ANALISA VISUAL MENGGUNAKAN ETETOOLKIT FRAMEWORK TERHADAP PENYAKIT BETA-THALASSEMIA DI JAWA TENGAH BAGIAN SELATAN

Moh Reza Syaifur Rizal[1], Wayan Tunas Artama[2], Rohmatul Fajriyah[3], Izzati Muhimmah[4], Lantip Rujito[5], Lalu Mutawali[6]

[1,] Jurusan Teknik Informatika, Universitas Islam Indonesia, jln Kaliurang Km. 14,5, Yogyakarta, Krawitan, Umbulmartani, Ngemplak, Kabupaten Sleman, Daerah Istimewa Yogyakarta 55584

[2,] Pusat Studi Bioteknologi, Universitas Gajah Mada, jln Teknika Utara, Kocoran, Caturtunggal, Depok, Kabupaten Sleman, Daerah Istimewa Yogyakarta 55281

[3,] Jurusan Statistika, Universitas Islam Indonesia, jln Kaliurang Km. 14,5, Yogyakarta, Krawitan, Umbulmartani, Ngemplak, Kabupaten Sleman, Daerah Istimewa Yogyakarta 55584

[4,] Jurusan Teknik Informatika, Universitas Islam Indonesia, jln Kaliurang Km. 14,5, Yogyakarta, Krawitan, Umbulmartani, Ngemplak, Kabupaten Sleman, Daerah Istimewa Yogyakarta 55584

[5,] Laboratorium Riset Biologi Molekuler Fakulatas Kedokteran dan ilmu kesehatan, jln Professor DR. HR Boenyamin, Dukuhbandong, Grendeng, Purwokerto utara, Kabupaten Banyumas, Jawa Tengah 53122

[6,] Jurusan Teknik Informatika, STMIK LOMBOK, jln Basuki Rahmat, Praya, Mataram, Kabupaten Lombok Tengah, Nusa Tenggara Barat 83511

[1] 16917112@students.uii.ac.id, [2] artama@ugm.ac.id, [3] rohmatul.fajriyah@uii.ac.id, [4] emma@fti.uii.ac.id, l.rujito@unsoed.ac.id[5], laluallistilo@gmail.com[6]

Abstract

Detection of biomolecular events in the visual to be analyzed using computational to detect effectiveness and accuracy of disease. As the main result, many visual analyzes, ranging from gene clustering to phylogenetic, produce hierarchical trees. The Tree Exploration Environment (ETE) toolkit that assists in automated hierarchical trees manipulation, analysis and visualization. Then, in this paper, list β-thalassemia mutations that are a group of hereditary blood disorders characterized by anomalies in the synthesis of the hemoglobin beta chains resulting in variable phenotypes ranging from severe anemia to clinically asymptomatic individuals. This result is ETEToolkit can elaborate with these Mutations to showing through tree and alignment in one frame, then we can customize and render into PDF image. These mutations located in the center of Java, Indonesia.

***Keywords** : β-thalassemia, Mutations, ETEToolkit, Central Java*

Abstrak

Deteksi peristiwa biomolekuler dalam visual yang akan dianalisis menggunakan komputasi untuk mendeteksi efektivitas dan akurasi penyakit. Sebagai hasil utama, banyak analisis visual, mulai dari pengelompokan gen hingga filogenetik, menghasilkan pohon hierarkis. Toolkit Lingkungan Eksplorasi Pohon (ETE) yang membantu manipulasi, analisis, dan visualisasi pohon hierarkis otomatis. Kemudian, dalam makalah ini, daftar mutasi β-thalassemia yang merupakan kelompok kelainan darah herediter yang ditandai oleh anomali dalam sintesis rantai beta hemoglobin yang menghasilkan berbagai fenotipe mulai

dari anemia berat hingga individu tanpa gejala klinis. Hasil ini adalah ETEToolkit dapat menguraikan mutasi ini untuk ditampilkan melalui pohon dan penyelarasan dalam satu bingkai, kemudian kita dapat menyesuaikan dan merender ke dalam gambar PDF. Mutasi ini berlokasi di pusat Jawa, Indonesia.

Kata kunci : β-talasemia, Mutasi, ETEToolkit, Central Java

1. Pendahuluan

Teknologi biomolekuler, yang telah berkembang pesat selama 30 tahun terakhir, telah membuat revolusi besar dalam dunia genetika manusia yang memungkinkan studi variasi genetik, evolusi, dan pengelompokan populasi ke tingkat molekuler, selain terkait erat dengan epidemiologi suatu penyakit, dapat juga memberikan informasi tentang asal usul suatu populasi dan afinitas genetik antar populasi. Sehubungan dengan epidemiologi penyakit, telah dilaporkan bahwa berbagai penyakit genetik dan non-genetik memiliki pola distribusi tertentu dalam populasi (Cavalli 1997). Kerangka ETEToolkit (Hernanda et al. 2010) adalah salah satu teknologi pemrograman biomolekuler untuk analisis visual.

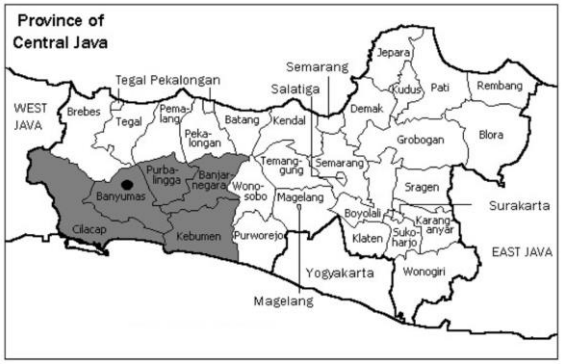
Environment Tree Exploration (ETE), toolkit pemrograman python untuk menganalisis, memanipulasi atau memvisualisasikan segala jenis hierarki pohon, memperluas fungsionalitas dalam toolkit lain dan memungkinkan tingkat kustomisasi yang tinggi. Meskipun kurang lengkap daripada editor mandiri, fitur menggambar ETE mengandalkan bahasa skrip python, yang memungkinkan untuk menggabungkan analisis pohon maju dan visualisasi pohon ke dalam satu program tunggal. ETE mengimplementasikan dua modul khusus untuk bekerja dengan pohon filogenetik dan pohon klaster. Ekstensi filogenetik memungkinkan pohon untuk dihubungkan ke masing-masing keberpihakan urutan ganda, termasuk dua algoritma untuk prediksi ortologi dan paralogi, dan mengimplementasikan penanggalan duplikasi yang dijelaskan dalam (Huerta et al. 2010) menyediakan akses ke database phylomeDB (Huerta et al. 2007). Demikian pula, pohon pengelompokan dapat dihubungkan ke data sumbernya, memungkinkan partisi pohon dianalisis menggunakan beberapa teknik validasi. Selain itu, ETE mengimplementasikan mesin gambar yang sepenuhnya dapat diprogram yang dapat digunakan untuk menghasilkan representasi pohon kustom yang dinamis dalam Gambar PDF.

Talasemia beta berasal dari bahasa Yunani, Thalassa (laut) dan haima (darah). Thalassemia beta mencakup tiga bentuk utama: Thalassemia Major, bervariasi disebut sebagai Cooley's Anemia dan Mediterranean Anemia, Thalassemia Intermedia dan Thalassemia Minor, juga disebut sebagai beta-thalassemia, sifat beta-thalassemia atau beta-thalassemia heterozigot. Terlepas dari bentuk dominan yang langka, subjek dengan talasemia mayor adalah homozigot atau heterozigot majemuk untuk gen beta^0 atau beta ^+, subjek dengan talasemia intermedia sebagian besar homozigot atau senyawa heterozigot dan subjek dengan talasemia minor sebagian besar heterozigot (Galanello dan Origa 2010).

