

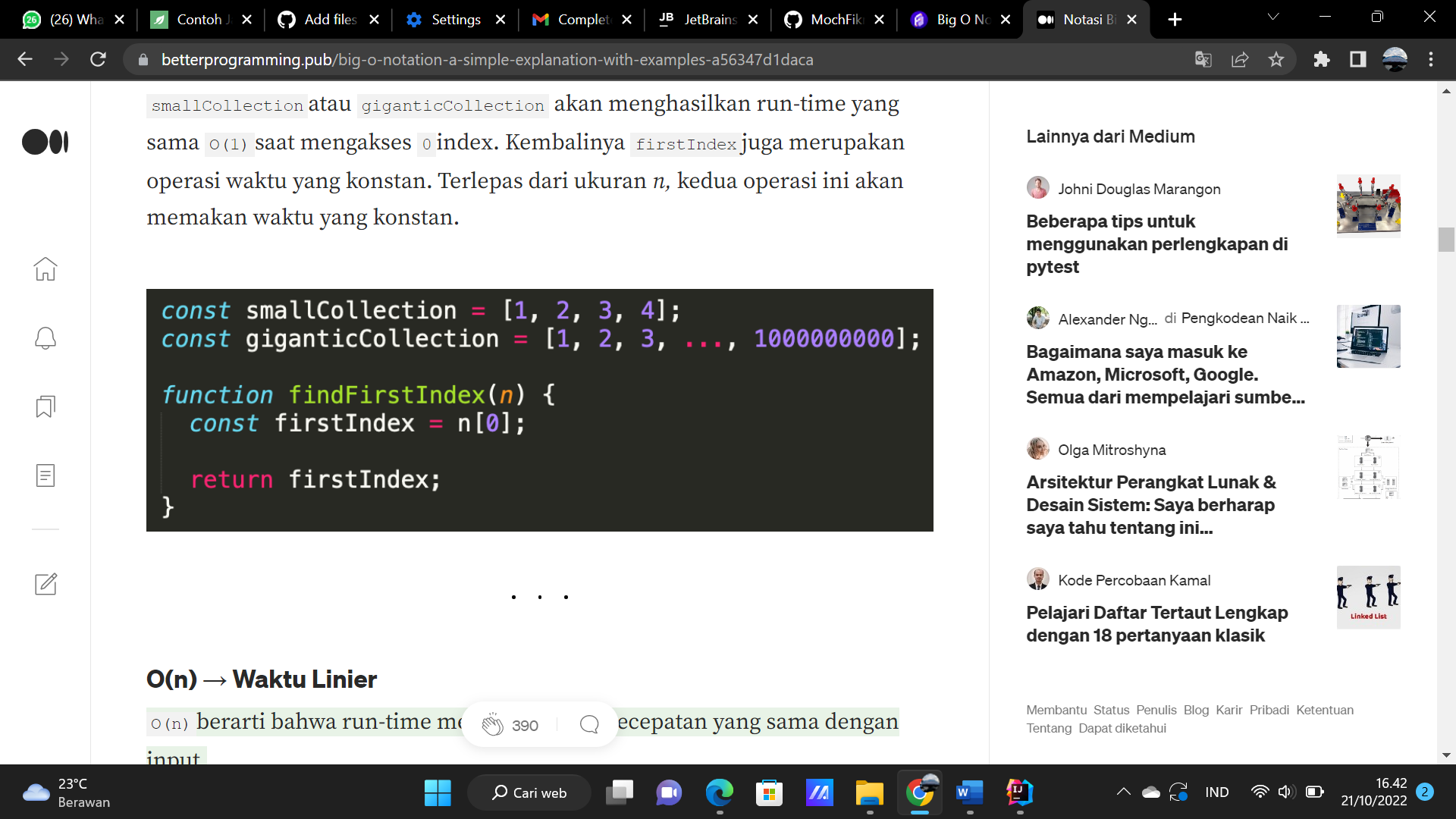
Disusun oleh:

|  |  |
| --- | --- |
| Nama | : Moch Fikry Deniansyah |
| NPM | : 22552011150 |
| Mata Kuliah | : Struktur Data |
| Kelas | : TIF RM – 22A |

