

Oleh Kelompok Tomat

UNIVERSITAS INDONESIA
DEPOK
2020

I. Latar Belakang dan Tujuan

Di era globalisasi saat ini, akses informasi menjadi sangat mudah dengan berkembangnya teknologi digital. Perkembangan teknologi digital tidak hanya terjadi di aspek-aspek industri besar tapi juga merambah masuk hingga ke aspek-aspek kecil dalam kehidupan sehari-hari salah satunya dalam kegiatan jual beli makanan oleh pelaku Usaha Mikro Kecil Menengah (UMKM). Kegiatan jual beli secara konvensional yang biasa terjadi di kehidupan sehari-hari dapat dipermudah dengan adanya digitalisasi. Selain itu, digitalisasi dapat meningkatkan daya saing usaha jual beli pelaku UMKM di era saat ini.

Seiring dengan adanya pandemi COVID-19 ini, banyak UMKM seperti warteg / rumah makan mengalami kebingungan dalam menjalankan operasionalnya. Hal tersebut dikarenakan banyaknya aktivitas yang dilakukan menjadi serba di rumah. Belajar di rumah, kerja di rumah, mengajar di rumah dan lain sebagainya. Sehingga, warteg menjadi kehilangan konsumennya. Untuk mendukung warteg tetap bertahan pada masa pandemi ini, maka kami mencanangkan sebuah program untuk mendukung warteg tetap beroperasi secara digital. Upaya digitalisasi ini membuat warteg dapat dipesan secara digital. Rancangan aplikasi yang kami buat yaitu “warteg digital” diharapkan dapat memudahkan segala proses jual beli makanan dalam kehidupan sehari-hari.

Tujuan dari proyek ini yaitu:

1. Membantu usaha warteg agar tetap bertahan di era pandemi ini secara digital.
2. Untuk memenuhi tugas proyek 1 mata kuliah Algoritma dan Pemrograman.
3. Mengimplementasikan pembelajaran mengenai array, array sebagai tabel referensi, modular (fungsi *parameter passing*, *non parameter passing*), *user friendly*, *sorting* (*bubble sort*), *searching* (*sequential searching*), dan rekursif dalam satu program.

II. Algoritma

Pada dasarnya, dengan program ini, *user* dapat melakukan pemesanan makanan dari pilihan warteg tersedia pada program. Program akan memunculkan beberapa pilihan warteg yang berada di sekitar *user*, yang ditampilkan adalah nama warteg, jaraknya dan juga *rating*-nya. *User* dapat memfilter warteg berdasar jaraknya atau ratingnya.

Jika *user* memilih filter jarak, maka *user* dapat mencari warteg pada rentang jarak tertentu. *User* menentukan jarak terdekat dan terjauh untuk memfilter warteg berdasar jarak. Jika *user* memilih filter *rating*, maka *user* dapat memasukkan nilai *rating* tertentu dan program akan memunculkan warteg dengan rating yang diinginkan oleh *user*.

Setelah difilter, *user* dapat memilih warteg mana yang akan dipesan. Lalu, *user* dapat memesan makanan dan/atau minuman dari warteg tersebut. Selanjutnya, akan dikalkulasikan harga dari pesanan dan juga tambahan ongkos kirimnya.

Berikut ini algoritma dasar dari program dalam bentuk pseudocode.

```
Print: cetak pilihan warteg (nama warteg, jarak, rating)
Print: filter warteg berdasar (1) jarak, (2) rating
Input: pilih (angka 1 atau 2)

//searching
Jika (pilih == 1)
    //cari warteg berdasar jarak
    Input: jarak_dekat
    Input: jarak_jauh
    //mengecek pada semua pilihan warteg
    Jika (jarak dekat <= jarak warteg <= jarak jauh)
        Maka, print: nama warteg, jarak, rating

Jika (pilih == 2)
    //cari warteg berdasar rating
    Input: cari_rating
    //mengecek pada semua pilihan warteg
    Jika (rating warteg == cari_rating)
        Maka, print: nama warteg, rating, jarak

Print: list warteg setelah difilter

//memilih warteg
Input: pilih warteg yang ingin dipesan
Print: cetak list menu dan harganya dari warteg yang dipilih

//input menu
Input: pilih makanan/minuman
Print: apakah masih ingin memilih yang lain?
    Jika ya, maka input pilih makanan / minuman lagi

//kalkulasi total
Jika telah selesai memilih makanan dan minuman,
    kalkulasi total harga pesanan (harga)
    ongkir = jarak(km)*2000
    total = harga + ongkir
```

```
//cetak struk  
Tampilkan struk pesanan  
Struk pesanan di-sorting berdasar nominal harga masing2 menuanya
```

Berikut ini merupakan penjelasan dari algoritma di atas.

