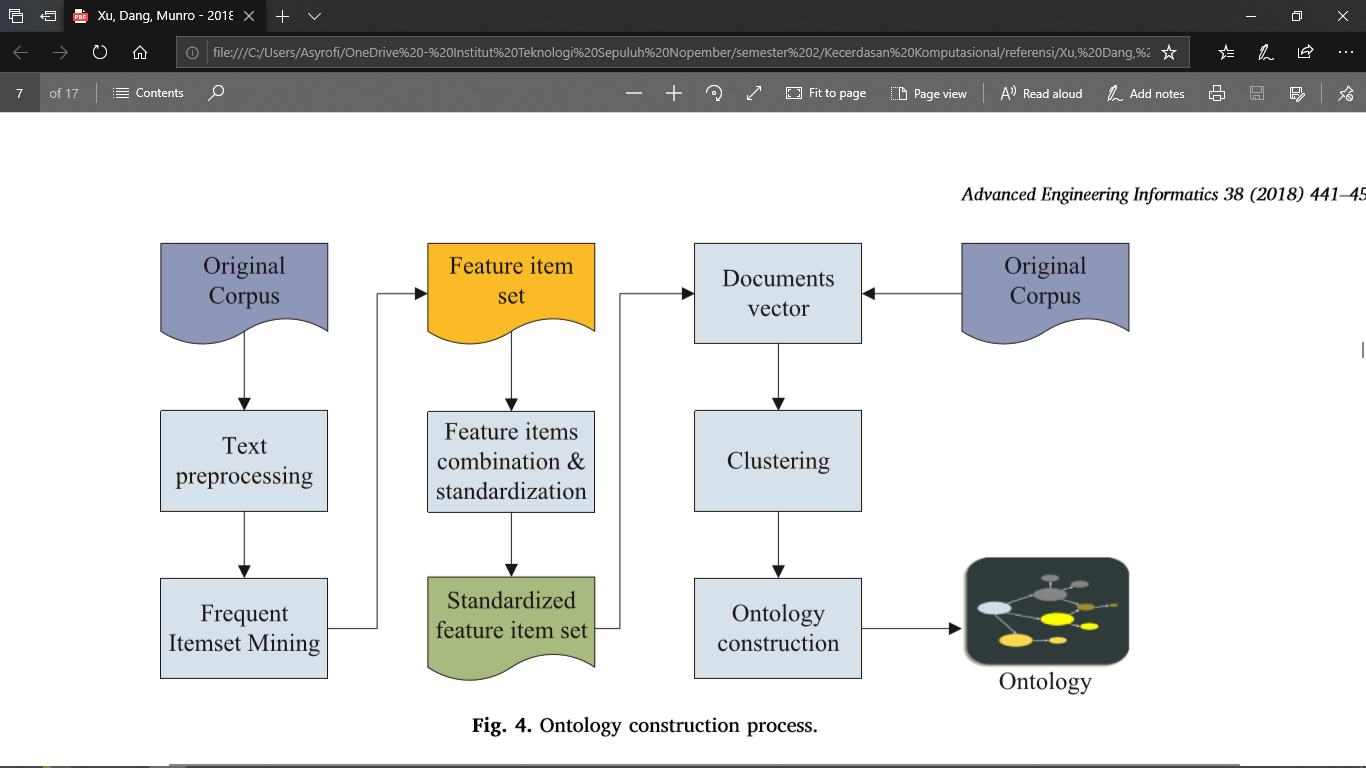
**LAPORAN TUGAS**

Nama : Rakha Asyrofi / Taufik Hidayat Mata Kuliah : Kecerdasan Komputasional

NRP : 05111950010038 / 05111950015004 Dosen : Dr. Eng. Chastine Fatichah, S.kom, M.Kom.

