**LAPORAN**

**KONSTRUKSI INDEKS**

**SISTEM TEMU BALIK INFORMASI**

**Semester Genap 2019/2020**

****

Disusun oleh:

1118012 – Benedict Reydo

1118021 – Michelle Natasha Irawan

1118023 – Daniel Alexander

**Departemen Informatika**

**Institut Teknologi Harapan Bangsa**

**Jalan Dipatiukur No. 80-84, Bandung**

**2020**

**BAB 1**

**PENDAHULUAN**

1. **LATAR BELAKANG**

Informasi adalah sekumpulan data atau fakta yang telah diproses dan dikelola sehingga menjadi suatu yang mudah dimengerti serta bermanfaat bagi banyak orang. Informasi tersebut diperoleh dari data di mana data merupakan fakta yang bersifat mentah dan belum diolah. Dengan kemajuan teknologi yang ada untuk menemukan sebuah informasi kita dapat menemukannya dalam jaringan internet. Internet yang mengimplementasikan sistem temu-balik informasi akan memberikan informasi sesuai keinginan pengguna agar kegiatan yang pengguna lakukan dapat terpenuhi.

1. **RUMUSAN MASALAH**
2. Apakah kegunaan sistem temu-balik informasi?
3. Apa yang dimaksud indeks language model?
4. Bagaimana indeks language model bekerja?
5. **TUJUAN PEMBAHASAN**

Berdasarkan rumusan masalah di atas, adapun tujuan bagaimana indeks terbalik dilakukan:

1. Mengetahui bagaimana sistem temu-balik informasi bekerja.
2. Mengetahui indeks language model serta cara indeks language model bekerja.

**BAB 2**

**PEMBAHASAN**

Pencarian informasi saat ini dilakukan oleh mesin pencari yaitu dengan cara pengguna menuliskan kueri yang dibutuhkan. Kueritersebut akan digunakan oleh mesin pencari di mana mesin pencari tersebut akan menampilkan hasil perolehan pencarian yang banyak dengan bentuk dokumen dengan hasil relevan sesuai keinginan pengguna.

Sistem temu-balik informasi atau yang disebut sebagai temu balik informasi digunakan untuk menemukan informasi-informasi tersebut yang relevan dengan kebutuhan pengguna yang bertujuan untuk menjawab kebutuhan informasi pengguna. Aplikasi umum yang digunakan sistem temu-balik informasi adalah *search-engine* atau mesin pencari yang terdapat dalam jaringan internet yang salah satunya adalah Google.

Untuk mendapatkan manfaat kecepatan pengambilan indeks kita harus membangun indeks di muka yaitu dengan cara mengumpulkan dokumen untuk diindeks, mengubah daftar token, hingga menghasilkan indeks yang terdiri dari sebuah kamus dan posting.

Temu balik informasi dapat mencakup semua jenis data dan informasi yang ditentukan oleh pengguna. Temu balik informasi membutuhkan data terstruktur untuk memudahkan *search-engine* dalam pencarian data yang ada. Data tersebut akan diberi indeks yang terurut secara alfabet guna mengatur file agar terstruktur rapi.

Biasanya untuk mencari informasi seorang pengguna mengajukan pertanyaan sesuai dengan pemikiran berupa kata-kata yang mungkin akan muncul dalam dokumen yang relevan, serta menggunakan kata-kata tesebut sebagai kueri. Ada beberapa model yang digunakan untuk menentukan dokumen sesuai dengan permintaan pengguna berdasarkan kata kunci yang didapatkan. Model yang digunakan antara lain model boolean, pembobotan term dan model ruang vektor, dan language modeling. Model boolean memiliki kekurangan di mana pengembalian berdasarkan kriteria keputusannya tanpa gagasan pencocokan parsial sehingga kita tidak mendapatkan output, informasinya pun tidak relevan.

Dengan kelemahan model boolean maka ada model pembobotan term dan model ruang vektor. Pembobotan term yaitu memberikan tanda pada term yang berguna untuk mengevaluasi pentingnya term dalam dokumen.

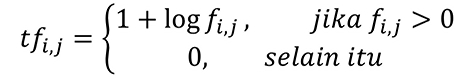
temu balik informasimenggunakan *language modeling* untuk langsung memodelkan gagasan tersebut di mana sebuah dokumen sangat cocok dengan kueri jika model dokumen tersebut cenderung menghasilkan kueri.

1. **RANCANGAN STRUKTUR DATA METODE RUANG VEKTOR**

Dalam pembuatan aplikasi ini, metode yang digunakan untuk metode ruang vektor adalah TF IDF atau yang disebut dengan pembobotan term. Pembobotan term berguna untuk mengevaluasi pentingnya term dalam dokumen untuk pencarian kembali informasi yang muncul pada dokumen.

Berikut merupakan tahap-tahap pembuatan aplikasi menggunakan metode ruang vektor TF IDF:

1. Mencari nilai f atau frekuensi pada kamus beserta dengan kuerinya. Frekuensi tersebut akan menghasilkan nilai ni di mana ni merupakan nilai hasil perhitungan kueri yang muncul pada berapa dokumen dan termasuk frekuensi pada kueri.
2. Nilai ni tersebut akan diakumulasikan untuk tiap term yang hendak dicari.
3. Nilai ni pada aplikasi ini akan mencari nilai tf. Nilai tf tersebut didapatkan menggunakan log normalization dengan rumus:

.