**Pengertian dari Contoh Notasi Big-O :**

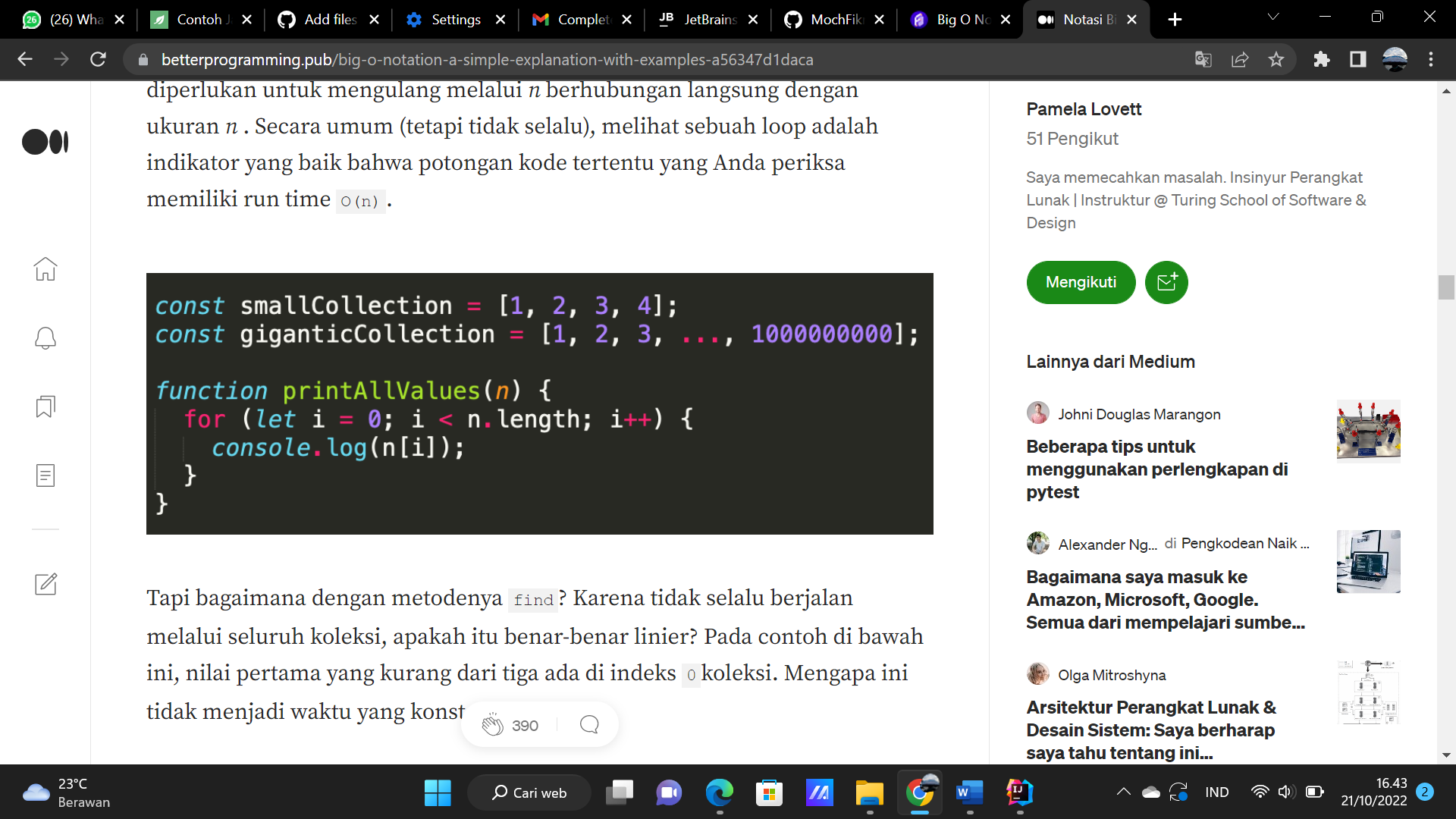
1. **O(1) – Constant Time**

berarti dibutuhkan waktu yang konstan untuk menjalankan suatu algoritma, terlepas dari ukuran inputnya.