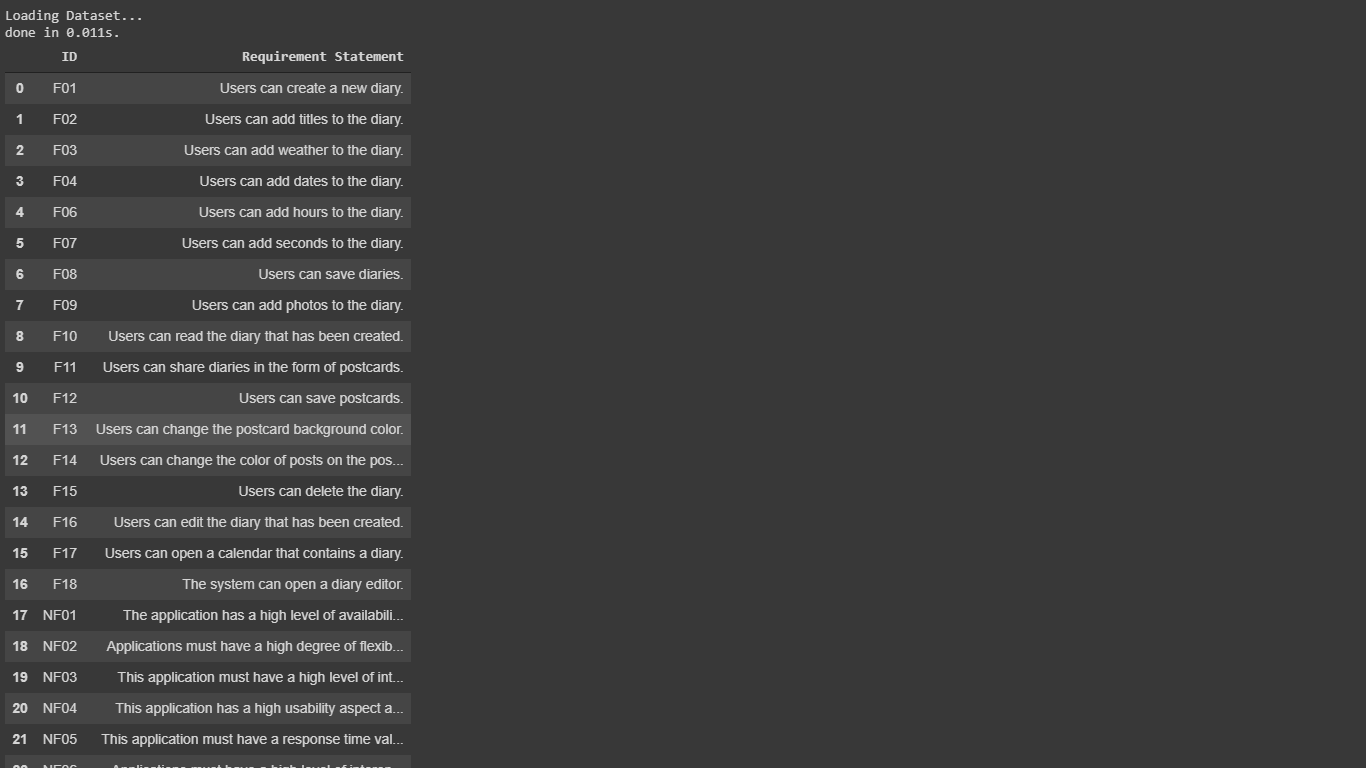


Gambar 1. Ontology Construction

Pertama, praproses corpus yang asli dan kemudian lakukan bagian dari penandaan ucapan. Selanjutnya, gunakan metode TF-IDF untuk menambang item fitur dalam teks dan melakukan pemangkasan buatan untuk menghasilkan set item fitur. Oleh karena itu, berdasarkan pendapat pakar domain, gabungkan dan standarisasi item fitur yang sama dalam set item fitur. Berdasarkan set item fitur standar, model ruang vektor dari dokumen asli dibuat. Kemudian gunakan algoritma pengelompokan k-means untuk mengelompokkan teks asli beberapa kali. Menurut metode tujuh langkah, makalah ini membangun ontologi masalah kualitas berdasarkan hasil pengelompokan.

1. Original Corpus

Corpus yang dibuat ini berdasarkan dari dataset dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) pada Gambar 2.. DImana dari hasil pengambilan corpus tersebut dimuat dalam bentuk Comma Separated Value (CSV) untuk diambil kolom fitur Statement yang berupa Kebutuhan fungsional dan kebutuhan non fungsional. Copus ini berguna untuk salah satu proses yang dilakukan untuk mendapatkan konstruksi ontologi.