Seperti yang ditemukan dalam penelitian sebelumnya, tingkat prevalensi untuk pembawa talasemia beta bervariasi dari 3,0 hingga 10,0 persen di seluruh wilayah Indonesia (Departemen Kesehatan 2010). Data sekitar 2500 hingga 5.000 bayi yang lahir dengan pola klinis Beta-Thalassemia mayor dan Beta-Thalassemia intermedia dirujuk setiap tahun kepada pemerintah (Lantip et al. 2015). Sebuah program nasional untuk mengatasi thalassemia belum diterapkan secara serius. Strategi penting untuk mendukung program pencegahan nasional adalah memahami spektrum mutasi talasemia dari berbagai etnis untuk mengembangkan program pencegahan yang sesuai berdasarkan alel mutasi lokal (Giordano et al, 2014). Beberapa studi memetakan beberapa mutasi lokal di kota-kota multi-etnis besar di Indonesia. Namun, studi ini terbatas pada ukuran sampel terbatas (Hernanda et al. 2012; Lantip et al. 2015; Setianingsih et al. 1998; Tamam et al. 2010). Studi ini mencoba untuk fokus pada populasi Jawa, salah satu kelompok etnis terbesar di Indonesia, khususnya di Jawa Tengah. Penelitian pemindaian molekuler ini telah dilakukan oleh "Lantip Rujito" pada makalahnya yang berjudul "Pemindaian molekuler Beta Thalassemia di Wilayah Selatan Jawa Tengah, Indonesia; sebuah langkah menuju program pencegahan lokal." Lihat peta Jawa Tengah pada Gambar 1.

Di atas kertasnya, 209 pasien pria dan wanita dikumpulkan, mulai dari usia 6 bulan hingga 65 tahun. Pasien berasal dari Yayasan Thalassemia Indonesia, Cabang Banyumas. Pasien yang menerima transfusi reguler dan tidak teratur

dimasukkan dalam penelitiannya. Hasil pemindaian molekul subjek ini mengungkapkan genotipe berikut: kodon 26 (Hb E; HBB: c.79G> A) / IVS-I-5 (HBB: c.95 + 5G> C) adalah yang paling sering (40,67%), diikuti oleh IVS-I-5 / IVS-I-5 sebesar 14,83%. Genotipe yang lebih jarang adalah: IVS-I-5 / kodon 41/42 (HBB: c.126 129delCTTT) (0,96%) dan IVS-I-5 / IVS-1-1 (HBB: c.92 + 2T> C), (0,48%). IVS-I-5 adalah yang paling umum di 43,5 persen, sedangkan kodon 40 (HBB: c.123delG) dan Cap + 1 (HBB: c.-50A> C) adalah yang paling jarang di 0,2 persen (Lantip et al. 2015). Namun sayangnya, tidak ada visual yang menarik untuk disajikan di meja publik saja dalam penelitian itu. Itu sebabnya penelitian ini untuk mengetahui / memvisualisasikan yang mana dapat disesuaikan untuk deteksi penyakit yang lebih representatif dan akurat.



Gambar 1. Peta Jawa Tengah dan Pusat Talasemia Banyumas, cakupan Banyumas, Indonesia (Lantip et al, 2015).

2. Tinjuan Pustaka

Penelitian-penelitian terdahulu yang telah terpublikasi yaitu penelitian dari Lantip Rujito yang berjudul "*Molecular scanning of b-thalassemia in southern region of Central Java, Indonesia; a step towards a local prevention program*" yang mana memindaikan pasien penderita *b-thalassemia* berikut dengan mutasi-mutasi tersebut. Kesimpulannya penelitian tersebut menganalisa visual hanya menggunakan merepresentasikan dalam bentuk tabel. Sehingga Analisa visual yang digunakan kurang beragam.
Penelitian selanjutnya dari Encarnacao yang berjudul "*Information Visualization*" menjelaskan tentang definisi, area kerja, dan tantangan dalam Analisa visual. hanya saja penjelasan di penilitian ini tidak menyinggung tentang *biomolecular.*
Penelitian selanjutnya dari Samuel Smith yang berjudul "*Visual Analytics for large Scale Bioinformatics Data Sets*" yaitu mejelaskan tentang menampilkan data besar gen dengan menngunakan *software Regulon Explorer*. Hanya saja visual tersebut tidak ditampilkan pengurutan dan pohon filogenetik(*Pyhlogenetic tree*).
Penelitian selanjutnya dari James Huerta Cepas yang berjudul "*A python Environment for (Phylogenetics) Tree Exploration*" menjelaskan sebuah framework yang ditulis dengan bahasa pemrograman python untuk menganalisa data gen dalam bentuk visual. penelitian tersebut menunjang dalam penulisan proses analisa di penelitian ini.
Penelitian selanjutnya dari Huson yang berjudul "*Dendroscope: an interactive viewer for large phylogenetic tress*" sebuah *software* yang menganalisa gen dengan hasil presentasinya menggunakan visual. hanya saja dalam aplikasi tersebut hanya menampilkan pohon saja.
Penelitian selanjutnya dari Notredame yang berjudul "*T-coffe: A novel method for fast and accurate multiple sequence alignment*" menejelaskan tentang sebuah *software* analisa gen dengan hasil pengurutan(*alignment*) dengan metode *progessive alignment*. Hanya saja *software* tersebut hanya menampilkan pengurutan saja.

3. Metodologi Penelitian

Pada bagian ini menjelaskan cara memvisualisasikan data dari genotipe di bagian ini. Data yang biasa digunakan adalah FASTA dan Newick. FASTA biasanya digunakan untuk mengurutkan banyak keberpihakan (Pearson 1990), sementara Newick menggunakan tanda kurung bersarang untuk mewakili struktur data hierarkis sebagai string teks. Standar Newick asli mampu menyandikan informasi topologi pohon, jarak cabang, dan nama simpul (Tim ETEToolkit. 2016). Format Fasta Karena kesulitan data privasi / rekam medis. Dalam ETEToolkit, genotipe hanya akan diproses 7 DNA, yaitu: IVS 1-5, IVS 1-1, Codon 15, HBE Codon 26, Codon 35, Codon 40, dan Codon 41/42.

3.1 Newick Format

Berikut ini adalah aturan di pohon. Dari braket terbuka ke braket ujung dengan banyak sub-kurung: ((HBB NORMAL, (Beta thalassemia IVS 1-5, Beta-thalassemia IVS 1-1, Beta-thalassemia Codon 15, Beta-thalassemia Codon 35, Beta-thalassemia Codon 35, Beta-thalassemia Codon 40), Beta-thalassemia 41/42), (HBE Codon 26))) Merupakan cabang pohon dan koma merupakan pemisah nama cabang.

