

# Penggunaan Algoritma Brute Force dalam Membandingkan Harga Sebuah Produk di E-commerce

Azka Nabilah Mumtaz/13516013  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung (ITB)  
Bandung, Indonesia  
azkanabilah@gmail.com

**Abstrak**—Dalam kehidupan sehari – hari, manusia selalu membeli barang – barang yang diinginkan dan dibutuhkan. Kebutuhan – kebutuhan manusia tersebut disediakan oleh perusahaan – perusahaan yang kemudian akan diambil keuntungan dari hasil penjualan kebutuhan tersebut oleh perusahaan. Sebelum kemajuan teknologi, perusahaan melakukan penjualan melalui supermarket dan toko – toko yang ada di sekitar kita. Namun, saat ini, seiring dengan perkembangan teknologi, banyak perusahaan yang sudah menggunakan e-commerce, yaitu perdagangan online melalui internet, sebagai salah satu cara untuk melakukan penjualan. Dalam makalah ini, akan dijelaskan tentang pengambilan harga minimum dari sebuah jenis produk yang ditawarkan oleh seluruh e-commerce tersebut dengan menggunakan algoritma brute force.

**Kata kunci**—perdagangan, e-commerce, brute force.

## I. PENDAHULUAN

E-commerce atau perdagangan *online* / elektronik merupakan transaksi jual beli barang dan jasa serta pemasarannya yang dilakukan melalui internet sehingga pertemuan antar penjual dan pembeli serta sistem pembayarannya pun dilakukan di dunia maya.

Sebelum e-commerce diciptakan, untuk memenuhi kebutuhan manusia, praktik jual beli masih dilakukan secara langsung, yaitu penjual bertemu langsung dengan pembeli untuk melaksanakan pemasaran dan penawaran. Pembayaran pun dapat dilakukan secara *cash*, tawar – menawar juga dapat dilakukan secara *face to face* antar penjual dengan pembeli.

Namun, sejak internet pertama kali muncul, teknologi kian berkembang sehingga timbulah e-commerce sebagai solusi perdagangan yang lebih efektif dan efisien. Dapat dilihat dari namanya, e-commerce juga memberikan layanan transaksi dana secara elektronik, mengumpulkan data perjual-belian secara elektronik, dan seluruh kegiatan jual beli lainnya dilakukan secara elektronik yaitu melalui internet.

Secara lebih ringkas, e-commerce dapat menangani masalah – masalah ini :

1. Otomatisasi  
Proses otomatisasi dapat menggantikan proses manual, proses ini dilakukan dengan konsep *enterprise resource planning*
  2. Integrasi  
Proses terintegrasi sehingga dapat meningkatkan keefektifan dan keefisienan proses, proses ini dilakukan dengan konsep *just in time*
  3. Publikasi  
Memberikan jasa promosi serta komunikasi atas produk barang dan jasa yang ditawarkan proses ini dilakukan dengan konsep *electronic cataloging*
  4. Interaksi  
Terjadi interaksi dan pertukaran antar data dan informasi sehingga meminimalisasi terjadinya *human error*, proses ini dilakukan dengan konsep *electronic data interchange*
  5. Transaksi  
Terjadi kesepakatan antar dua pihak, dalam hal ini tentunya adalah penjual dan pembeli, sehingga dalam transaksi tersebut melibatkan institusi lainnya sebagai pihak yang menangani pembayaran, proses ini dilakukan dengan konsep *electronic payment*
- Selain dapat menangani berbagai masalah di atas, e-commerce juga memiliki beberapa keuntungan bagi penggunaannya, yang pertama adalah penjual barang atau jasa dapat melakukan penjualan secara *online*, sehingga mereka tidak perlu mendirikan toko atau kantor besar seperti yang dilakukan oleh para penjual offline, dan tidak perlu mengeluarkan uang untuk itu. Mereka hanya perlu jaringan internet dan membayar kuotanya.

Keuntungan lainnya adalah mudahnya komunikasi antara penjual dengan pembeli, karena penjual hanya perlu meletakkan

informasi penjualannya di internet, dan pembeli hanya perlu menghubungi penjual yang tersedia untuk membeli barang tersebut. Dan semua proses tersebut dapat dilakukan hanya dengan menggunakan jaringan internet, sehingga penjual tidak perlu mengeluarkan biaya yang tinggi untuk melakukan promosi produk yang ditawarkan.