1. **O(n) – Linear Time**

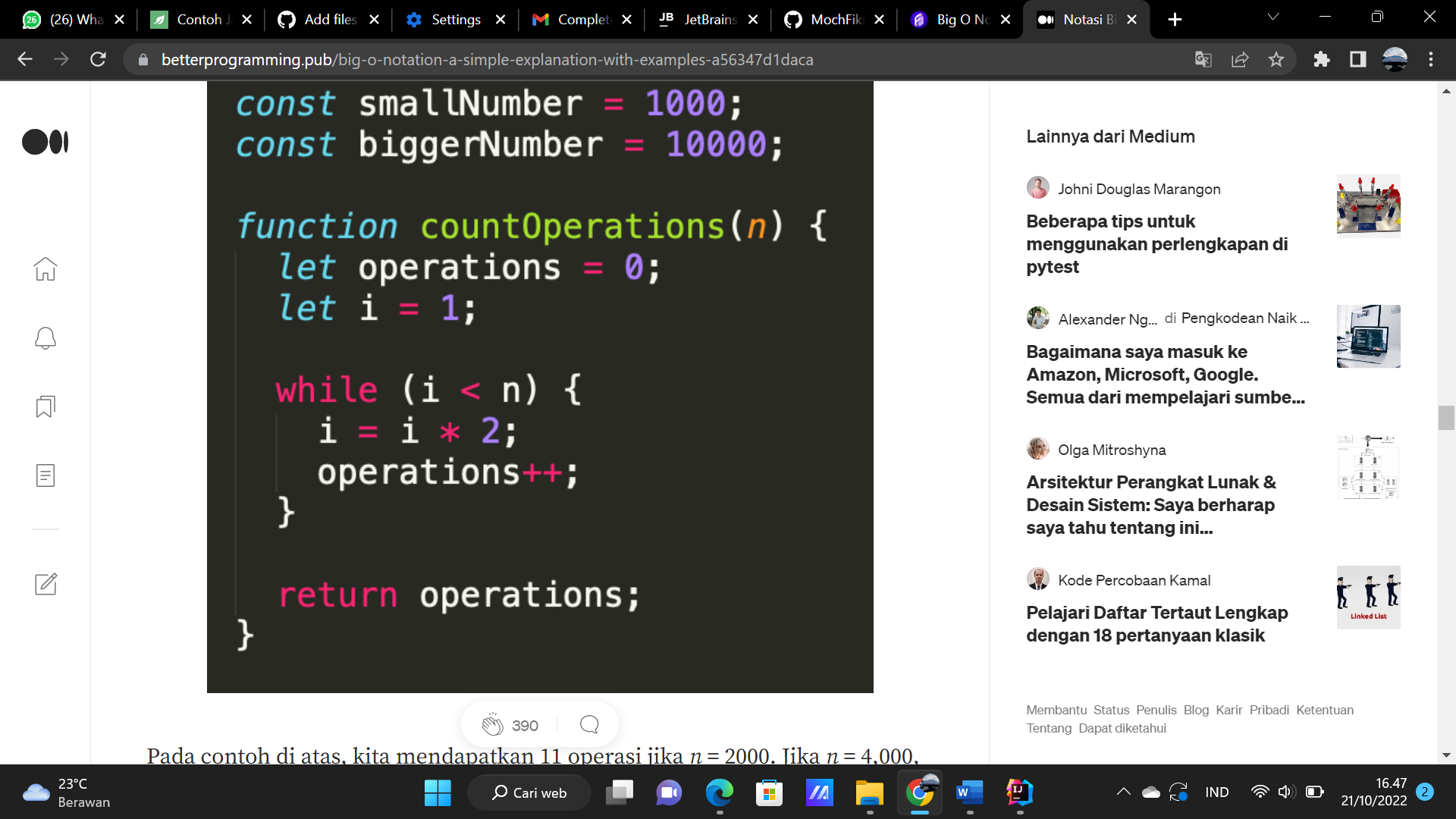
berarti bahwa run-time meningkat pada kecepatan yang sama dengan input.



fungsi Jumlah operasi yang diperlukan untuk mengulang melalui *n* berhubungan langsung dengan ukuran *n* . Secara umum (tetapi tidak selalu), melihat sebuah loop adalah indikator yang baik bahwa potongan kode tertentu yang Anda periksa memiliki run time