3.2 Metode

ETE sepenuhnya ditulis dalam Python, bahasa pemrograman yang memberikan dukungan kuat untuk integrasi dengan bahasa dan alat lain yang

popularitasnya meningkat di antara komunitas bioinformatika (Huerta *et al*. 2016).

```
  from ete3 import PhyloTree, TreeStyle
alg ="""
>HBB Normal
TTTTTAGTAGCAATTTGTACTGATGGTATGGGGCCAAGAG
ATA...
>Beta talasemia IVS 1-5
TTTTTAGTAGCAATTTGTACTGATGGTATGGGGCCAAGAG
ATA...
>Beta talasemia IVS 1-1
TTTTTAGTAGCAATTTGTACTGATGGTATGGGGCCAAGAG
ATA...
>Beta talasemia Codon 15
TTTTTAGTAGCAATTTGTACTGATGGTATGGGGCCAAGAG
ATA...
>Beta talasemia Codon 35
TTTTTAGTAGCAATTTGTACTGATGGTATGGGGCCAAGAG
ATA...
Beta talasemia Codon 40
TTTTTAGTAGCAATTTGTACTGATGGTATGGGGCCAAGAG
ATA...
>Beta thalassemia Codon 41/42
TTTTTAGTAGCAATTTGTACTGATGGTATGGGGCCAAGAG
ATA...
>HBE Codon 26
TTTTTAGTAGCAATTTGTACTGATGGTATGGGGCCAAGAG
ATA...
"""
def get_tree():
    #menganalisa pohon dan pengurutan
    gene_tree_nw: str = '((HBB NORMAL,(
Beta talasemia IVS 1-5,Beta talasemia
IVS 1-1,Beta talasemia Codon 15, Beta
talasemia Codon 35, Beta talasemia Codon
40, Beta talasemia 41/42),(HBE Codon
26)));'

gentree = PhyloTree(gene_tree_nw)
  gentree.link_to_alignment(alg)
  return gentree, TreeStyle()
```

Gambar 2. kode yang ditulis dengan python, memanggil ete3 sebagai framework

Dalam Gambar 2, kode berkisar dari memanggil ete3 di atas sebagai kerangka kerja yang ditulis python dan kemudian mengimpor perpustakaan PhyloTree untuk memvisualisasikan pohon dan TreeStyle penyelarasan dari format Fasta mulai dari HBB normal. HBB normal sebagai patokan untuk mutasi beta-thalassemia, kemudian mengelompokkan setiap mutasi yaitu Beta thalassemia mewakili beta-thalassemia mayor dan Hb E melalui format Newick yang disediakan oleh tim pengembang ETEToolkit (2016), membuat definisi get_tree () kemudian membuat format Newick dan diberi nama gene_tree_nw dengan datatype 'string' kemudian menginisialisasi 'genetree' untuk mendeklarasikan Phylotree dengan atribut gene_tree_nw kemudian link_to_alignment dengan atribut 'alg' yang mana metode yang digunakan menggunakan *progessive alignment* (Huerta, *et al* 2010). Akhirnya, kembali untuk visualisasi ke TreeStyle. Kode ini ditulis dalam Komunitas Pycharm. Untuk menghitung waktu proses eksekusi. Kode berikut adalah cara untuk mengukur waktu eksekusi skrip dengan menggunakan modul Python bawaan waktu.

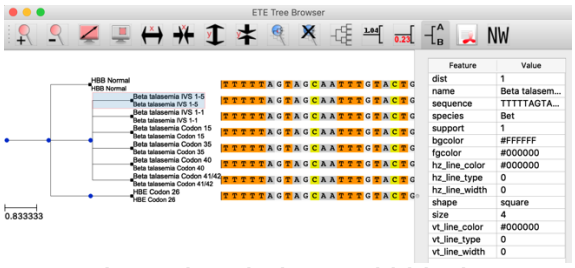
```
Import time
start = time.time()
#koding utama ditulis
ditengah-tengah start
dan end
end = time.time()
```

Gambar 3. kode untuk menampilkan waktu pelaksanaan proses pada Gambar 2.

Menjalankan kode pada Gambar 3, memanggil 'time' sebagai *library* kemudian membuat variabel 'start' untuk mendapatkan 'time()', kemudian memulai eksekusi dalam hitungan detik setelah kode utama mendapat waktu eksekusi dengan perolehan 9.5367431 /detik. Hasil itu masih merupakan respon normal dan cepat karena hanya 7 DNA telah terdaftar dalam format Fasta.

4. Hasil dan Pembahasan

Hasilnya diformat dalam gambar PDF yang dapat mengkustomisasi cabang atau label nama pada Gambar 4 di Fitur ETE.



Gambar 4 Pada tata letak utama adalah hasil Beta Thalassemia dengan 7 mutasi yang diberikan sebagai penanda dengan panjang tree0.833333 dan DNA. Menu terbalik digunakan untuk memperbesar dan memberi nilai setiap cabang. Informasi cabang dapat menyesuaikan cabang warna di sisi kanan.

Membandingkan Beta-Thalassemia dengan wilayah atau spasial mana pun lebih divisualisasikan secara mendalam dan komunitas molekuler / bioinformatika membutuhkan lebih banyak kolaborasi dengan multi-disiplin ilmu

lainnya. Selain itu, menganalisis dalam molekul adalah biaya yang sangat tinggi. Tantangan untuk masa depan adalah untuk memenuhi data besar (Stamatakis 2005). Karena mereka membutuhkan lebih banyak dukungan untuk data besar di antaranya:

4.1 Efisiensi memori

Mengurangi konsumsi memori dan meningkatkan efisiensi *cache* karena dua alasan utama: pertama, komputasi pohon yang sangat besar telah layak dengan generasi baru algoritma pencarian dan, karena akumulasi data yang sangat besar, keberpihakan juga terus meningkat di kedua dimensi: panjang penyelarasan dan jumlah taksa. Penerapan sebagian besar program saat ini untuk masalah yang lebih besar dibatasi oleh konsumsi memori. Kedua, selama beberapa tahun sekarang, kecepatan Central Processing Unit (CPU) telah meningkat pada tingkat yang lebih tinggi daripada kecepatan akses memori, sehingga kinerja aplikasi ilmiah skala besar sekarang dibatasi oleh pola akses memori mereka daripada CPU. kecepatan. Tidak mungkin bahwa tren ini akan terbalik saat ini. Beberapa strategi untuk mengatasi beban ini termasuk mengoptimalkan program untuk mengurangi konsumsi memori secara teknis, menyebarkan strategi membagi dan menaklukkan untuk mengurangi ukuran masalah dan mengeksploitasi prosesor memori bersama yang kuat (Stamatakis 2005). Akhirnya, pendekatan baru-baru ini juga mengeksploitasi potensi komputasi yang sangat besar dari perangkat keras periferal seperti GPU (Graphics Processing Units) (Nobile *et al*, 2016).

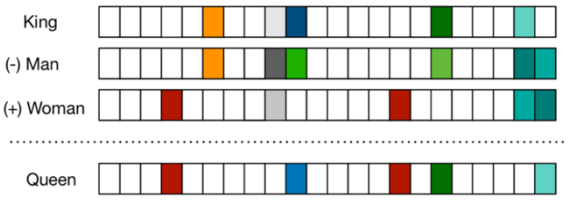
4.2 Implementasi dan optimalisasi fungsi kemungkinan(Likelihood function).