1. Jarak dan *rating* dari warteg akan di-generate secara *random* dengan `srand()`.
2. Menampilkan pilihan warteg yang dapat dipilih, yang ditampilkan berupa nama warteg, jaraknya, dan *rating*-nya.
3. *User* dapat memfilter warteg berdasarkan jaraknya / *rating*-nya dengan algoritma ***sequential searching*** dengan modifikasi pencarian range untuk bagian jarak.
4. Misal, *user* memfilter jarak warteg dengan *range* 2 – 5 km, maka akan dicari dari daftar warteg tersebut warteg mana yang berada dalam *range*.
5. Misal, *user* mencari warteg dengan *rating* 4, maka akan ditampilkan warteg-warteg yang mempunyai *rating* 4.
6. Setelah warteg terfilter, *user* dapat memilih warteg mana yang akan dipesan.
7. *User* memilih makanan dan/atau minuman yang ingin dipesan. Setelah itu total harga akan dikalkulasikan beserta ongkos kirimnya.
8. Lalu, program akan menampilkan struk yang nanti akan mengurutkan pesanan berdasar harga pesanan dengan algoritma ***bubble sorting***.

III. Realisasi Program

Setelah melakukan realisasi dari algoritma yang telah dirancang, didapat 2 output yang kami telah kami kerjakan. Pertama, file program utama dengan nama file 'Proyek1_Tomat.c' dan file *header* berisi fungsi-fungsi tambahan dengan nama file 'user_friendly_baru.h'.

Penjelasan dari masing-masing variabel dan fungsi pada kedua file tersebut:

- int a: untuk menyimpan angka dari input pengguna untuk fungsi pengulangan.
- int i: untuk melakukan iterasi tingkat pertama contohnya mencetak menu.
- int j: untuk melakukan iterasi tambahan maupun untuk penyimpanan nilai.
- int k = 0: digunakan untuk melakukan rekursi.
- int l: untuk melakukan iterasi tingkat kedua, misal *sorting* variabel dalam *array*
- int swap: digunakan untuk penyimpanan sementara saat *sorting*.
- int pilih_f: digunakan untuk menyimpan mode filter yang digunakan, jika pilih_f bernilai 1, maka dilakukan filter jarak, jika pilih_f bernilai 2 maka dilakukan filter rating.
- int pilih_w: digunakan untuk menyimpan angka yang merujuk pada nomer warteg yang dipilih.
- int x: banyak digunakan sebagai petunjuk element dalam *array* yang berguna untuk memenuhi *conditional statements*. Variabel x menyimpan nilai pilih_w.
- int ongkir: untuk menyimpan harga ongkos kirim dari pemesanan. Nilai ongkir ditentukan dari jarak warteg (dalam kilometer) dikalikan dengan 2000 rupiah.
- int total = 0: variabel untuk menyimpan total harga dari pesanan ditambah dengan ongkir. Pada awalnya nilai total diinisialisasi dengan 0.
- int jarak_baru[5] = {0}: untuk menyimpan *array* jarak setelah difilter.
- int rating_baru[5] = {0}: untuk menyimpan *array rating* setelah difilter.
- int f_jarak_a: variabel input jarak warteg terdekat yang diinginkan pengguna.
- int f_jarak_b: variabel input jarak warteg terjauh yang diinginkan pengguna.
- int f_rating: untuk menyimpan input *rating* warteg yang diinginkan pengguna
- int ada: untuk menyimpan nilai apabila hasil *searching* ditemukan.
- int fharga[20]: untuk menyimpan harga dari menu yang dipilih oleh pengguna.
- int jumlah[100]: untuk menyimpan jumlah suatu menu sesuai pesanan pengguna
- int random_rating(int rat[5]): fungsi untuk meng-*generate* secara *random rating* dari warteg. *Rating* dihasilkan dari angka acak dengan rentang 1 sampai 5. Terjadi *passing parameter* dari variabel rating[5] ke rat[5].
- int random_jarak(int jar[5]): fungsi untuk meng-*generate* secara *random* jarak dari warteg. Jarak dihasilkan dari angka acak dengan rentang 1 sampai 10 (dalam satuan kilometer). Terjadi *passing parameter* dari variabel jarak[5] ke jar[5].
- int print_list_warteg(char war[5][30], int jar[5], int rat[5]): fungsi untuk mencetak tabel referensi seperti di bawah ini. Array warteg[5][30], jarak[5], dan rating[5] di-passing ke parameter war[5][30], jar[5], dan rat[5].