1. Mengimplementasikan idf, dengan menggunakan inverse *frequency smooth*, kami menggunakan rumus: **log(1 + N/ni)**.
2. Mencari nilai w atau *weight* yang didapatkan dengan mengkalikan nilai tf dengan nilai idf untuk masing masing term yang dicari.
3. Mencari nilai panjang dokumen yang akan kita misalkan dengan kata dj. Nilai panjang vektor di sini berguna untuk mendapatkan vektor normal.
4. Mencari nilai kesamaan model vektor antara dokumen ke-j dengan kueri menggunakan pendekatan *cosine* *similarity* atau nilai **sim(dj,q)**. dengan rumus total dari perkalian *weight* ke j dengan *weight* ke q untuk sebuah dolumen dibadi dengan akar dari vektor normal dari dokumen ke j dikali dengan akar dari vektor normal kueri untuk masing-masing dokumen.
5. Tahap ke-7 menghasilkan ranking dokumen-dokumen tersebut yang akan dibandingkan dengan kueri yang hendak dicari.
6. **RANCANGAN STRUKTUR DATA METODE PROBABILISTIK**

Adapun model pembuatan indeks *language model*. Awal pembuatan *language model* kelompok kami melakukan iterasi untuk membaca setiap dokumen lalu melakukan pemrosesan awal atau *preprocessing* untuk mendapatkan dokumen-dokumen yang relevan dengan kueri pengguna. Sistem temu balik informasi memungkinkan pengguna menentukan secara cepat dan akurat apakah isi dari dokumen yang diterima memenuhi kebutuhan pengguna.

Pada setiap iterasi kami melakukan penambahan frekuensi masing-masing term dalam suatu dokumen, penambahan frekuensi masing-masing term pada seluruh dokumen, dan menambahkan jumlah kata dalam dokumen tersebut. Hasil-hasil yang telah kami peroleh lalu kami masukkan ke dalam file csv. Hasil jumlah masing-masing term tersimpan dalam struktur data dictionary yang doibah ke dalam file Frequencies.csv, sedangkan total jumlah kata dalam masing-masing dokumen disimpan ke dalam Document Frequencies.csv.

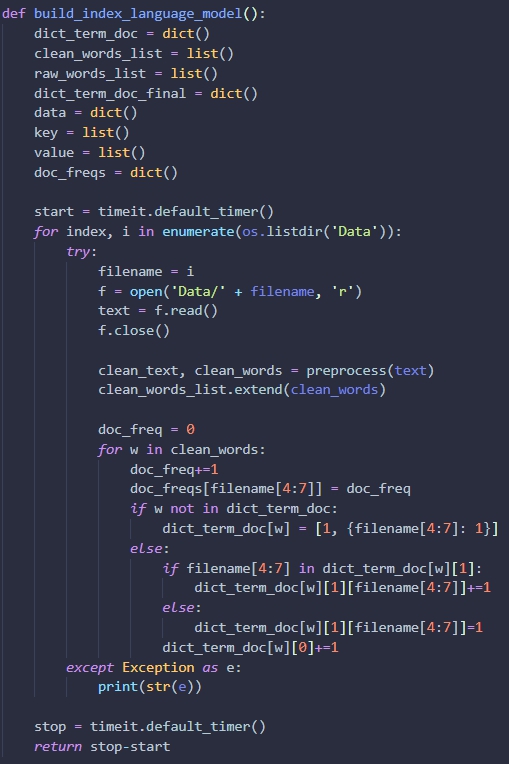
Pencarian dokumen language model merupakan fungsi pencarian yang akan menrima input dari pengguna. Inputan yang diminta dari pengguna berupa input kueri, limit, dan lambda dengan bentuk float atau desimal. Awalnya kami melakukan pemrosesan awal atau *preprocessing* kueri sama seperti yang dilakukan pada pembuatan struktur data indeks. Proses awal yang dilakukan yaitu membaca dokumen Frequencies.csv dan Document Frequencies.csv yang telah dibuat sebelumnya. Lalu, kami melakukan iterasi pada setiap token dalam kueri. Pada setiap iterasi kami melakukan pengambilan total frekuensi setiap dokumen untuk term yang bersangkutan, pengambilan total frekuensi keseluruhan untuk term yang bersangkutan, dan menghitung nilai relevansi dengan menggunakan rumus P().