Gambar 2. Original Copus

1. Text preprocessing

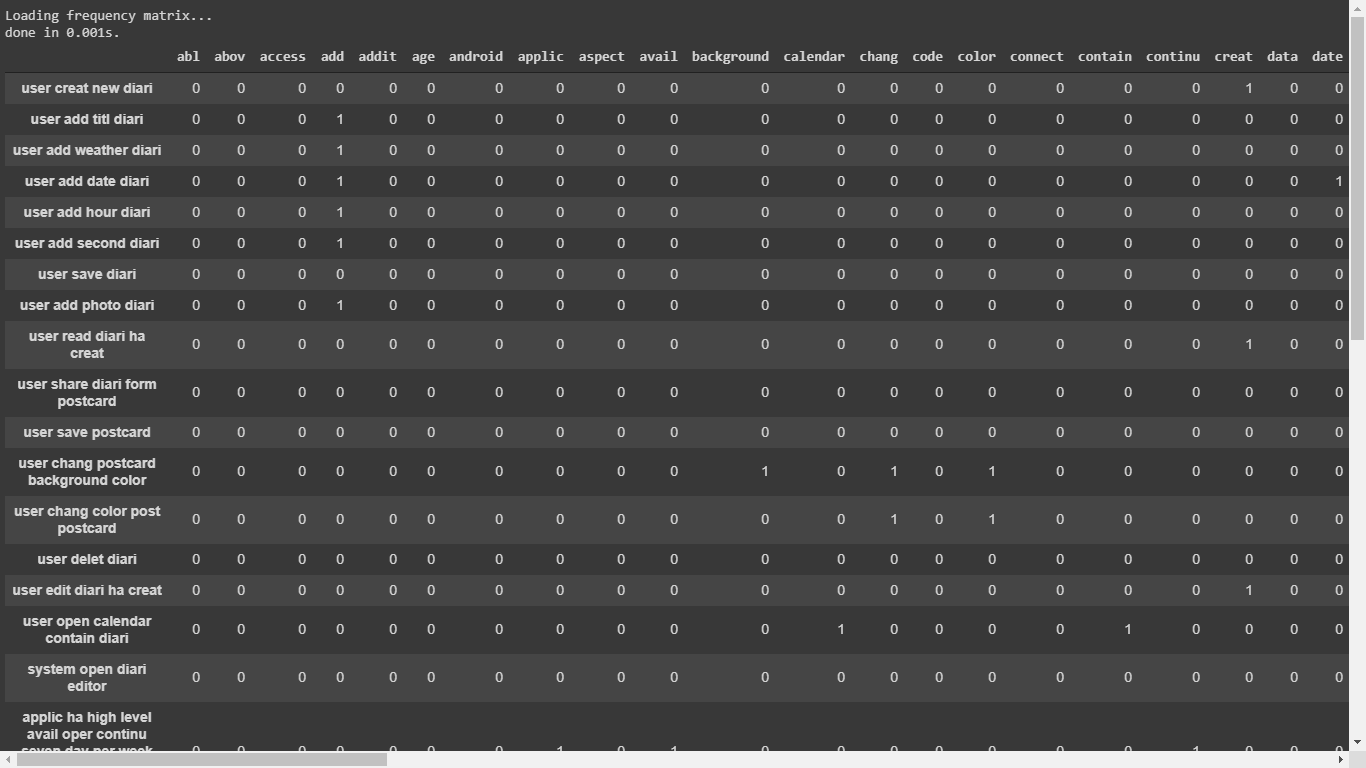
Pada corpus yang berupa kumpulan kalimat kebutuhan dari dokumen spesifikasi kebutuhan perangkat lunak (SKPL) dilakukan text preprocessing pada Gambar 3 seperti text lowering untuk mengubah kalimat ke dalam bentuk lowercase, punctuation removal untuk menghilangkan tanda baca, tokenisasi untuk memperoleh tiap tiap kata dari suatu kalimat, stopword removal untuk menghilangkan kata kata yang berupa stopword dalam bahasa inggris, stemming untuk merubah kata ke dalam bentuk kata dasar. Sehingga dari proses text preprocessing didapatlah kata kata yang benar benar bersih dari noise.