1. **O(log n) – Logarithmic Time**

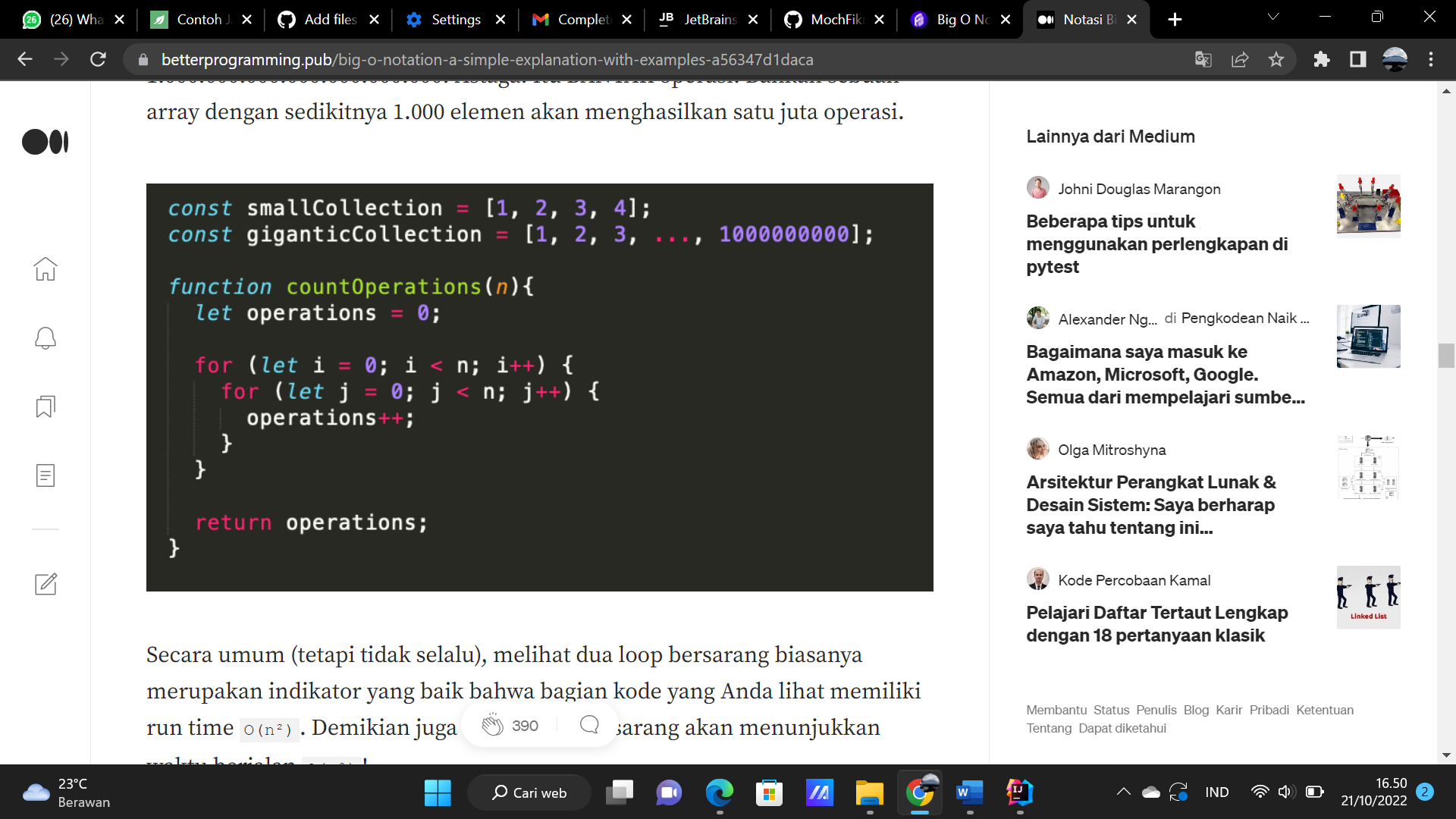
berarti bahwa waktu berjalan tumbuh secara proporsional dengan logaritma dari ukuran input. ini berarti bahwa waktu berjalan hampir tidak meningkat saat Anda meningkatkan input secara eksponensial.



Pada contoh di atas, kita mendapatkan 11 operasi jika n = 2000. Jika n = 4,000, kita akan mendapatkan 12 operasi. Setiap kali kita menggandakan jumlah n , jumlah operasi hanya bertambah satu

1. **O(n2) – Quadratic Time**

berarti bahwa perhitungan berjalan dalam waktu kuadrat, yang merupakan ukuran kuadrat dari data input.



1. **(2 n) – Exponential Time**

Exponential Time biasanya digunakan dalam situasi dimana kita tidak terlalu tahu terhadap permasalahan yang dihadapi, sehingga mengharuskan kita mencoba setiap kombinasi dan permutasi dari semua kemungkinan.