Bidang penelitian penting lainnya adalah optimalisasi fungsi kemungkinan, yang biasanya menghabiskan lebih dari 90 persen dari total waktu eksekusi dalam program-program seperti RAxML atau PHYML (Stamatakis et al. 2005). Beberapa pendekatan fokus pada mendeteksi pola yang sama dalam keberpihakan dan menggunakan kembali nilai yang dihitung sebelumnya alih-alih menghitungnya kembali setiap waktu. Selain itu, penggunaan implementasi terpisah dari fungsi intensif-komputasi untuk masing-masing model substitusi nukleotida, serta penerapan optimasi teknis tingkat rendah (mis. Loop manual membuka gulungan) ke fungsi probabilitas, akan menjadi penting untuk perhitungan pohon yang sangat besar (Kosakovsky dan Muse 2004; Stamatakis et al. 2002). Akhirnya, banyak kemajuan dalam penggunaan pembelajaran mesin seperti pembelajaran dalam adalah metode yang menggunakan beberapa lapisan pemrosesan untuk mempelajari representasi data secara hierarkis dan telah menghasilkan hasil yang canggih di banyak bidang (Binhua et al. 2019) dan Natural Language Process adalah serangkaian teknik komputasi yang dimotivasi secara teoritis untuk analisis dan representasi otomatis (Cambria E dan White B 2014). Berbagai desain model dan metode telah berkembang baru-baru ini. Ada beberapa model dan metode signifikan yang terkait dengan pembelajaran mendalam yang telah digunakan untuk berbagai tugas NLP dan memberikan langkah-langkah evolusi mereka yaitu:

4.2.1 Penyuluhan kata (*word embeddings*)

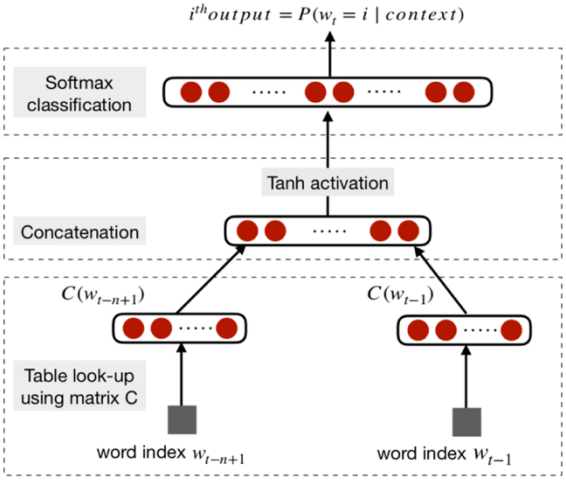
Vektor distribusi atau penyisipan kata pada Gambar 5 pada dasarnya mengikuti hipotesis distribusi bahwa kata-kata dengan makna yang sama cenderung terjadi dalam konteks yang sama. Dengan demikian, vektor-vektor ini berupaya menangkap karakteristik tetangga kata. Manfaat utama vektor distribusi adalah bahwa mereka menangkap kesamaan antara kata-kata. Dimungkinkan untuk mengukur kesamaan antara vektor menggunakan ukuran seperti kesamaan cosinus. Word embeddings sering digunakan dalam model pembelajaran yang mendalam sebagai lapisan pertama dari pemrosesan data. Biasanya, embedding kata sudah dilatih sebelumnya dengan mengoptimalkan tujuan tambahan dalam corpus besar yang tidak berlabel, seperti memprediksi kata berdasarkan konteksnya (Mikolov et al 2013a, 2013b), di mana vektor kata yang dipelajari dapat menangkap informasi sintaksis umum dan semantik. Dengan demikian, sistem embedding ini terbukti efisien dalam menangkap kesamaan kontekstual, analogi dan karena dimensinya yang lebih kecil, mereka cepat dan efisien dalam memproses tugas-tugas inti NLP. Selama bertahun-tahun, model yang membuat embedding seperti itu telah menjadi jaringan saraf yang dangkal dan ada tidak perlu jaringan yang dalam untuk membuat embeddings yang baik. Model pembelajaran NLP yang mendalam, bagaimanapun, selalu mewakili kata-kata mereka, frasa, dan bahkan kalimat menggunakan metode embedding ini. Memang, ini adalah perbedaan utama antara model berbasis jumlah kata tradisional dan model pembelajaran yang mendalam. Penyuluhan kata bertanggung jawab untuk hasil mutakhir dalam berbagai tugas NLP. (Weston et al. 2011; Socher et al. 2013; Turney dan Pantel 2010; Cambria et al. 2017). Sebagai contoh, Glorot et al (2011) menggunakan penyertaan bersama dengan autoencoder de-

noising ditumpuk untuk adaptasi domain dalam klasifikasi sentimen dan Hermann dan Blunsom (2013) menyajikan kombinasi autoencoder kategorial untuk mempelajari komposisi kalimat. Penggunaannya yang luas dalam literatur terbaru menunjukkan efektivitas dan pentingnya mereka dalam setiap model pembelajaran mendalam yang melakukan tugas NLP. Representasi terdistribusi (embedding) dipelajari terutama melalui konteks. Pada 1990-an, beberapa perkembangan penelitian (Elman 1991) menandai dasar penelitian semantik distribusi. Ringkasan yang lebih rinci dari tren awal ini dapat ditemukan di (Glenberg dan Robertson 2000; Dumais 2003). Perkembangan selanjutnya adalah adaptasi dari karya-karya awal ini, menghasilkan penciptaan model topik seperti alokasi Dirichlet laten (Blei et al. 2003) dan model bahasa (Bengio et al. 2003). Karya-karya ini meletakkan dasar untuk pembelajaran representasi dalam bahasa alami. Mengusulkan model bahasa saraf yang mempelajari representasi terdistribusi untuk kata-kata pada Gambar 6 (Bengio et al. 2003). Penulis berpendapat bahwa representasi kata ini, setelah dikompilasi menjadi representasi kalimat menggunakan probabilitas gabungan dari urutan kata, mencapai jumlah eksponensial dari kalimat semi-tetangga. Ini, pada gilirannya, membantu dalam generalisasi sebagai kalimat yang tidak terlihat sekarang bisa mendapatkan kepercayaan yang lebih besar jika urutan kata dengan kata-kata yang serupa (dalam hal representasi kata terdekat) sudah terlihat. Collobert dan Weston (2008) adalah karya pertama yang menunjukkan kegunaan dari kata embeddings pra-terlatih. Mereka mengusulkan arsitektur jaringan saraf yang membentuk fondasi untuk banyak pendekatan saat ini. Pekerjaan itu juga mengatur embedding kata sebagai alat yang berguna untuk tugas-tugas NLP. Namun, mempopulerkan besar kata embedding mungkin disebabkan oleh Mikolov et al (2013), yang mengusulkan model kata-kata (CBOW) dan loncatan-gram terus menerus untuk secara efisien membangun representasi vektor terdistribusi berkualitas tinggi. Mendorong popularitas mereka adalah efek samping yang tidak terduga dari vektor yang menunjukkan komposisionalitas, yaitu, menambahkan hasil vektor dua kata dalam vektor yang merupakan gabungan semantik dari kata-kata individu, misalnya, ‘man '+‘ royal' = ‘king '. Pembenaran teoritis untuk perilaku ini baru-baru ini diberikan oleh Gittens et al (2017), yang menyatakan bahwa komposisionalitas terlihat hanya ketika asumsi tertentu dipegang, seperti asumsi bahwa kata-kata perlu didistribusikan secara seragam di ruang embedding. Glove oleh Pennington et al (2014) adalah metode penyisipan kata terkenal lainnya yang pada dasarnya merupakan model "*count-based*".



Gambar 5. Vektor terdistribusi diwakili oleh vektor dimensi-D di mana D <<V adalah ukuran kosakata (Azzimonti dan Ginsbourger 2018).

Di sini, dengan menormalkan penghitungan dan operasi penghalusan log, kata matriks penghitungan kemunculan kata diproses lebih dulu. Matriks ini kemudian difaktorkan untuk mendapatkan representasi dimensi yang lebih rendah dengan meminimalkan "kehilangan rekonstruksi". Di bawah ini adalah deskripsi singkat dari metode word2vec yang disarankan oleh Mikolov et al. (2013).