Gambar 3. Hasil Text Preprocessing

1. Frequent Itemset Mining

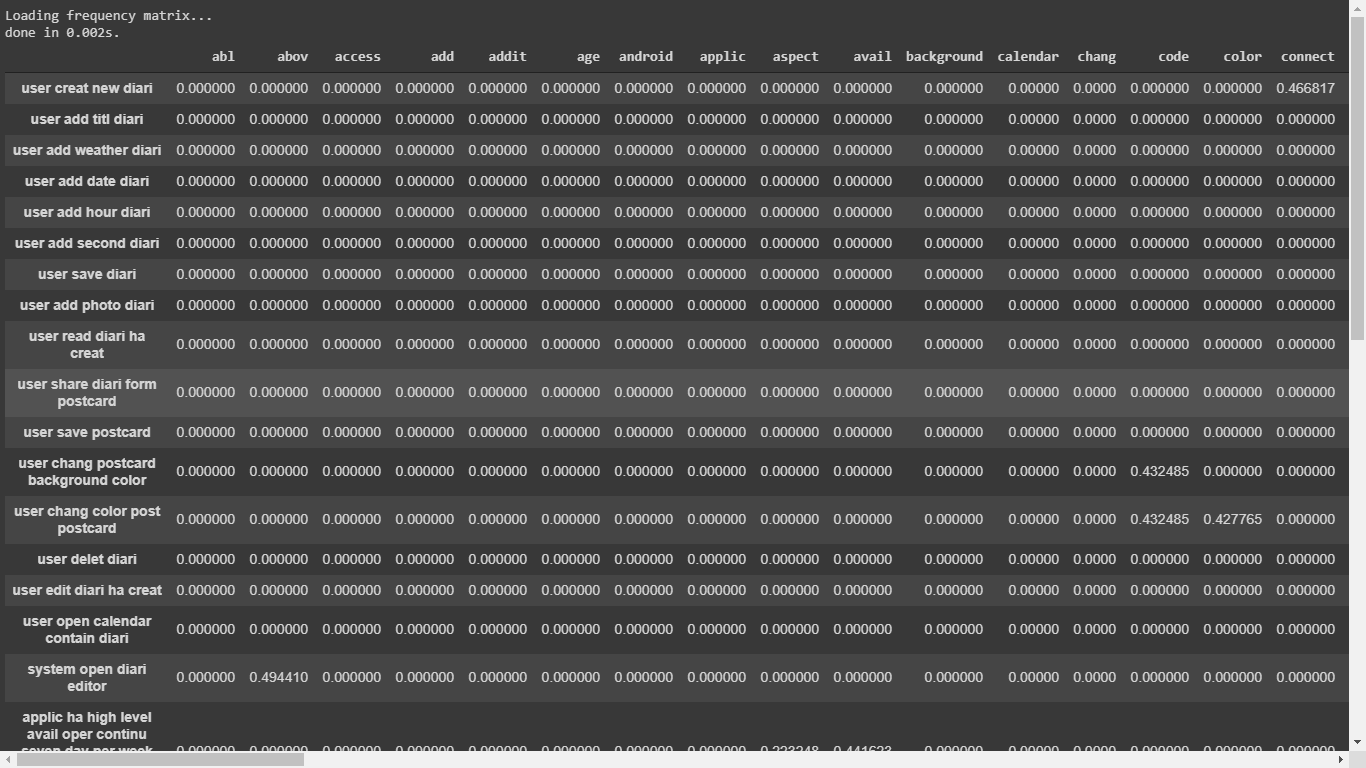
Selanjutnya dalam Gambar 4. yang menyebutkan seputar Frequent Itemset Mining dimana, kita mengambil nilai frekuensi yang muncul dari dokumen tersebut. dengan proses metode Bag of Words (BOW). Dimana dalam proses tersebut kita membuat fungsi vektorisasinya. lalu dilanjutkan dengan fitur nama mana saja yang terkandung dalam corpus tersebut. Kemudian kita transformasikan dalam bentuk array untuk melihat sebuah angka biner antara 0 dan 1 dalam sebuah korpus tersebut. Lalu membuatnya menjadi sebuah matrix frequency.