char war[5][30]	int jar[5]	rat[5]
Nama Warteg	Jarak warteg (km)	Rating warteg

- int pilih_filter(char war[5][30], int jar[5], int rat[5]): fungsi bagi pengguna untuk memilih jenis filter yang akan digunakan untuk memilah-milah warteg yang diinginkan.
- int filter_jarak(char war2[5][30], int jar2[5], int rat2[5]): fungsi untuk mencetak tabel referensi sesuai filter jarak yang diinginkan. Pada fungsi ini *user* menginput jarak terdekat dan terjauh lalu fungsi ini akan memfilter warteg berdasar range jarak tersebut.

char war2[5][30]	int jar2[5]	rat2[5]
Nama Warteg yang memenuhi kondisi filter	Jarak warteg (km) yang memenuhi kondisi filter	Rating warteg yang bersangkutan

- int filter_rating(char war2[5][30], int jar2[5], int rat2[5]): fungsi untuk mencetak tabel referensi sesuai filter rating yang diinginkan. Pada fungsi ini *user* menginput *rating* yang diinginkan lalu fungsi ini akan memfilter warteg berdasar *rating* tersebut.

char war2[5][30]	int jar2[5]	rat2[5]
Nama Warteg yang memenuhi kondisi filter	Jarak warteg (km) yang bersangkutan	Rating warteg yang memenuhi kondisi filter

- int pilih_warteg(): fungsi bagi pengguna untuk memilih warteg yang diinginkan.
- int print_menu_bahari(): fungsi untuk mencetak menu dari warteg bahari, memesan makanan dan/atau minuman dari warteg bahari, dan mencetak struk pesanan dengan harga pesanan yang telah di-*sorting*.
- int print_menu_shinta(): fungsi untuk mencetak menu dari warteg shinta, memesan makanan dan/atau minuman dari warteg shinta, dan mencetak struk pesanan dengan harga pesanan yang telah di-*sorting*.
- int print_menu_samudra(): fungsi untuk mencetak menu dari warteg samudra, memesan makanan dan/atau minuman dari warteg samudra, dan mencetak struk pesanan dengan harga pesanan yang telah di-*sorting*.

- `int print_menu_tegal()`: fungsi untuk mencetak menu dari warteg tegal, memesan makanan dan/atau minuman dari warteg tegal, dan mencetak struk pesanan dengan harga pesanan yang telah di-*sorting*.
- `int print_menu_makmur()`: fungsi untuk mencetak menu dari warteg makmur, memesan makanan dan/atau minuman dari warteg makmur, dan mencetak struk pesanan dengan harga pesanan yang telah di-*sorting*.
- `char menu[20][100]` dan `int harga[20]` sebagai array tabel referensi.

i	char menu[20][100]	int harga[20]	Kepemilikan
0	ayam goreng	7000	warteg bahari
1	telur balado	3000	warteg bahari
2	sop jamur	4000	warteg bahari
3	es jeruk	4500	warteg bahari
4	ikan tongkol	6000	warteg shinta
5	telur ceplok	4000	warteg shinta
6	sop tahu	3000	warteg shinta
7	es teh	2000	warteg shinta
8	rendang	10000	warteg samudra
9	telur dadar	5500	warteg samudra
10	sayur nangka	2500	warteg samudra
11	jus mangga	8000	warteg samudra
12	ayam balado	9500	warteg tegal
13	tempe orek	2000	warteg tegal
14	kangkung	2000	warteg tegal
15	air putih	0	wartag tegal
16	sate	15000	warteg makmur
17	tahu	2000	warteg makmur
18	tongseng	13000	warteg makmur
19	boba	20000	warteg makmur

- `char fmenu[20][100]`: *array* untuk menyimpan menu yang telah dipilih.
- `char temp[100]`: untuk penyimpanan sementara saat pertukaran `char fmenu`.