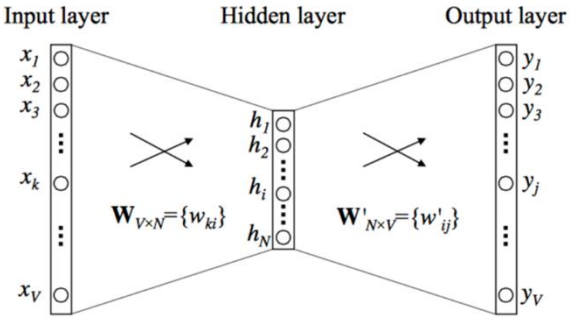


Gambar 6. *Neural Language Model* (Bengio *et al*, 2003). C (i) adalah embedding kata ke-i.

4.2.2 Word2vec

Word embeddings mengalami revolusi (Mikolov *et al* 2013a, 2013b) yang mengusulkan model CBOW dan skip-gram. CBOW menghitung probabilitas bersyarat dari kata target mengingat kata-kata konteks yang mengelilinginya pada ukuran jendela k. Model skip-gram, di sisi lain, melakukan kebalikan dari model CBOW dengan memprediksi kata-kata konteks sekitarnya yang diberikan kata target pusat. Kata-kata konteks diasumsikan berada secara simetris pada kedua arah dalam jarak yang sama dengan ukuran jendela dengan kata-kata target. Dalam pengaturan tanpa pengawasan, dimensi penyematan kata ditentukan oleh akurasi prediksi. Dimensi penyisipan kata ditentukan oleh keakuratan prediksi dalam pengaturan tanpa

pengawasan. Saat dimensi penyematan meningkat, akurasi prediksi juga meningkat hingga konvergen pada titik tertentu, yang dianggap sebagai dimensi penyematan optimal karena merupakan dimensi tersingkat tanpa mengurangi akurasi. Mari kita pertimbangkan versi sederhana dari model CBOW di mana hanya satu kata yang dipertimbangkan dalam konteks. Ini pada dasarnya mereplikasi model bahasa bigram. Seperti yang ditunjukkan pada Gambar 7, model CBOW adalah jaringan saraf sederhana, yang terhubung sepenuhnya dengan satu lapisan tersembunyi. Lapisan input yang mengambil vektor konteks kata satu-panas memiliki neuron V sedangkan lapisan tersembunyi memiliki neuron N. Lapisan output adalah probabilitas softmax atas semua kata dalam kosakata. Masing-masing layer dihubungkan dengan *weight matrix* $W \in R^{V \times N}$ dan $W' \in R^{HxV}$



, Setiap kata dari kosakata akhirnya direpresentasikan sebagai dua vektor yang dipelajari VC dan $_w$VW, masing-masing sesuai dengan representasi konteks dan kata target.

Dengan demikian, kata kth dalam kosakata akan memiliki.

$$V_c = W_{(k,.)} \text{ and } V_w = W_{(k,.)} \quad (1)$$

Secara keseluruhan, untuk setiap kata w_i dengan kata konteks yang diberikan c sebagai input,

$$p\left(\frac{w_i}{c}\right) = y_i = \frac{e^{u_i}}{\sum_i^V e^{u_i}} where, u_i = v_{w_i}^T . \quad _{VC} \quad (2)$$

Parameter $\theta = \{ v_w, v_c \}_{w,c \in Vocab}$ dipelajari dengan mendefinisikan fungsi tujuan sebagai kemungkinan log dan menemukan gradien sebagai

$$1(\theta) = \sum_{w \in Vocab} \log\left(p\left(\frac{w}{c}\right)\right) \quad (3)$$

$$\frac{\partial 1(\theta)}{\partial v_w} = V_c(1 - p\left(\frac{w}{c}\right)) \quad (4)$$

Semua bidikan kata konteks diambil secara bersamaan sebagai input dalam model CBOW umum, yaitu,

$$h = W^T(= X_1 + X_2 + \cdots + X_c) \quad (5)$$

Salah satu batasan penyisipan kata individu adalah ketidakmampuan mereka untuk mewakili frasa (Mikolov *et al*. 2013), di mana kombinasi dua atau lebih kata — misalnya, idiom seperti "kentang panas" atau entitas bernama seperti "Boston Globe" —tidak mewakili kombinasi makna kata-kata individual. Salah satu solusi untuk masalah ini, seperti yang dieksplorasi oleh Mikolov *et al* (2013), adalah untuk secara terpisah mengidentifikasi frasa tersebut berdasarkan kata co-kejadian dan melatih embedding untuk mereka. Metode selanjutnya telah mengeksplorasi pembelajaran langsung penyematan n-gram dari data yang tidak berlabel (Johnson dan Zhang 2015). Keterbatasan lain berasal dari pembelajaran embedding hanya berdasarkan jendela kecil kata-kata sekitarnya, kadang-kadang kata-kata seperti berbagi baik dan buruk hampir embedding yang sama (Socsher *et al*. 2011), yang bermasalah ketika digunakan dalam tugas-tugas seperti analisis sentimental (Wang *et al*, 2015). Terkadang kata-kata yang tertanam ini mengelompok kata-kata yang mirip secara semantik yang memiliki polaritas perasaan yang berlawanan. Ini menghasilkan model hilir yang digunakan untuk tugas analisis sentimen karena tidak dapat mengidentifikasi perbedaan polaritas yang mengarah pada kinerja yang buruk. Tang *et al* (2014) membahas masalah ini dengan mengusulkan penyisipan kata spesifik sentimen (SSWE). Penulis memasukkan polaritas teks dari sentimen yang diawasi ke dalam fungsi yang hilang saat mempelajari embeddings. Peringatan umum untuk embedding kata adalah bahwa itu sangat tergantung pada aplikasi yang digunakan. Labutov dan Lipson (2013) telah mengusulkan embedding tugas khusus yang melatih kembali kata embedding untuk menyelaraskannya di

ruang tugas saat ini. Ini sangat penting karena pelatihan penyisipan dari awal membutuhkan banyak waktu dan sumber daya. Mikolov *et al* (2013) mencoba untuk mengatasi masalah ini dengan mengusulkan pengambilan sampel negatif yang melakukan pengambilan sampel berbasis frekuensi dari istilah negatif sambil melatih model word2vec. Algoritma penyematan kata tradisional memberikan vektor berbeda pada setiap kata. Ini membuat mereka tidak mungkin memperhitungkan polisemi. Upadhyay *et al* (2017) menyediakan cara inovatif untuk mengatasi defisit ini dalam pekerjaan terbaru. Menggunakan data paralel multibahasa untuk mempelajari penyisipan kata multi-akal. Misalnya, kata bank Inggris, ketika diterjemahkan ke dalam bahasa Prancis menyediakan dua kata yang berbeda: banc dan banque yang mewakili arti finansial dan geografis. Informasi distribusi multibahasa seperti itu telah membantu mereka menjelaskan polisemi. Tabel 1 menyediakan direktori kerangka kerja yang ada yang sering digunakan untuk membuat *embedding* yang selanjutnya dimasukkan ke dalam model pembelajaran yang mendalam.