Gambar 4. Frequent Itemset Mining

1. Feature Item Set

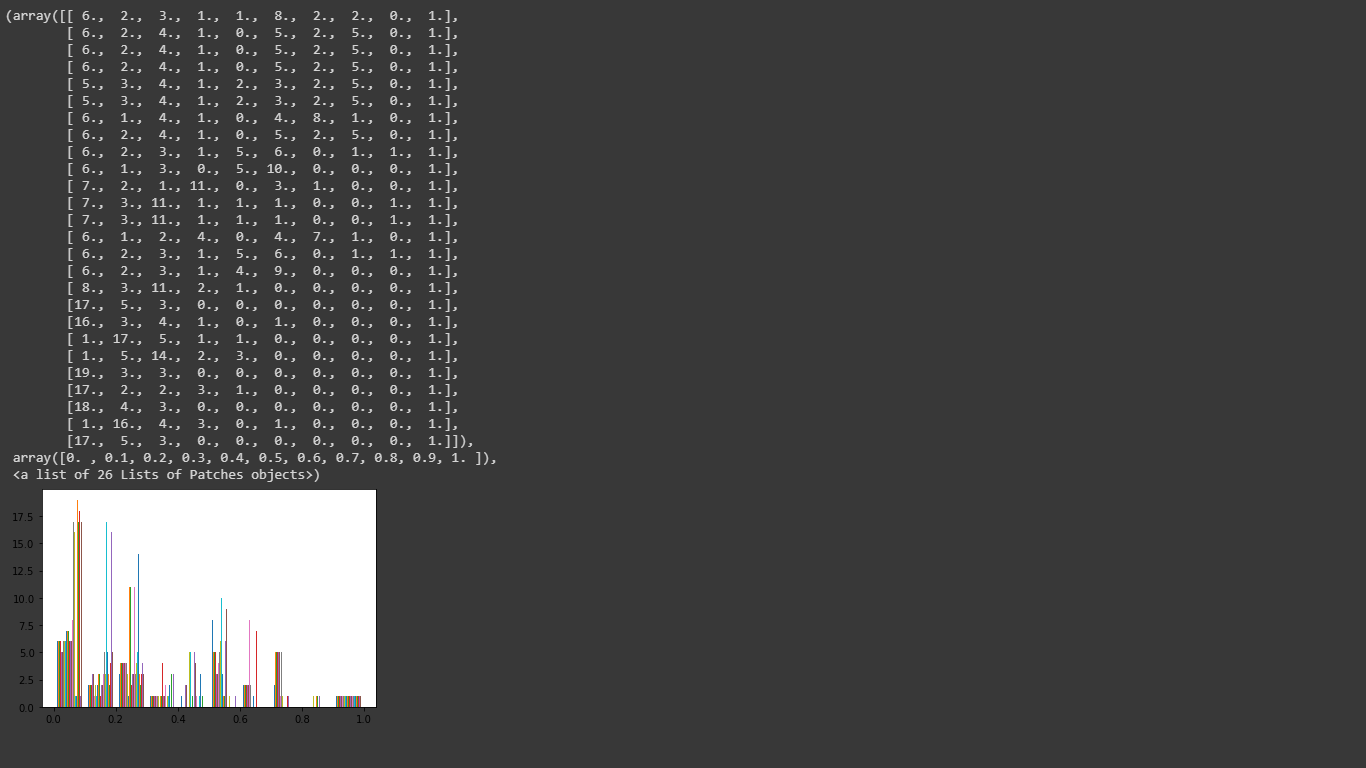
Pada Gambar 5 terlihat nilai pembobotan dari setiap kata yang dilakukan pada proses Feature Item Set. Pembobotan ini dilakukan menggunakan pendekatan Term Frequency-Inverse Document Frequency (TF-IDF). TF menghitung jumlah kemunculan kata pada suatu dokumen, artinya semakin, kata itu sering muncul, maka pembobotannya akan semakin meningkat secara linear, misal kata ‘user’ muncul 2 kali pada dokumen, maka pembobotan TF akan memberikan nilai pembobotan 2, sedangkan IDF menghitung invers fungsi logaritma dari jumlah dokumen yang mengandung kata tersebut, artinya semakin banyak dokumen yang mengandung kata tersebut, bobot dari kata tersebut akan semakin turun menurut fungsi logaritmik, sehingga nilai bobot totalnya adalah hasil perkalian dari pembobotan TF dan pembobotan IDF.