1. **EVALUASI APLIKASI**

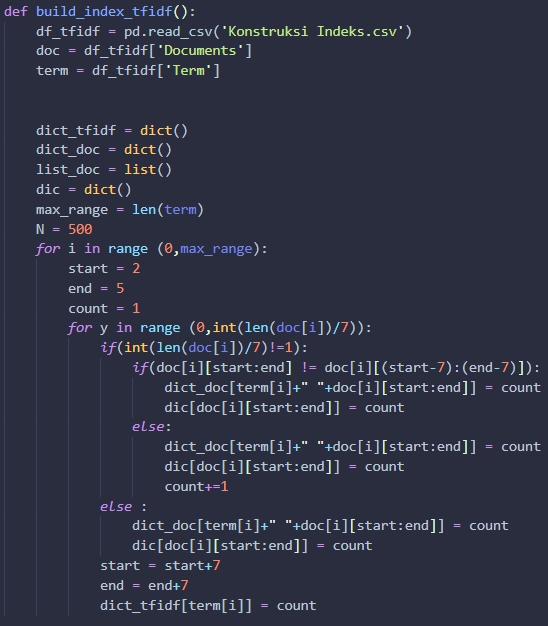
Aplikasi ini menggunakan temu balik informasi berbobot dengan salah satu metode ruang vektor dan salah satu metode probabilistik. Metode yang kami gunakan untuk ruang vektor adalah TF-IDF sedangkan untuk metode prabilistik adalah Model Language.

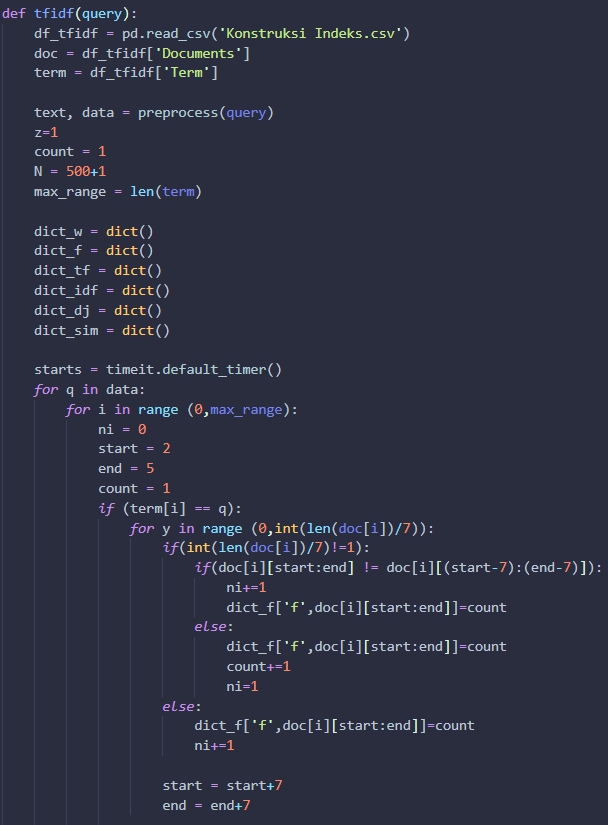
Fungsi-fungsi yang tersedia pada aplikasi:

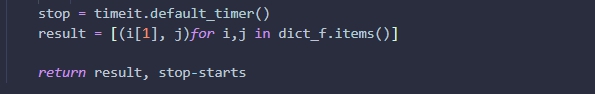
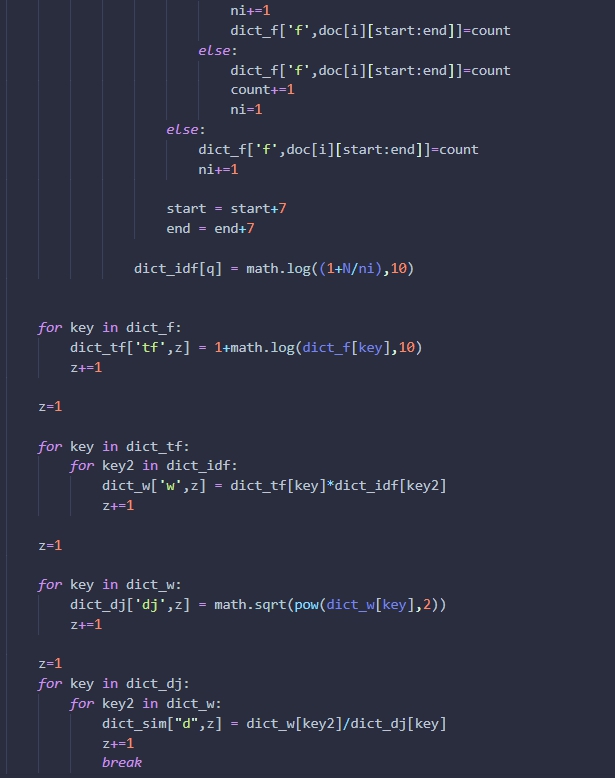
1. TF-IDF



1. Model Language







Untuk mengevaluasi aplikasi ini, kami melakukan perhitungan pada setiap kueri menggunakan perhitungan *precision*, *recall*, dan *F-Measure* seperti tabel di bawah ini:

**Query: 1991 Total: 8**

|  |  |  |  |
| --- | --- | --- | --- |
| **Dokumen** | **Recall** | **Precision** | **F-Measure** |
| 31 | 12.50% | 100.00% | 22.22% |
| 190 | 25.00% | 100.00% | 40.00% |
| 301 | 37.50% | 100.00% | 54.55% |
| 105 | 50.00% | 100.00% | 66.67% |
| 70 | 62.50% | 100.00% | 76.92% |
| 422 | 75.00% | 100.00% | 85.71% |
| 466 | 87.50% | 100.00% | 93.33% |
| 61 | 100.00% | 100.00% | 100.00% |