Contoh – contoh E-commerce di sekitar kita :

- Belanja online (Tokopedia, Shopee, Lazada, dll)



Gambar 1.1 Contoh e-commerce untuk belanja online

- Tiket online (Traveloka, tiket.com, dll)



Gambar 1.2 Contoh e-commerce untuk tiket online

- Booking hotel online (Booking.com, Traveloka, dll)



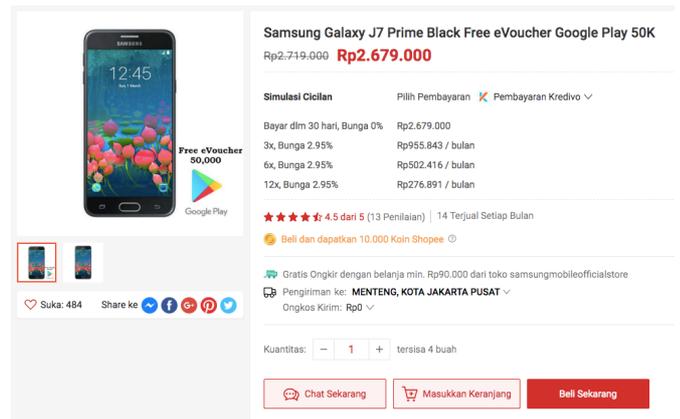
Gambar 1.3 Contoh e-commerce untuk booking hotel

- dsb.

## II. DASAR TEORI

### A. Transaksi E-Commerce

Transaksi E-Commerce diawali dengan penjual online, atau yang disebut dengan *e-merchant*, menawarkan produk barang atau jasa melalui server yang disewa dari *Internet Server Provider (ISP)* oleh *e-merchant*. Dalam menawarkan produk, *e-merchant* akan menyertakan foto, harga, serta deskripsi dari produk tersebut. Hal tersebut yang akan menjadi pedoman para pembeli, atau yang disebut dengan *e-customer*, dalam memilih produk yang mereka inginkan sebelum terjadi kesepakatan dan transaksi.



Gambar 2.1 *E-merchant* menyertakan foto, harga, serta deskripsi ketika menawarkan produk

Kemudian *e-customer* akan mencari produk yang ingin dibeli melalui server tersebut, dan setelah cocok, *e-customer* dan *e-merchant* akan bertemu dan berkomunikasi melalui dunia maya.

Transaksi *e-commerce* selalu disertai dengan *term of use* dan *sales term condition*, biasanya *e-merchant* meletakkan isi kesepakatan tersebut pada websitenya, sehingga *e-customer* dapat langsung menekan tombol *accept* jika berminat. Penekanan tombol *accept* tersebut adalah bentuk terjalannya kesepakatan jual beli antara *e-customer* dengan *e-merchant*.

Setelah mencapai kesepakatan, akan terjadi proses pembayaran, di mana proses tersebut akan melibatkan dua bank dari masing – masing pihak yaitu bank milik *e-customer* atau disebut juga dengan *issuing customer bank* dan bank milik *e-merchant* atau disebut juga dengan *acquiring merchant bank*. Prosedur pembayarannya *e-customer* memerintahkan kepada *issuing customer bank* untuk dan atas nama *e-customer* melakukan sejumlah pembayaran atas harga barang kepada *acquiring merchant bank* yang ditujukan kepada *e-merchant*.

Setelah proses pembayaran dilakukan, *e-merchant* akan mengirimkan produk yang telah disepakati menuju alamat *e-customer*. Proses transaksi *e-commerce* berakhir dan berhasil setelah produk sampai ke alamat *e-customer*.

### B. Basis Data

Basis data adalah suatu kumpulan data yang disusun data yang sudah teratur, teorganisir, dan tersusun dalam bentuk tabel – tabel baik yang saling berkaitan maupun berdiri sendiri. Kumpulan data tersebut disimpan secara bersama – sama ke dalam suatu media.