Gambar 5. Pembobotan untuk setiap kata

1. Feature Items Combination & Standardization

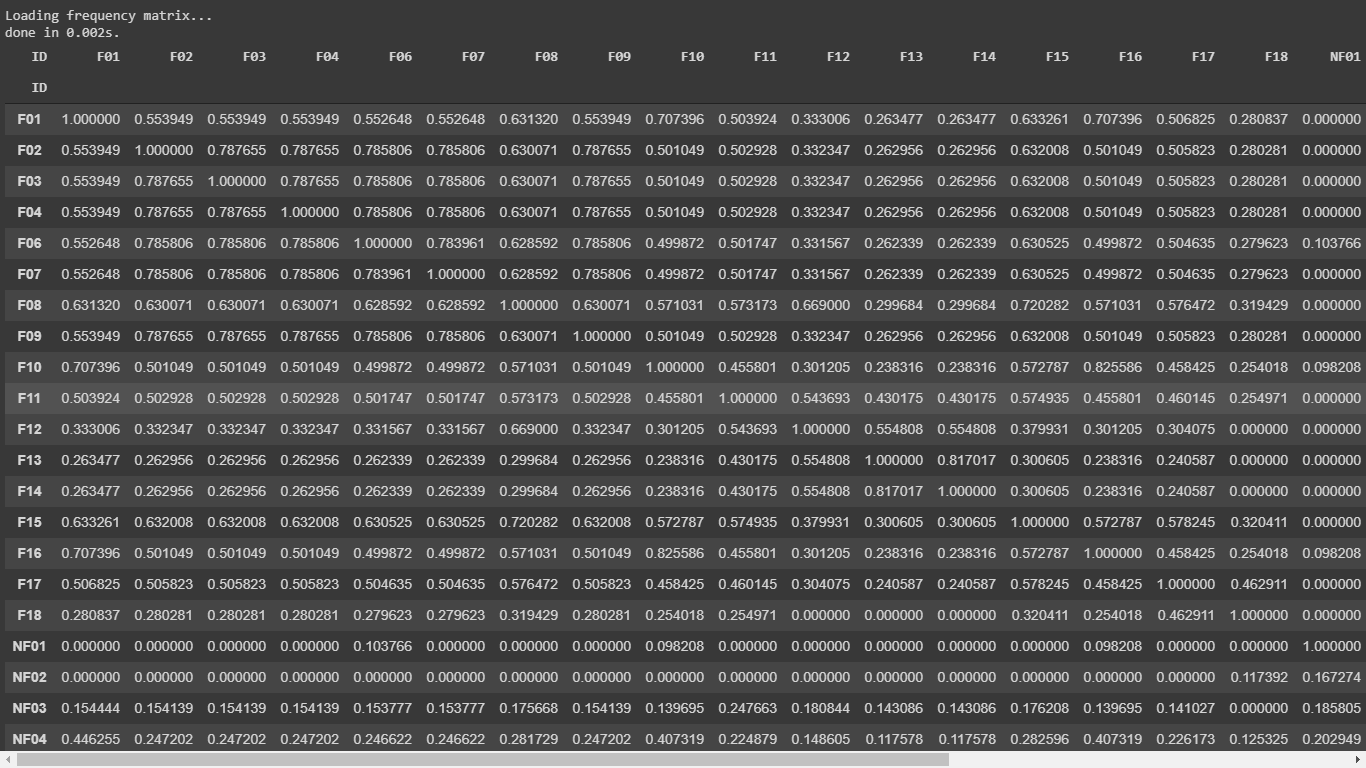
Lalu selanjutnya kita menggunakan Feature Item Combination & Standardization sesuai dengan Gambar 6. menggunakan metode similaritas cosine yang digabung berdasarkan nilai hasil dari Term Frequency - Inverse Document Frequency. Sehingga didapatkan nilai kesamaan antara F01 sampai NF07. Hal ini kita dapatkan nilai Histogram sebagai berikut.