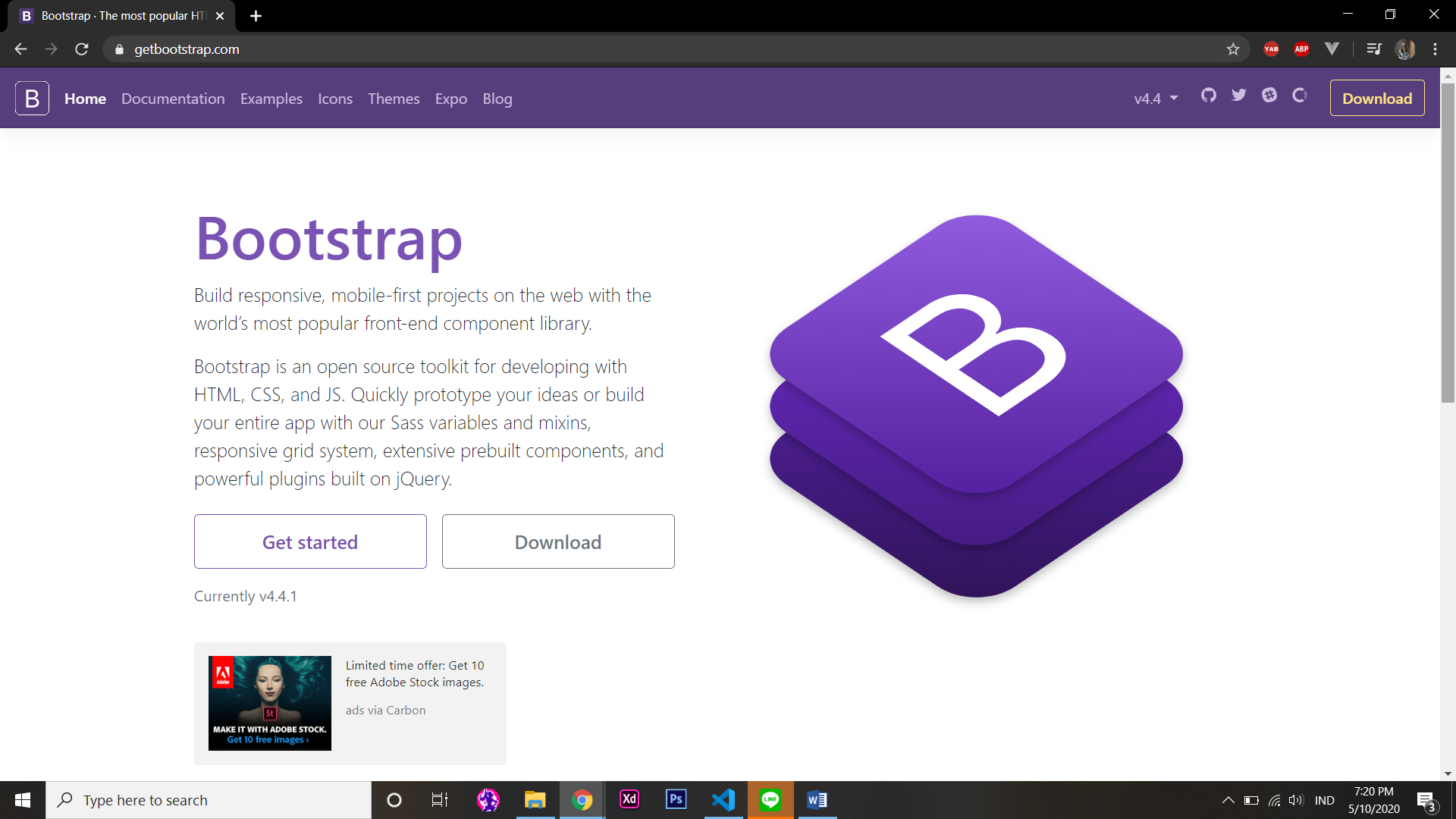
Proses perhitungan pada aplikasi ini dilakukan untuk setiap kueri pada metode pembobotan term dan probabilistik, selebihnya dapat dilihat pada dokumen Excel. Lalu dengan adanya data precision recall kami melakukan plotting grafik.