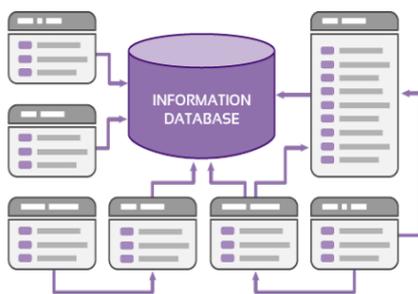
Saat mencari minimum harga sebuah produk dari beberapa *e-commerce*, tentunya diperlukan DBMS atau *Data Base Management System* untuk mengatur dan mengaplikasikan basis data tersebut agar dapat digunakan. Di bawah ini adalah beberapa fungsi utama DBMS dalam mengatur basis data :

- Mengintegrasikan data pada basis data ke computer klien, menyebabkan semua data yang terdapat pada *database* di dalam server bisa tersaji di computer *client*,

sehingga *client* dapat mengakses informasi dari data yang telah terintegrasi tersebut.

- DBMS dapat digunakan untuk memperbarui data atau disebut juga dengan *update data*, hal ini menyebabkan *user* dapat memperbarui data yang tersimpan di dalam basis data dengan mudah, hanya dengan menggunakan *software* yang mendukung DBMS dan tidak perlu membuka basis data.
- Melakukan pemanggilan basis data untuk kepentingan pengambilan informasi, sehingga *user* dapat melihat informasi dari basis data dengan cepat.
- Membantu *user* mengakses basis data, sehingga *user* dapat mengakses basis data yang tersedia tanpa perlu membuka file basis data. *User* dapat mengakses hanya dengan menggunakan aplikasi yang sudah terintegrasi dengan DBMS.
- DBMS dapat membantu toko *online* untuk melihat proses transaksi yang sedang berjalan, yang terekam, dan masuk ke dalam basis data toko atau perusahaan.

Tabel pada basis data merupakan kumpulan dari suatu *field*, yaitu suatu atribut yang mewakili sebuah *record*, dan *record*, yaitu kumpulan elemen – elemen yang saling berkaitan dan memberikan informasi isi data secara lengkap. Setiap entitas pada basis data juga memiliki *primary key*, yaitu suatu kolom atau *field* yang menjadi titik acuan untuk mengakses sebuah tabel. *Primary key* tersebut biasanya bersifat unik, tidak ada yang memiliki nilai yang sama.



Gambar 2.2 Ilustrasi basis data

Tabel Barang

ID Barang	Barang	Toko	Harga	Jumlah
014312	Galaxy J7	A	2.679.000	3
014313	Galaxy J7	B	2.450.000	5
014314	Galaxy J7	C	2.560.000	6
014315	Galaxy J7	D	2.150.000	7

Tabel Toko

ID Toko	Nama Toko	Nama Pemilik	Rating
13513	A	Andi	3,5
13514	B	Benny	4
13515	C	Carol	5
13516	D	Danny	4,5

Tabel Transaksi

ID Toko	ID Akun	ID Barang	ID Transaksi
13516	101397	014315	015ACD
13514	101398	014313	014RFG
13516	101399	014315	027BZK

Tabel Akun

ID Akun	Nama	Gender
101397	Azka	P
101398	Faiz	L
101399	Yasya	P

Tabel 2.1 Contoh database pada sebuah e-commerce

### C. Algoritma Brute Force

Algoritma Brute Force adalah algoritma pemecahan suatu masalah dengan menggunakan pendekatan yang lempang atau *straight forward*, sehingga algoritma Brute Force ini dapat memecahkan masalah dengan sangat sederhana, langsung, dan jelas. Penyelesaian algoritma Brute Force biasanya didasarkan pada :

- Pernyataan masalah

Beberapa contoh kasusnya adalah mencari elemen minimum atau maksimum pada sebuah array, mencari indeks sebuah nilai pada array, dll.

- Definisi konsep yang dilibatkan

Beberapa contoh kasusnya adalah menghitung  $a^n$ , menghitung  $n!$ , mengalikan dua matriks, melakukan *test* bilangan prima, *selection sort*, *bubble sort*, mencari hasil polinom, dll.

