconRank. Implementasi baru akan mempersiapkan tempat penyimpanan quadruple dengan virtuoso dan query SPARQL.

UCAPAN TERIMA KASIH

Sebagai ucapan terima kasih kami tujukan kepada panitia IRWNS dan pihak terkait yang telah menjembatani terlaksananya publisitas penelitian ini, juga rekan dosen atas atmosfir penelitian yang menginspirasi.

DAFTAR PUSTAKA

- [1]. Neumayer, *Semantic and Distributed Entity Search in the Web of Data*, PhD Thesis, Norwegian University of Science and Technology Faculty of Information Technology, Mathematics and Electrical Engineering Department of Computer and Information Science, Trondheim, October 2012
- [2]. Harth, A., Hogan, A., Delbru, R., Umbrich, J., O'Riain, S., & Decker, S. (2007). SWSE: Answers Before Links!.
- [3]. Harth, A., Hogan, A., Umbrich, J., & Decker, S. (2007). SWSE: A Search and Query Engine for Web Data. National University of Ireland.
- [4]. Langville, A. N., & Meyer, C. D. (2011). *Google's PageRank and beyond: The science of search engine rankings*. Princeton University Press.
- [5]. Aidan Hogan, Andreas Harth, and Stefan Decker, *ReConRank: A Scalable Ranking Method for Semantic Web Data with Context*
- [6]. R Usbeck, *Combining Linked Data and Statistical Information Retrieval*, Next Generation Information Systems, 2014.