1. **DESAIN MASUKKAN**

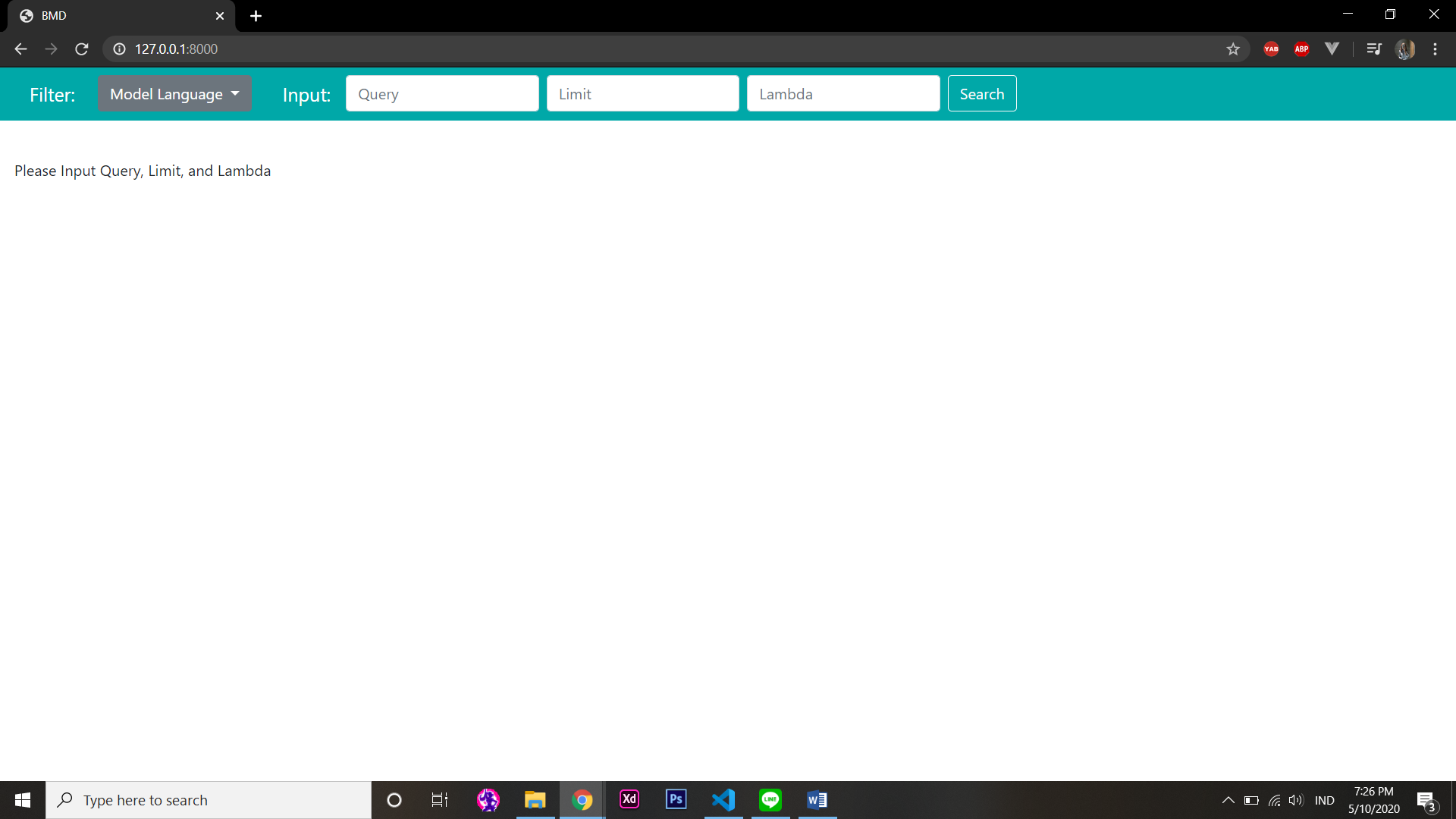
Dalam tugas ini kami menggunakan bahasa pemograman Python untuk mendapatkan data yang dari dokumen dan Django untuk tampilan atau *User Interface* disertai framework Bootstraps.