Algoritma Brute Force memiliki beberapa karakteristik. Yang pertama, algoritma ini bukanlah algoritma yang cerdas dan mangkus atau efisien, karena algoritma ini membutuhkan jumlah langkah yang lebih besar dalam penyelesaiannya jika dibandingkan dengan algoritma lainnya, sehingga waktu yang dibutuhkan juga berbanding lurus dengan jumlah langkah penyelesaiannya. Kedua, algoritma ini juga sering digunakan sebagai basis bila membandingkan beberapa alternatif menggunakan algoritma lainnya yang lebih mangkus. Dan yang

terakhir, untuk persoalan yang kecil – kecil, kesederhanaan algoritma Brute Force lebih diperhitungkan daripada ketidakmangkusannya.

Secara sistematis, cara kerja atau langkah – langkah yang dilakukan oleh algoritma Brute Force saat menyelesaikan masalah adalah :

1. **Enumerasi** yaitu membuat list dari setiap kemungkinan solusi dengan cara yang sistematis
2. **Evaluasi** yaitu menyeleksi setiap kemungkinan solusi secara satu persatu dan menyimpan solusi terbaik yang ditemukan
3. Bila pencarian solusi berakhir, umumkan solusi terbaik

*D. Pengambilan Harga Minimum*

Seperti yang sudah dijelaskan sebelumnya, persoalan ini melibatkan *e-commerce*, basis data, dan algoritma Brute Force, dan masing – masing sudah dijelaskan di atas. Hubungan dari ketiganya adalah, pengambilan nilai minimum harga dari sebuah produk yang ditawarkan oleh *e-commerce* akan menggunakan algoritma Brute Force, dan memanfaatkan basis data milik *e-commerce*. Dari tabel basis data *e-commerce* tersebut, akan didapatkan harga – harga suatu barang yang sama berdasarkan masukan pengguna, dari toko – toko yang berbeda, seperti pada contoh tabel 2.1 di samping. Pada tabel 2.1 di samping, khususnya pada tabel barang, menampilkan harga barang “Galaxy J7” dari toko A, B, C, dan D. Dari keempat toko tersebut, dapat diketahui toko yang menawarkan produk “Galaxy J7” dengan harga paling rendah adalah toko D. Pencarian harga terendah itulah yang akan menggunakan algoritma Brute Force. Namun, untuk kasus selanjutnya akan lebih rumit karena persoalan yang akan diselesaikan menggunakan basis data yang lebih besar, yaitu basis data seluruh *e-commerce*, dan tentunya tidak akan sesederhana seperti contoh tabel 2.1, karena data yang digunakan sangat banyak.

III. PENCARIAN SOLUSI DENGAN MENGGUNAKAN ALGORITMA BRUTE FORCE

*A. Ilustrasi Awal*

Ketika membandingkan dan menemukan harga minimum dari harga – harga sebuah produk yang sama pada banyak *e-commerce*, diperlukan untuk mengakses basis data yang dimiliki oleh seluruh *e-commerce* tersebut. Atribut yang dibutuhkan dalam basis data tersebut adalah nama *e-commerce*, id *e-commerce*, nama produk, harga, nama toko, dan jumlah barang.

User dapat menemukan informasi harga sebuah produk pada banyak *e-commerce* dengan melakukan input nama produk. Dengan input nama barang tersebut, sistem akan mengidentifikasi semua *record* pada basis data seluruh *e-commerce* yang memiliki atribut “nama produk” yang sama dengan masukan *user*. Kemudian, dari seluruh *record* tersebut akan dicari atribut harga yang memiliki nilai paling minimum, sehingga solusi dari permasalahan ini menghasilkan harga paling murah dari produk yang dicari oleh pengguna, serta nama toko dan nama *e-commerce* yang memiliki produk tersebut.

Berikut adalah contoh data yang sudah digabung antar tabel – tabelnya dan digunakan untuk mencari harga minimum sebuah produk dari berbagai *e-commerce* :

Data\_Bukalapak

ID <i>e-commerce</i>	Nama <i>e-commerce</i>	Nama Produk	Harga	Nama Toko	Jumlah Stok
0001	Bukalapak	Samsung Galaxy A7	5.452.000	berlinmobile	4
0001	Bukalapak	Samsung Galaxy J2	1.775.000	berlinmobile	7
0001	Bukalapak	Iphone 8+	11.300.000	berlinmobile	5
0001	Bukalapak	Iphone X	14.840.000	berlinmobile	2
0001	Bukalapak	Samsung Galaxy A7	5.450.000	astorshop	1
0001	Bukalapak	Iphone X	16.640.000	astorshop	8
0001	Bukalapak	Iphone X	17.475.000	LitGadget	6