Tabel 1. *Framework* menyediakan alat dan metode penyisipan kata

Framework	Bahasa	URL
Gensim	Python	https://radimrehurek.com/gensim
Pydsm	Python	https://github.com/jimmycallin/pydsm
Dissect	Python	https://clic.cimec.unitn.it/composes/toolkit
FastText	Python	https://fasttext.cc/
Elmo	Python	https://tfhub.dev/google/elmo/2
Semantic vectors	Java	https://github.com/semanticvectors/
S-Space	Java	https://github.com/fozziethebeat/S-Space

4.2.3 *Character Embeddings*

Penyisipan (*embedding*) kata dapat menangkap informasi sintaksis dan semantik, tetapi informasi morfologis dan pembentukan dalam kata juga dapat sangat berguna untuk tugas-tugas seperti penandaan POS dan NER. Secara umum, pembangunan sistem pemahaman bahasa alami tingkat karakter telah menarik perhatian penelitian (Kim et al. 2016; Santos dan Gatti 2014; Santos dan Guimaraes 2015; Santos dan Zadrozny 2014). Hasil yang lebih baik dilaporkan pada bahasa yang kaya secara morfologis dalam tugas NLP tertentu. Santos dan Guimaraes (2015) menerapkan representasi tingkat karakter, bersama dengan kata embedding untuk NER, mencapai hasil canggih di perusahaan Portugis dan Spanyol. Kim et al (2016) menunjukkan hasil positif dalam membangun model bahasa saraf dengan hanya menggunakan karakter tertanam. Ma et al (2016) menggunakan beberapa embeddings, termasuk trigram karakter, untuk menggabungkan informasi prototipikal dan hirarkis untuk mempelajari label embedding yang sudah dilatih dalam konteks NER. Masalah kata yang tidak diketahui, juga dikenal sebagai out-of-vocabulary (OOV), adalah fenomena umum untuk bahasa dengan kosa kata yang besar. Secara alami, karakter embedding berurusan dengan itu karena setiap kata dianggap tidak lebih dari komposisi huruf individual. Dalam bahasa di mana teks tidak terdiri dari kata-kata yang terpisah tetapi dari karakter individu dan makna semantik dari kata-kata memetakan ke karakter komposisi mereka (seperti Cina), membangun sistem pada tingkat karakter adalah pilihan alami untuk menghindari segmentasi kata (Ma et al . 2016). Dengan demikian, bekerja menggunakan aplikasi pembelajaran dalam pada bahasa tersebut cenderung lebih suka menanamkan karakter daripada vektor kata (Zheng et al. 2013). Sebagai contoh, Peng et al (2017) menunjukkan bahwa pemrosesan tingkat radikal dapat sangat meningkatkan kinerja perasaan klasifikasi. Secara khusus, ia mengusulkan dua jenis penanaman hierarkis berbasis radikal Tiongkok, yang menggabungkan tidak hanya semantik radikal dan tingkat karakter, tetapi juga informasi tentang sentimen. Bojanowski et al (2016) juga berusaha untuk meningkatkan representasi kata-kata dengan menggunakan informasi tingkat karakter dalam bahasa yang kaya secara morfologis. Mereka mendekati metode lompatan-gram dengan merepresentasikan kata-kata sebagai n-gram kantong karakter. Karenanya, pekerjaan mereka memiliki keefektifan model skip-gram bersamaan dengan mengatasi beberapa masalah penyematan kata yang persisten. Metode ini juga cepat, yang dengan cepat memungkinkan model pelatihan korpora besar. Dikenal sebagai FastText, metode seperti ini menonjol dalam hal kecepatan, skalabilitas, dan efektivitas dibandingkan metode sebelumnya. Selain menanamkan karakter, berbagai pendekatan untuk penanganan OOV

telah diusulkan. Herbelot dan Baroni (2017) menyediakan on-the-fly OOV handling dengan menginisialisasi kata-kata yang tidak diketahui sebagai jumlah dari kata konteks dan menyempurnakan kata-kata tersebut dengan tingkat pembelajaran yang tinggi. Namun, pendekatan mereka masih harus diuji pada tugas-tugas NLP yang khas. Pinter et al (2017) memberikan pendekatan yang menarik untuk pelatihan model berbasis karakter untuk menciptakan kembali penanaman pra-terlatih. Ini memungkinkan mereka untuk belajar dari karakter ke kata menanamkan pemetaan komposisi, sehingga mengatasi masalah OOV. Terlepas dari popularitas vektor distribusi yang semakin meningkat, diskusi baru-baru ini tentang relevansi jangka panjangnya bermunculan. Misalnya, Lucy dan Gauthier (2017) baru-baru ini mencoba untuk mengevaluasi seberapa baik kata vektor menangkap aspek yang diperlukan dari makna konseptual. Para penulis telah menemukan keterbatasan yang parah dalam pemahaman persepsi konsep di balik kata-kata, yang tidak dapat disimpulkan dari semantik distribusi saja. Arahan yang mungkin untuk mengurangi kekurangan ini akan didasarkan pada pembelajaran, yang telah mendapatkan popularitas dalam domain penelitian ini.

4.2.4 Contextualized Word Embedding

Kualitas representasi kata umumnya diukur dari kemampuannya untuk menyandikan informasi sintaksis dan mengelola perilaku polisemik (atau indra kata). Properti ini mengarah pada peningkatan representasi kata semantik. Pendekatan terbaru dalam bidang ini menyandikan informasi tersebut ke dalam penyematannya dengan memanfaatkan konteksnya. Metode-metode ini menyediakan jaringan yang lebih dalam untuk menghitung representasi kata sebagai fungsi dari konteksnya. Metode penyematan kata tradisional seperti Word2Vec dan Glove mempertimbangkan semua kalimat yang mengandung kata untuk membuat representasi vektor global dari kata itu. Namun, sebuah kata dapat memiliki arti atau makna yang sama sekali berbeda dalam konteksnya. Misalnya, mari kita perhatikan dua kalimat ini - 1) "Bank tidak akan menerima uang tunai pada hari Sabtu" 2) "Sungai meluap bank.". Kata indra bank berbeda dalam dua kalimat ini tergantung pada konteksnya. Secara masuk akal, orang mungkin menginginkan dua representasi vektor yang berbeda dari bank kata berdasarkan dua arti kata yang berbeda. Alasan ini diadopsi oleh kelas model baru dengan menyimpang dari konsep representasi kata global dan bukannya mengusulkan embedding kata kontekstual. Penyematan dari Model Bahasa (ELMo) (Peters et al. 2018) adalah salah satu metode yang menyediakan penyematan kontekstual yang mendalam. ELMo menghasilkan penyisipan kata untuk setiap konteks di mana kata tersebut digunakan, memungkinkan representasi yang berbeda untuk indera yang berbeda dari kata yang sama. Khususnya, untuk N kalimat berbeda di mana kata w hadir, ELMo menghasilkan N representasi berbeda dari w yaitu, w1, w2, ... ,wN. Mekanisme ELMo didasarkan pada representasi yang diperoleh dari model bahasa dua arah. Model bahasa dua arah (biLM) terdiri dari dua model bahasa (LM) 1) *forward* LM dan 2) *backward* LM. LM yang maju membutuhkan representasi input x_k^{LM} untuk tiap dari kth token dan meneruskannya melalui lapisan L penerusan LSTM untuk mendapatkan representasi $\underset{h}{\rightarrow}_{k,j}^{LM}$ dimana j = 1 ,...,L. Masing-masing representasi ini, menjadi representasi tersembunyi dari jaringan saraf berulang, tergantung konteks. LM ke depan dapat dilihat sebagai metode untuk memodelkan probabilitas gabungan dari urutan token: $p\,(t_1, t_2, \dots, t_N) = \prod_{k=1}^{N} p\,(t_1, t_2, \dots, t_{k-1})$. Pada langkah waktu k – 1 forward LM memprediksi token ^{t}k berikutnya mengingat token yang diamati sebelumnya $t_1, t_2, \dots, t_k$ Ini biasanya dicapai dengan menempatkan lapisan *softmax* di atas LSTM akhir dalam LM maju. Di sisi lain, model LM mundur probabilitas gabungan yang sama dari urutan dengan memprediksi token sebelumnya mengingat token masa depan: $p\,(t_1, t_2, \dots, t_N) = \prod_{k=1}^{N} p\,(t_1, t_2, \dots, t_{k-1})$. Dengan kata lain, LM terbalik mirip dengan LM penerusan yang memproses urutan dengan urutan dibalik. Pelatihan model biLM melibatkan pemodelan log-kemungkinan kedua orientasi kalimat. Terakhir, representasi tersembunyi dari kedua LM digabungkan untuk menyusun vektor token terakhir (Mousa dan Schuller 2017).