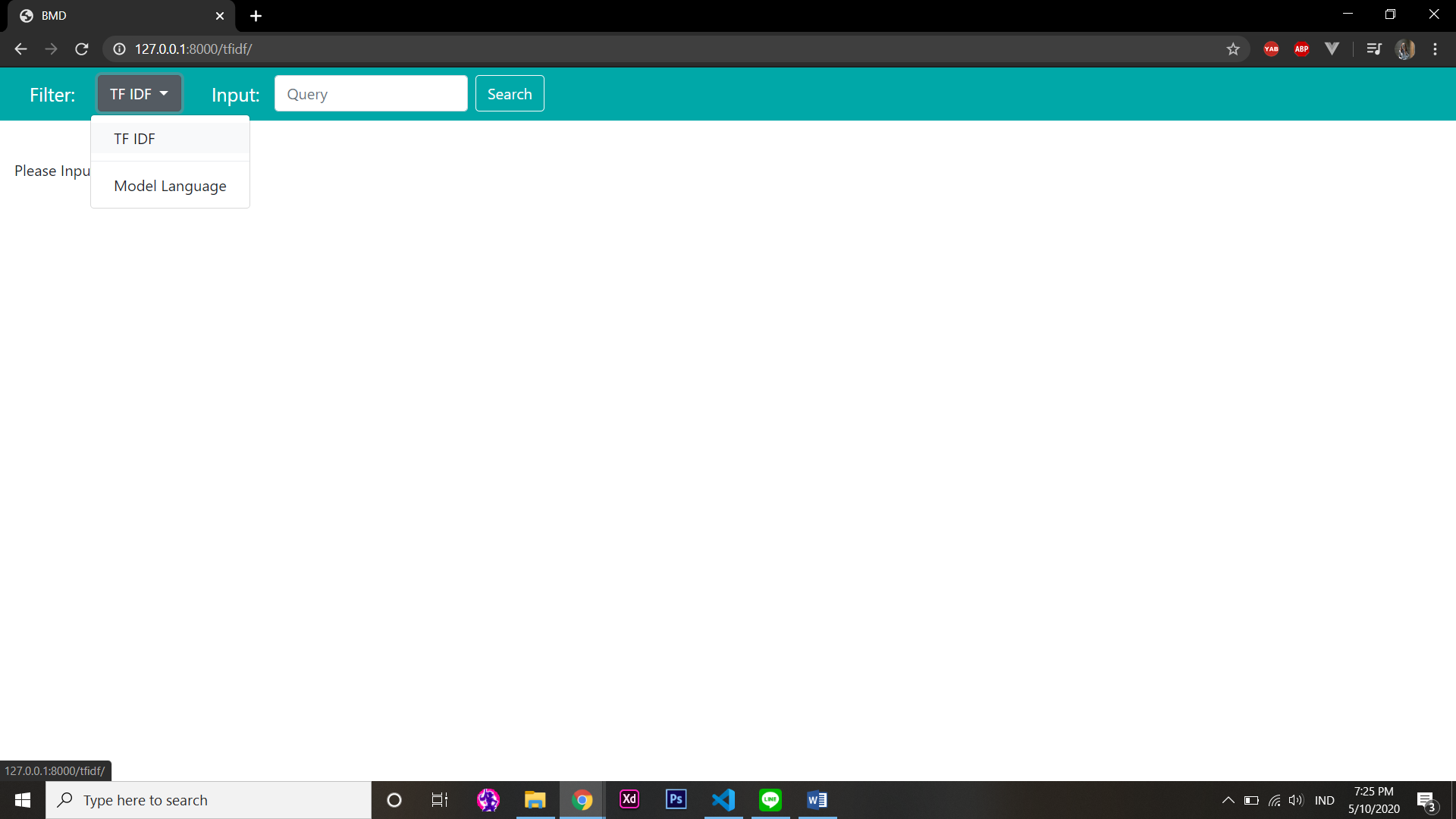
Django adalah high-level Web framework yang mendorong perkembangan dengan cepat serta desain yang pragmatis. Django dibangun oleh *experienced developers* yang telah menangani banyak kesulitan pada pengembangan web.

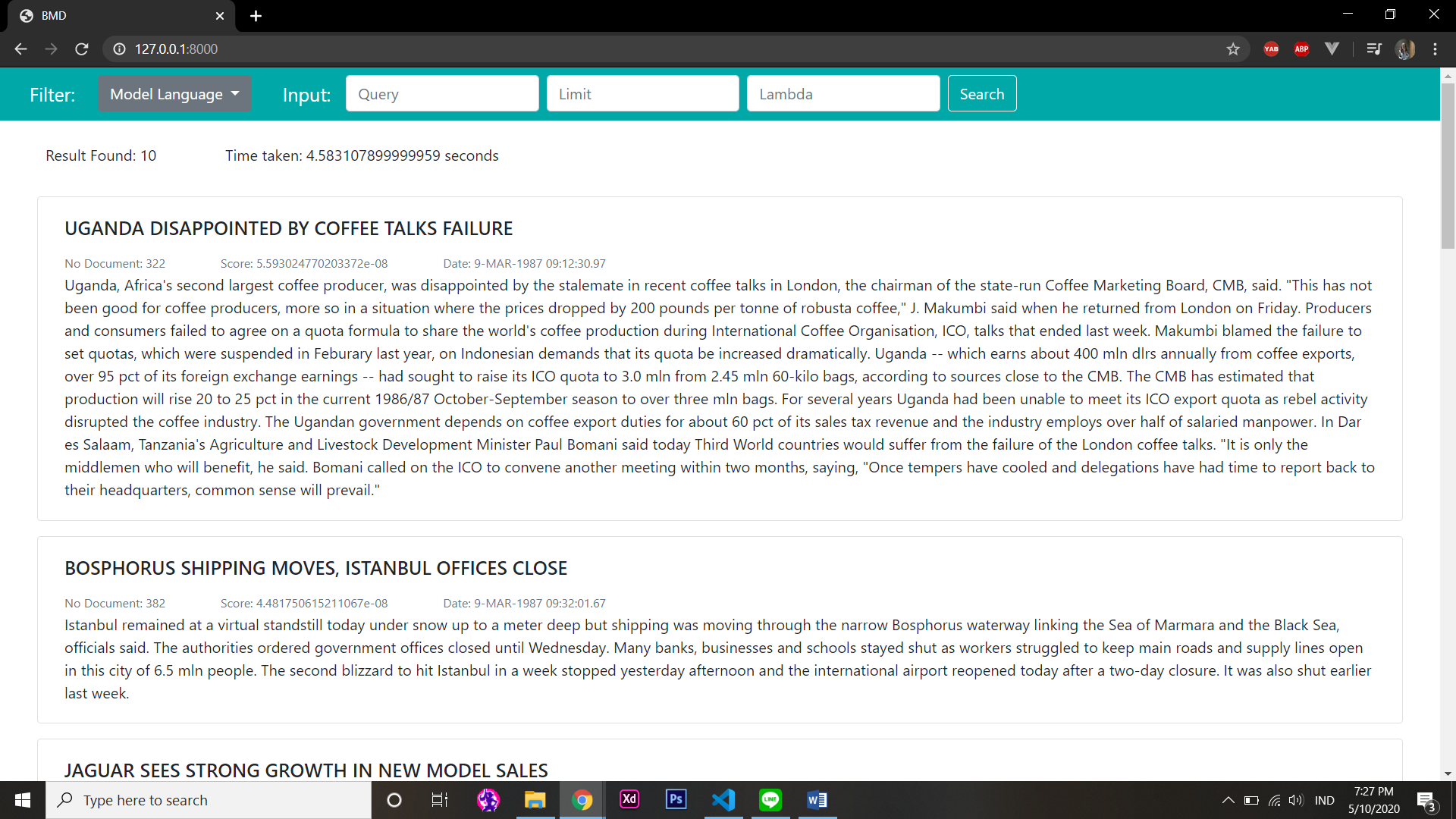
Sedangkan Bootstrap adalah CSS framework open-source yang berfokus pada pengembangan page web dengan cara yang sederhana. Bootstrap menyediakan pilihan-pilihan warna, size, layout, font sehingga pengguna dapat lebih mudah untuk membuat tampilan web.