Data\_Tokopedia

ID <i>e-commerce</i>	Nama <i>e-commerce</i>	Nama Produk	Harga	Nama Toko	Jumlah Stok
0002	Tokopedia	Iphone X	16.750.000	EvoStore	8
0002	Tokopedia	Samsung Galaxy S8	7.290.000	EvoStore	7
0002	Tokopedia	Iphone X	14.800.000	instamishop	6
0002	Tokopedia	Samsung Galaxy J5	2.800.000	instamishop	4
0002	Tokopedia	Iphone 8 plus	14.450.000	chocoapple	9

0002	Toko pe dia	Iphone X	17.300.000	chocoap ple	2
0002	Toko pe dia	Iphone X	16.800.000	pastiasli	11

Data\_Shopee

ID e-commerce	Nama e-commerce	Nama Produk	Harga	Nama Toko	Jumlah Stok
0003	Shopee	Samsung Galaxy J7	2.660.000	Vinzmobil e	3
0003	Shopee	Iphone X	16.950.000	Vinzmobil e	7
0003	Shopee	Samsung Galaxy J7	3.425.000	dolphinphone	8
0003	Shopee	Samsung Galaxy S8	8.570.000	dolphinphone	2
0003	Shopee	Iphone X	17.200.000	dolphinphone	10
0003	Shopee	Iphone 8+	14.400.000	indigokomunika	3
0003	Shopee	Iphone 7	10.400.000	indigokomunika	4

Tabel 3.1 Basis data pada beberapa e-commerce

**B. Menerima Masukan Pencarian dari Pengguna**

Ketika seorang pengguna ingin mencari sebuah produk pada e-commerce, pengguna akan mengakses link dari e-commerce tersebut. Kemudian pada kolom pencarian, pengguna akan memasukkan kata kunci dari sebuah produk, dan sistem akan mencocokkan string masukan pengguna dengan atribut "nama produk" dengan menggunakan algoritma string matching yang tidak akan dibahas pada makalah ini.

**C. Pengambilan Data dan Pengolahannya**

Pada setiap e-commerce, pasti memiliki basis data yang memuat berbagai informasi tentang e-commerce tersebut. Basis data tersebut dapat diambil dengan melalui beberapa cara, salah satunya adalah data scrapping, sehingga basis data tersebut dapat diakses untuk kemudian diolah dan digunakan untuk mendapatkan solusi dari permasalahan yang dibahas pada makalah ini. Beberapa informasi yang disimpan di dalam basis data tersebut adalah :

a. Tabel Akun

Memuat informasi tentang akun yang dimiliki oleh e-customer dan memiliki atribut seperti nama, ID akun, gender, e-mail, dan lainnya.

b. Tabel Toko

Memuat informasi tentang toko – toko yang berada di e-commerce tersebut dan memiliki atribut seperti ID toko, nama toko, rating, nama pemilik, kontak pemilik, dan lainnya.

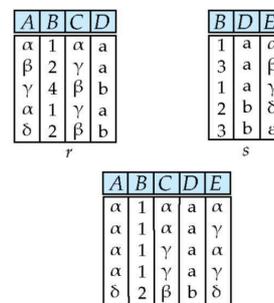
c. Tabel Transaksi

Memuat informasi tentang transaksi – transaksi jual beli yang telah dilakukan oleh e-merchant dan e-customer. Tabel ini memiliki atribut seperti ID transaksi, ID toko (ID milik e-merchant yang terlibat transaksi), ID akun (ID milik e-customer yang terlibat transaksi), waktu transaksi, dan lain – lain.

d. Tabel Produk

Memuat informasi tentang produk – produk yang ditawarkan di e-commerce tersebut dari berbagai e-merchant, dan memiliki atribut seperti ID Produk, nama produk, ID toko (ID e-merchant yang menawarkan produk tersebut), harga, jumlah stok, dan lainnya.

Tabel toko dan tabel produk tersebut akan dilakukan operasi natural join berdasarkan ID toko, sehingga setiap e-commerce hanya memiliki satu tabel data utama, seperti pada contoh Tabel 3.1, yang akan memuat informasi tentang produk – produk yang dijual di e-commerce tersebut beserta dengan nama tokonya. Tabel itulah yang akan dioperasikan sedemikian rupa hingga ditemukan solusi untuk harga termurah dari sebuah produk yang diinginkan oleh pengguna.