$$R_k = \{x_k^{LM}, \underset{h}{\rightarrow}_{k,j}^{LM}, \overleftarrow{h}_{k,j}^{LM} \mid j = 1, \dots, L\} \quad (6)$$

$$= \{h_{k,j}^{LM} | j = 0, \dots, L$$

Disini, $h_{k,0}^{LM}$ adalah representasi token di tingkat terendah. Seseorang dapat menggunakan karakter atau embeddings kata untuk menginisialisasi $h_{k,0}^{LM}$. Untuk nilai lain dari j,

$$h_{k,j}^{LM} = [\overrightarrow{h}_{k,j}^{LM}, \overleftarrow{h}_{k,j}^{LM}] \quad \forall j = 1, \ldots, L. \tag{7}$$

ELMo meratakan semua lapisan vektor tunggal dalam R, sehingga,

$$\boldsymbol{ELMo}_k^{task} = E(R_k; \theta^{task}) \tag{8}$$

$$= \gamma^{task} \sum_{j=0}^{L} s_j^{task} \boldsymbol{h}_{k,j}^{LM}$$

Dalam Persamaan 8, s_j^{task}adalah vektor bobot *softmax* dinormalisasi untuk menggabungkan representasi dari berbagai lapisan. γ^{task} adalah hyperparameter yang membantu dalam pengoptimalan dan tugas penskalaan spesifik representasi ELMo. ELMo menghasilkan representasi kata yang bervariasi dalam kalimat yang berbeda untuk kata yang sama. Menurut Peters et al (2018), menggabungkan representasi kata ELMo dengan representasi kata global standar seperti Glove dan Word2Vec selalu bermanfaat. Terlambat, ada lonjakan minat dalam model bahasa pra-dilatih untuk berbagai tugas bahasa alami (Dai dan Le 2015). Pemodelan bahasa dipilih sebagai tujuan dari pra-pelatihan karena secara luas dianggap menggabungkan banyak sifat pemahaman dan generasi bahasa alami. Model bahasa yang baik membutuhkan pembelajaran karakteristik bahasa yang kompleks yang melibatkan sifat sintaksis dan juga koherensi semantik. Oleh karena itu, diyakini bahwa pelatihan tanpa pengawasan pada tujuan tersebut akan menanamkan pengetahuan linguistik yang lebih baik ke dalam jaringan daripada inisialisasi acak. Juga diinginkan adalah pra-pelatihan generatif dan prosedur fine-tuning diskriminatif karena pra-pelatihan tidak diawasi dan tidak memerlukan label manual. Radford et al (2018), dengan mengadaptasi Transformer, mengusulkan model pra-terlatih serupa, OpenAI-GPT. Devlin et al (2018) baru-baru ini mengusulkan BERT untuk menggunakan jaringan transformator untuk pra-melatih model bahasa untuk mengekstraksi penyisipan kata kontekstual. Tidak seperti ELMo dan OpenAI-GPT, untuk pemodelan bahasa, BERT menggunakan tugas pra-pelatihan yang berbeda. BERT secara acak menutupi persentase kata dalam kalimat di salah satu tugas dan hanya memprediksi kata-kata yang di-mask itu. Dalam tugas lain, BERT memprediksi kalimat berikutnya yang diberi kalimat. Secara khusus, tugas ini mencoba memodelkan hubungan antara dua kalimat yang seharusnya tidak ditangkap oleh model bahasa dua arah tradisional. Akibatnya, skema pra-pelatihan khusus ini membantu BERT untuk mengungguli teknik-teknik canggih dalam tugas-tugas utama NLP seperti QA, Natural Language Inference (NLI), di mana pemahaman antara dua kalimat sangat penting. Pendekatan yang diuraikan untuk menanamkan kata-kata kontekstual menjanjikan representasi kualitas yang lebih baik untuk kata-kata. Dalam bentuk pembelajaran transfer, model-model bahasa dalam pra-terlatih juga memberikan awal untuk tugas-tugas hilir. Pendekatan ini sangat populer dalam tugas penglihatan komputer. Apakah tren serupa akan terjadi di komunitas NLP, di mana para peneliti dan praktisi akan lebih suka model seperti itu daripada varian tradisional, masih harus dilihat di masa depan.

4.3 Information Visualization

Merancang alat pengamat pohon novel sangat penting untuk mempercepat proses analisis serta mengekstraksi informasi yang berguna dari data dan mempercepat proses kognitif. Gambar-gambar filamen, radial, dan miring adalah di antara representasi yang paling populer (Munzner et al. 2003). Program pengamatan pohon umum seperti ATV (Zmasek dan Eddy 2001) menyediakan representasi seperti itu. Akan tetapi, layout dan program ini ditargetkan pada pohon berukuran sedang dengan maksimum 300-400 taksa. Jadi, dengan ribuan taksa, mereka tidak cocok untuk memvisualisasikan pohon besar. Pendekatan untuk pohon yang lebih besar menggunakan ruang hiperbolik dua dimensi dan tiga dimensi (Hughes et al, 2004) untuk secara bersamaan memberikan pandangan rinci, serta kontekstual, dari pohon tersebut. Pendekatan lain, seperti SpaceTree (Plaisant et al. 2002). Ini hanya menampilkan bagian representatif dari pohon yang sangat besar. Namun, para ahli biologi biasanya lebih menyukai tampilan simultan yang terperinci dan pandangan filogeni yang

kontekstual. Baru-baru ini telah diusulkan untuk menggunakan treemaps untuk menampilkan pohon filogenetik (Arvelakis et al. 2005), tetapi konsep ini juga terbatas maksimum 2.000-3.000 taksa. Ada juga pendekatan yang didasarkan pada realitas virtual (Stolk et al. 2002) yang, bagaimanapun, tidak dapat diakses oleh sebagian besar peneliti karena besarnya biaya infrastruktur masing-masing. (Carrizo 2004) memberikan tinjauan yang dapat dibaca dan komprehensif tentang upaya untuk menampilkan pohon filogenetik secara memadai dari perspektif visualisasi informasi. Namun demikian, karena saat ini tidak ada solusi yang benar-benar memuaskan, desain alat visualisasi yang tepat menjadi masalah yang semakin penting, karena jika tidak, informasi yang terkandung dalam filogeni besar tidak akan berguna.

5 Kesimpulan dan saran

ETE memenuhi kebutuhan analisis skala besar struktur data hierarki pohon. Itu dirancang sebagai toolkit yang sangat luas dan dapat diprogram. ETEToolkit Tree Browser memungkinkan kustomisasi pohon, rendering gambar PDF, zooming, dan detail informasi. Visualisasi Beta-Thalassemia membuatnya menarik dan mudah diketahui oleh setiap peneliti dalam bioinformatika atau konsultasi genetika untuk mempelajari lebih lanjut tentang biologi molekuler untuk membuat keputusan yang baik untuk mencegah Beta-Thalassemia, khususnya di Jawa Tengah, Indonesia.
Saran dalam waktu dekat, analitik visual dapat lebih dioptimalkan dengan *Natural Language Processing* berdasarkan *Deep Learning* dan dengan GPU (Unit Pemrosesan Grafis).