Gambar 6.Histogram Feature Item Combination & Standarization

1. Standardized Feature Item Set

Pada Gambar 7 terlihat matriks tingkat kemiripan dari masing masing kalimat kebutuhan yang ada pada dokumen. Nilai kemiripannya ini mempunyai rentang dari 0 sampai 1, 1 artinya kedua kebutuhan yang dibandingkan sangat mirip sedangkan 0 menandakan tidak ada kemiripan sama sekali

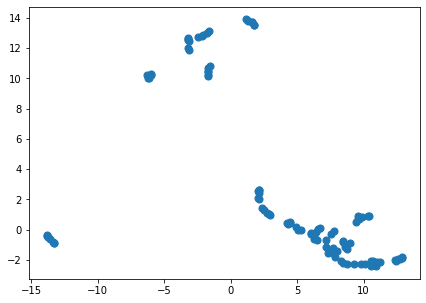


Gambar 7. Nilai kemiripan dari masing-masing kalimat kebutuhan

1. Document Vector

Pada nilai vektorisasi yangd dijelaskan pada Gambar 8. bahwa penggunaan word2vec berdasarkan perbandingan kata dasar yang berjumlah 104 dan nilai dari term setiap pernyataan kebutuhan berjumlah 26. Sehingga didapatkan nilai dari pendekatan untuk mendapatkan kata-kata yang muncul pada dokumen tersebut serta label pendekatan mana saja yang dekat dengan kata dasar tersebut serta buat nilai biner 0 dan 1 dari term tersebut. Selanjutnya dilakukan proses training antara data X dan label Y menghasilkan nilai bentuk yang sama yaitu 666 bentuk dan panjang 104 kata.

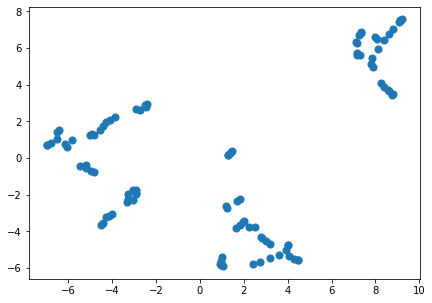
Kemudian dilanjutkan dengan pengujian pada weight dan backpropagation, untuk mendapatkan nilai prediksi vektor yang kita uji. Dari hasil beberapa training yang kita uji dari 10000 iterasi dan diambil nilai loses sekitar 2.26% dan kita ambil nilai vektorisasinya berdasarkan weight dan backpropagation tersebut. Lalu kita mengujinya bila kata coba cari fungsi pencarian kata yang terdekat dari ‘abl’ adalah ‘run’. Setelah itu baru kita cari nilai jarak terdekat dari kedua kata tersebut, antara ‘abl’ dan ‘run’ sekitar 1.64 sedangkan bila kita bandingkan kata ‘abl’ dan ‘abov’ 4.57. sehingga bisa simpulkan dengan pendekatan word2vec ini kita bisa latih kata mana saja yang sama dengan berdekatan sesuai dengan pernyataan kebutuhan pada dokumen tersebut.



Gambar 8. Representasi Word2Vec

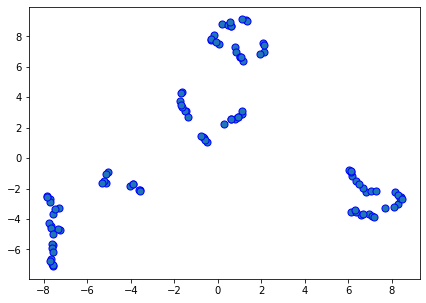
1. Clustering

Pada Gambar 9 terlihat visualisasi dari hasil cosine similarity yang didapat untuk setiap kalimat kebutuhan. pada gambar terlihat ada beberapa kalimat kebutuhan yang berdekatan dan ada juga yang berjauhan, dengan begini kita dapat melakukan clustering untuk mengelompokkan kalimat kalimat kebutuhan tersebut.