Gambar 3.1 Ilustrasi operasi natural join

Kemudian, setelah sistem menerima masukan string untuk kata kunci pencarian dari pengguna, sistem akan memilih record – record mana saja yang memiliki atribut nama produk sama dengan masukan user, dan semua record tersebut digabung menjadi satu tabel. Misalkan contoh dari Tabel 3.1, user memasukkan kata kunci "Iphone X", berikut adalah hasil tabel yang dihasilkan oleh sistem setelah pencarian :

ID e-commerce	Nama e-commerce	Nama Produk	Harga	Nama Toko	Jumlah Stok
0001	Bukalapak	iPhone X	14.840.000	berlinmobile	2
0001	Bukalapak	iPhone X	16.640.000	astorshop	8
0001	Bukalapak	iPhone X	17.475.000	LitGadget	6
0002	Tokopedia	iPhone X	16.750.000	EvoStore	8
0002	Tokopedia	iPhone X	14.800.000	instamishop	6
0002	Tokopedia	iPhone X	17.300.000	chocoapple	2
0002	Tokopedia	iPhone X	16.800.000	pastiasli	11
0003	Shopee	iPhone X	16.950.000	Vinzmobile	7
0003	Shopee	iPhone X	17.200.000	dolphinphone	10

Tabel 3.2 Tabel hasil pencarian

```
for a in range(1,9):
    harga = int(matriksdata[a][3])
    if harga < minharga :
        minharga = harga
        mincommerce = matriksdata[a][1]
        mintoko = matriksdata[a][4]
```

Algoritma di atas adalah algoritma untuk mencari harga termurah dari sebuah produk dengan menggunakan bahasa Python dan data matriks dengan format “.txt”, data ini pun hanya merupakan sample sebuah basis data *real* dari *e-commerce* tersebut. Algoritma di atas hanya untuk mencari harga minimum, untuk kasus khusus yang perlu menggunakan *sorting* ID dan *sorting* ID yang sudah dijelaskan sebelumnya tidak akan dibahas pada makalah ini, sehingga solusi yang dihasilkan oleh algoritma ini adalah harga termurah yang pertama kali ditemukan pada basis data. Jika sistem harus menangani kasus tersebut, maka matriks harus diolah terlebih dahulu dengan mengurutkan data berdasarkan *rating* dan ID produk. Setelah sistem memroses matriks, sistem akan menghasilkan keluaran seperti di bawah ini :

```
azkanab's mac:MAKALAH azkanab$ python cheapest.py
0001 Bukalapak iPhone_X 14840000 berlinmobile 3 2
0001 Bukalapak iPhone_X 16640000 astorshop 4 8
0001 Bukalapak iPhone_X 17475000 LitGadget 5 6
0002 Tokopedia iPhone_X 16750000 EvoStore 4.5 8
0002 Tokopedia iPhone_X 14800000 instamishop 3.5 6
0002 Tokopedia iPhone_X 17300000 chocoapple 4 2
0002 Tokopedia iPhone_X 16800000 pastiasli 5 11
0003 Shopee iPhone_X 16950000 Vinzmobile 4.5 7
0003 Shopee iPhone_X 17200000 dolphinphone 3 10

Harga termurah dari produk iPhone_X adalah 14800000 yang dimiliki oleh toko instamishop pada e-commerce Tokopedia_
```

Gambar 3.3 Bentuk output sistem

#### D. Implementasi Algoritma Pencarian Nilai Minimum dan Hasilnya

Ketika tabel basis data hasil pencarian oleh *user* sudah didapatkan, tabel tersebut akan diubah menjadi sebuah matriks agar dapat diakses oleh sistem dan dicari harga minimumnya. Matriks akan berbentuk seperti di bawah ini :