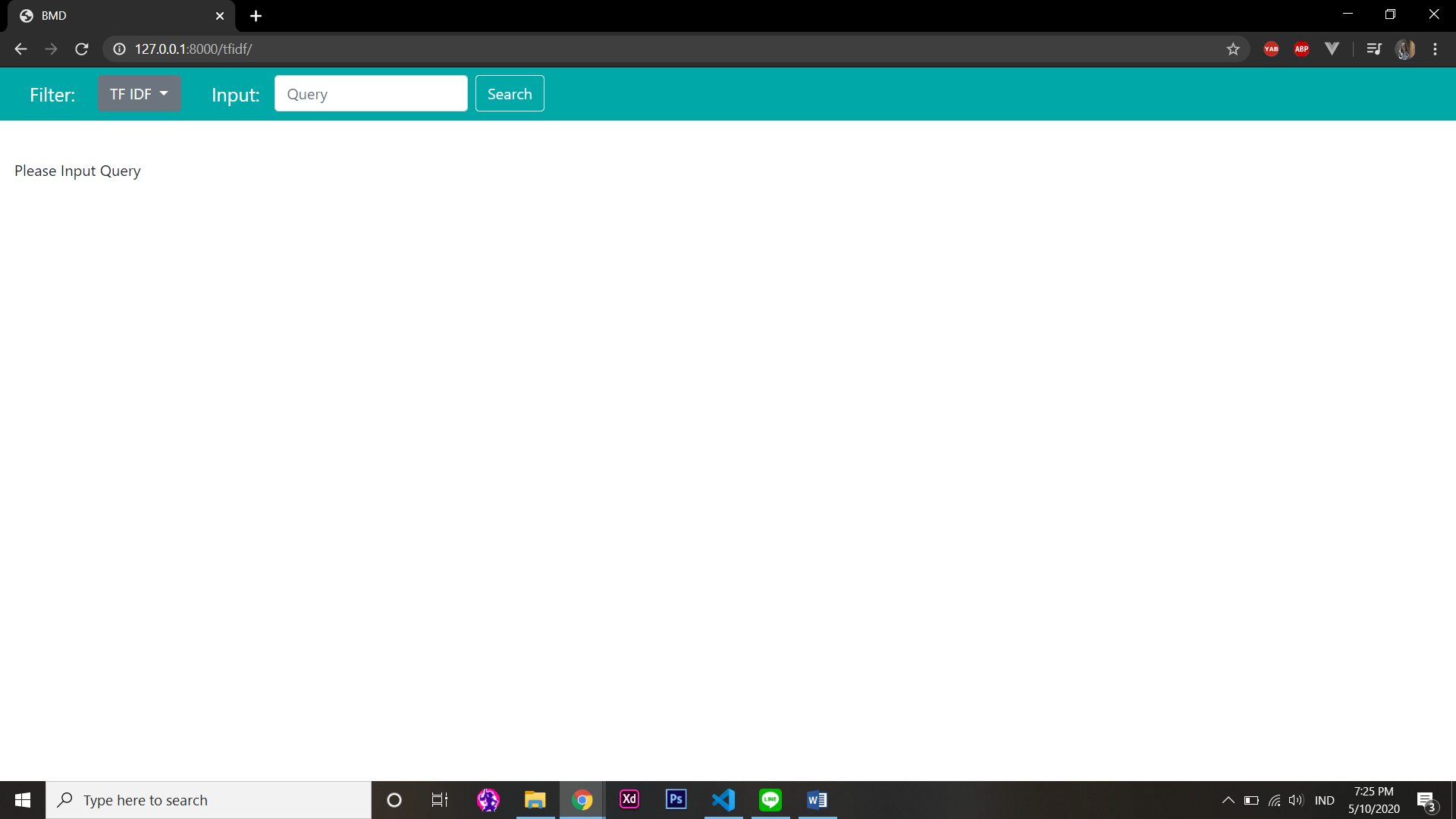


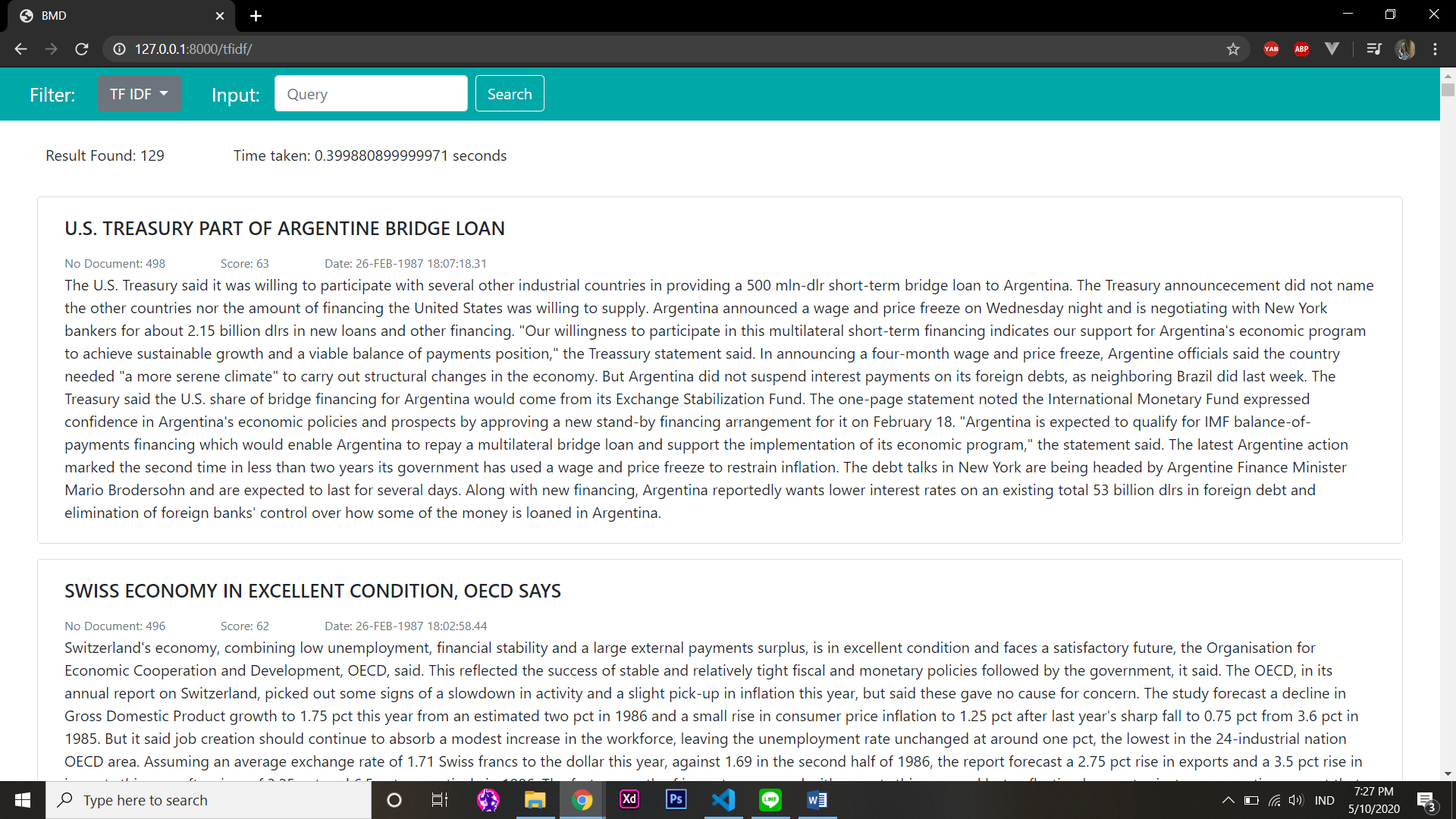
Desain aplikasi:











1. **KESIMPULAN**

Search engine pada aplikasi ini dapat mengembalikan semua dokumen kunci jawaban untuk setiap kueri, sehingga plot *precision* dan *recall* dapat tergambar secara linear. Perbedaan didapatkan berdasar pada waktu saat pembuatan indeks dan pencarian.

Pada aplikasi yang kami buat sesuai dengan grafik maka disimpulkan bahwa pencarian menggunakan metode probabilistik yaitu model language lebih cepat dibandingkan dengan metode ruang vektor.

1. **JOB DESCRIPTION**

|  |  |  |
| --- | --- | --- |
| **NIM** | **Nama** | **Job** |
| 1118012 | Benedict Reydo |  |
| 1118021 | Michelle Natasha Irawan |  |
| 1118023 | Daniel Alexander |  |