Daftar Pustaka:

[1] Arvelakis, A., Reczko, M., Stamatakis, A., Symeonidis, A., Tollis, I.G., 2005, *Using treemaps to visualize phylogenetic trees*, Proc of ISMBDA2005. 283-293. doi:10.1007/11573067_2

[2] Azzimonti, D., Ginsbourger, D., 2018, *Estimating orthant probabilities of high dimensional Gaussian vectors with an application to set estimation*, J of Comp and Graph Stat. 27(2):255-267. doi: 10.1080/10618600.2017.1360781

[3] Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C., 2003, *A neural probabilistic language model*. Innovations in machine learning. 194:137–186. doi:10.1.1.133.9693

[4] Binhua T, Zixiang P, Kang Y, Asif. 2019. Recent Advances of Deep learning in Bioinformatics and Computational Biology, Front Gen, 10:214. doi:10.3389/fgene.2019.00214

[5] Bojanowski P, Grave E, Joulin A, Mikolov T. 2017. Enriching word vectors with subword information, arXiv preprint. arXiv:1607.04606

[6] Blei, D.M., Ng, A.Y., Jordan, M.I., 2003, *Latent dirichlet allocation*, J of machine Learning res. 3:993–1022. doi:10.1.1.110.4050

[7] Cavalli, S.L.L., 1997. *Genes Peoples and Languages*. Proc Nat Aca Sci. 94(15): 7719 – 7724, doi:10.1046/j.1365-2540.2001.0962c.x

[8] Carrizo, S.F., 2004, *Phylogenetic trees: an information visualisation perspective*, Proc of Asia-Pacific Bioinformatics Conf (APBC2004). 29:315-320. acmid: 976563

[9] Cambria, E., White, B., 2014, *Jumping NLP curves: A review of natural language processing research*, IEEE Comp Intelligence Mag. 9(2): 48–57. doi:10.1109/MCI.2014.2307227

[10] Cambria, E., Poria, S., Gelbukh, A., Thelwall, M., 2017, *Sentiment analysis is a big suitcase*, IEEE Intelligent Systems, 32(6): 74–80, doi:10.1109/MIS.2017.4531228

[11] Collobert, R., Weston, J., 2008, A unified architecture for natural language processing: Deep neural networks with multitask learning, Proc of the 25th int conf on Machine learning, 160–167, doi.:10.1145/1390156.1390177

[12] Chen, X., Xu, L., Liu, Z., Sun, M., Luan, H., 2015, Joint learning of character and word embeddings, Proc of the 24th Int Joint Conf on Artificial Intelligence, 1236-1242. acmid:2832421

[13] Dai, A.M., Le, Q.V., 2015, Semi-supervised sequence learning. Adv in neural inf proc sys. Proc of the 28th Int Conf on Neural Inf Proce Sys. 2:3079–3087. acmid:2969583

[14] Devlin, J., Chang, M.W., Lee, K., Toutanova, K.,2018, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint, arXiv:1810.04805

[15] Santos, C.N., Gatti, M., 2014, Deep convolutional neural networks for sentiment analysis of short texts. Proc of

COLING the 25th Int Conf on Comp Ling, 69–78
[16] Dumais, S.T., 2004, Latent semantic analysis, Ann rev of inf sci and tech, 38(1):188-230, doi:10.1002/aris.1440380105
[17] Encarnacao, L.,2017, Information Visualization, IEEE Comp Graph and App, 37(2):6-7, doi:10.1109/MCG.2017.25
[18] ETEToolkit Developer Team ., 7 April 2019, ETEToolkit Documentation, http://etetoolkit.org/docs/latest/tutorial/index.html
[19] Elman, J.L., 1991, Distributed representations, simple recurrent networks, and grammatical structure, Machine learning, 7(2,3):195–225. doi:10.1007/BF00114844
[20] Galanello, R., Origa, R.. 2010, Beta-thalassemia, Orp J of Rare Dis, 5:11, doi:10.1186/1750-1172-5-11
[21] Giordano, P., Harteveld, C., Bakker, E., 2014, Genetic epidemiology and preventive healthcare in multiethnic societies: The hemoglobinopathies, Int J Env Res Pub Health, 11(6):6136–6146
[22] Gittens, A., Achlioptas, D., Mahoney, M.W., 2017, Skip-gram-zipf+ uniform= vector additivity, the 55th A M of the Association for Comp Ling, 1:69–76, doi:10.3390/ijerph110606136
[23] Glorot, X., Bordes, A., Bengio, Y.. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach, the 28th int conf on machine learning (ICML-11), 513–520, acmid:3104547
[24] Glenberg, A.M., Robertson, D.A., 2000, Symbol grounding and meaning: A comparison of high-dimensional and embodied theories of meaning, J of mem and lang, 43(3):379–401, doi:10.1006/jmla.2000.2714
[25] Hernanda, P.Y., Tursilowati, L., Arkesteijn, S.G., 2012, Towards a prevention program for b-thalassemia: The molecular spectrum in East Java, Indonesia, Hemoglobin, 36(1):1–6, doi: 10.3109/03630269.2011.642914
[26] Herman, K.M., Blunsom, P., 2013, The role of syntax in vector space models of compositional semantics, Proc of the 51th A M of the Association for Comp Ling, Association for Computational Linguistics. 894-904
[27] Herbelot, A., Baroni, M., 2017, High-risk learning: acquiring new word vectors from tiny data, arXiv preprint, arXiv:1707.06556
[28] Huerta, C.J., Dopazo, J., Gabaldon, T., 2010, ETE: a python Environment for Tree Exploration, BMC Bioinformatics 11:24, doi:10.1186/1471-2105-11-24
[29] Huerta, C.J., Dopazo, H., Dopazo, J., Gabaldón, T., 2007., The human phylome. Genome Biologyl. 8(6):R109.
[30] Huerta, C.J., Bueno, A., Dopazo, J., Gabaldón, T., 2008, PhylomeDB: a database for genome-wide collections of gene phylogenies. Nuc Acids Res, 36:491-496, doi:10.1186/gb-2007-8-6-r109
[31] Huerta, C.J., Bork, P., Serra, F., 2016, ETE 3: Reconstruction, Analysis, and Visualization of Phylogenomic Data, Mol Bio Eco, 33(6):1635-1638, doi:0.1093/molbev/msw046
[32] Hughes, T., Hyun, Y., Liberles, D.A., 2004, Visualizing very large phylogenetic trees in three dimensional hyperbolic space, BMC Bioinformatics. 5:48, doi:10.1186/1471-2105-5-48
[33] Johnson, R., Zhang, T., 2015, Semi-supervised convolutional neural networks for text categorization via region embedding. neural information processing systems, 919–927, arxiv:1504.01255
[34] Kosakovsky, P.S.L., Muse, S.V., 2004, Column sorting: rapid calculation of the phylogenetic likelihood function, Syst Biol 53(5): 685-692, doi:10.1080/10635150490522269
[35] Kim, Y., Jernite, Y., Sontag, D., Rush, A.M., 2016, Character-aware neural language models, AAAI, 2741–2749, arxiv:1508.06615
[36] Lantip R, Basalamah M, Mulatsih S, Sofro ASM. 2015. Molecular Scanning of β-Thalassemia in the Southern Region of Central Java, Indonesia; Towards a Local Prevention Program. Int J for hemoglobin res. 39(5):330-333. doi:10.3109/03630269.2015.1065420.