ID	e-commerce	Produk	Harga	Nama Toko	Rating	Jumlah
0001	Bukalapak	iPhone_X	14840000	berlinmobile	3	2
0001	Bukalapak	iPhone_X	16640000	astorshop	4	8
0001	Bukalapak	iPhone_X	17475000	LitGadget	5	6
0002	Tokopedia	iPhone_X	16750000	EvoStore	4.5	8
0002	Tokopedia	iPhone_X	14800000	instamishop	3.5	6
0002	Tokopedia	iPhone_X	17300000	chocoapple	4	2
0002	Tokopedia	iPhone_X	16800000	pastiasli	5	11
0003	Shopee	iPhone_X	16950000	Vinzmobile	4.5	7
0003	Shopee	iPhone_X	17200000	dolphinphone	3	10

Gambar 3.2 Matriks translasi Tabel 3.2

Kemudian akan dicari harga minimum dari produk yang sama dan *e-commerce* yang berbeda – beda berdasarkan matriks tersebut. Jika harga sama, akan diurutkan berdasarkan ID *e-commerce*, dan jika ada harga yang sama pada *e-commerce* yang sama, maka akan diurutkan berdasarkan atribut *rating*, dari *rating* terbesar hingga terkecil, dan akan diambil toko dengan *rating* yang lebih besar. Kemudian jika beberapa toko tersebut memiliki nilai atribut *rating* yang sama, akan diurutkan berdasarkan ID barang, dari ID terkecil hingga ID terbesar, di mana ID barang tersebut bersifat unik untuk setiap barang yang ada di *e-commerce* tersebut, dan akan diambil *record* yang memiliki ID barang yang paling kecil. Berikut adalah algoritma *brute force* untuk mencari nilai minimum tersebut :

```
minharga = int(matriksdata[0][3])
mincommerce = matriksdata[0][1]
mintoko = matriksdata[0][4]
```

#### IV. KESIMPULAN

Untuk mencari harga termurah sebuah produk yang sama dari banyak toko dan *e-commerce* dapat menggunakan algoritma *brute force* serta basis data yang diambil dari seluruh *e-commerce* yang ada, yang kemudian dioperasikan dan diolah menjadi matriks agar dapat diakses oleh algoritma pencarian harga minimum pada sistem. Pencarian nilai atau elemen minimum hanya dapat menggunakan algoritma *brute force*.

Kemudian, sistem ini juga dapat dikembangkan menjadi sistem yang dapat mengurutkan juga dari produk yang memiliki harga termurah hingga produk yang memiliki harga termahal.

#### UCAPAN TERIMA KASIH

Pertama – tama penulis ingin memanjatkan puji syukur kepada Tuhan yang Maha Esa karena atas rahmat dan berkah – Nya, penulis dapat menyelesaikan makalah ini dengan baik dan tepat waktu. Penulis juga ingin mengucapkan terima kasih kepada keluarga khususnya orang tua, karena tanpa dukungan mereka penulis tidak akan bisa menyelesaikan makalah ini. Terima kasih juga penulis ucapkan kepada rekan – rekan terdekat penulis yang sudah menemani penulis dan memberikan masukan kepada penulis untuk topik makalah ini. Dan terakhir, penulis berterima kasih kepada dosen pengajar mata kuliah IF2211 Strategi Algoritma, Dr. Masayu Leylia Khodra M.T., Dr. Nur Ulfa Maulidevi S.T., M.Sc., dan Dr. Ir. Rinaldi Munir, M.T., karena atas bimbingan dan penyampaian

ilmunya yang sangat membantu penulis dalam pengerjaan makalah ini.

#### REFERENSI

- [1] <https://www.kompasiana.com/novikristiadi/5992634e93be2508e06c5402/e-commerce-manfaat-dan-keuntungannya>
- [2] Munir, Rinaldi. 2004. Diktat IF2211 Strategi Algoritmik: Algoritma Brute Force.
- [3] <https://yahuuu.wordpress.com/2010/12/22/latar-belakang-e-commerce/>
- [4] <http://indokuzu.blogspot.co.id/2011/04/tahapan-tahapan-dalam-transaksi-e.html>
- [5] <https://www.kajianpustaka.com/2012/10/teori-basis-data-database.html>
- [6] <https://dosenit.com/software/dbms/fungsi-dbms-dalam-basis-data>
- [7] <http://www.webtol.com/teknologi/pengertian-database-dan-jenis-serta-struktur/>
- [8] <http://ardiansyaha.blogspot.co.id/2015/01/algoritma-brute-force.html>

#### PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 6 April 2018  
Ttd



Azka Nabilah Mumtaz  
13516013