Gambar 9. Sebelum K-Means Clustering

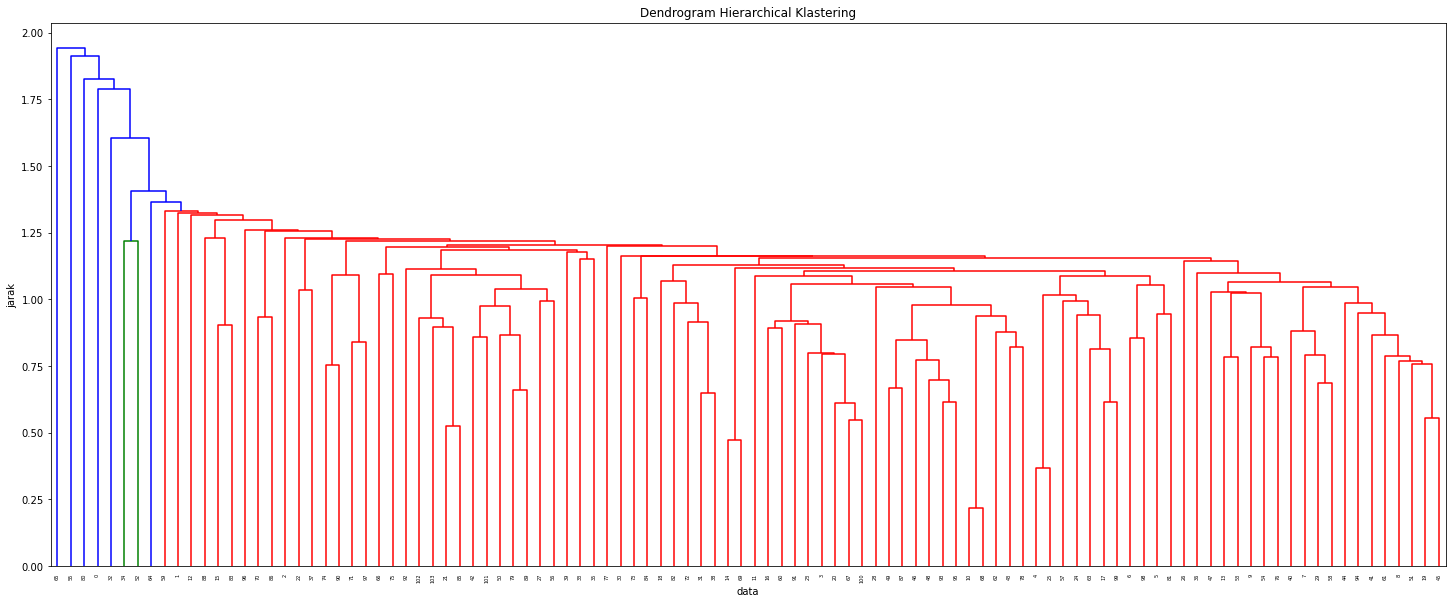
Clustering yang dilakukan menggunakan algoritma K-Means Clustering, hasil dari proses clustering dapat dilihat pada Gambar 8. Representasi nilai kemiripan dari setiap kalimat kebutuhan terlihat semakin jelas.



Gambar 10. Setelah K-Means Clustering

1. Ontology Construction

Setelah dilakukan proses clustering yang telah dilakukan oleh K-Means clustering pada proses sebelumnya. Maka dilakukan proses Hirarki Clustering sesuai dengan Gambar 11. bisa dilihat kita membuat kluster sesuai dengan jarak diambil sesuai dengan linkage masing-masing simpul yaitu berupa beberapa kelas antara lain berdasarkan term dan masing-masing node yang terhubung.



Gambar 9. cluster.HIrarki