- `char warteg_baru[5][30]`: *array* untuk menyimpan nama-nama warteg yang ada (bahari, shinta, samudra, tegal, dan makmur).
- `int pilihAwal`: variabel yang akan di-*passing* ke `print_header()` pada *library* `userFriendly.h`, lalu variabel tersebut akan menyimpan pilihan awal dalam bentuk angka. Lalu akan diteruskan ke fungsi `pilih_menu()` pada *library* `userFriendly.h`.
- `char warteg[5][30] = {"bahari_", "shinta_", "samudra", "tegal_", "makmur_"}`: men-*declare* nama-nama warteg. Ditambahkan *underscore* agar saat mencetak tabelnya tidak berantakan.
- `int jarak[5]`: *array* untuk menyimpan jarak dari warteg. Hasil dari randomisasi pada fungsi `random_jarak(int jar[5])` akan disimpan pada *array* ini.
- `int rating[5]`: *array* untuk menyimpan *rating* dari warteg. Hasil dari randomisasi pada fungsi `random_rating(int rat[5])` akan disimpan pada *array* ini.
- `int pembuka()`: fungsi yang berisikan fungsi-fungsi dasar pada program warteg digital. fungsi `pembuka()` dibuat agar pengguna dapat melakukan pengulangan program melalui `pembuka()` tanpa harus mengulang dari fungsi `main()`. Hal tersebut dilakukan untuk memudahkan pengguna dan mencegah terjadinya pengacakan jarak dan rating oleh program.

Saat program di-*run*, maka otomatis akan masuk ke fungsi '`main()`'. Lalu, terjadi *parameter passing* '`pilihAwal`' ke parameter '`a`' pada fungsi '`print_header(int a)`' pada file *header* '`user_friendly_baru.h`'. Pada fungsi '`print_header`' tersebut akan mencetak tampilan awal dari program dan juga meminta input berupa angka untuk memilih pilihan yang tersedia pada awal program. Fungsi '`print_header()`' mengembalikan nilai parameter '`a`' ke parameter '`pilihAwal`'.

Selanjutnya, '`pilihAwal`' di-*passing* ke parameter '`a`' pada fungsi '`pilih_menu(int a)`' yang juga berada pada file *header* '`user_friendly_baru.h`'. fungsi `pilih_menu` ini bertujuan untuk mengecek nilai '`pilihAwal`', Jika '`pilihAwal`' (atau '`a`') berupa angka 1 maka akan lanjut menjalankan program. Jika input berupa angka 2 maka akan memunculkan bantuan / *help* melalui fungsi '`bantuan()`' yang ada pada file *header* '`user_friendly_baru.h`', dan setelah itu melanjutkan program. Jika input selain angka 1 dan 2 maka akan menutup program.

Setelah itu, dilanjutkan dengan menjalankan fungsi '`random_rating(int rat[5])`' dengan mem-*passing* parameter `rating[5]` ke parameter `rat[5]` dan menjalankan fungsi '`random_jarak(int jar[5])`' dengan mem-*passing* parameter `jarak[5]` ke parameter `jar[5]`. Setelah randomisasi selesai, maka kembali ke '`main()`'.

Lalu, dilanjutkan ke fungsi '`pembuka()`'. Fungsi '`pembuka()`' ini merupakan fungsi penanda dimulainya program, karena jika terjadi pengulangan pada akhir program, maka akan kembali ke fungsi `pembuka()`. Pada fungsi '`pembuka()`' ini terdapat proses dengan urutan sebagai berikut:

- menjalankan fungsi `'print_list_warteg(char war[5][30], int jar[5], int rat[5])'` dengan mempassing parameter `'warteg[5][30]'`, `'jarak[5]'` dan `'rating[5]'` ke parameter pada fungsi `'print_list_warteg'` tersebut. Pada fungsi ini akan mencetak pilihan warteg yang dapat dipilih oleh user. Yang dicetak berupa nama warteg, jaraknya & juga *rating*-nya.
- menjalankan fungsi `'pilih_filter(char war[5][30], int jar[5], int rat[5])'` dengan *mem-passing* parameter warteg, jarak, dan rating ke fungsi tersebut. Pada fungsi ini akan diminta input variabel `'pilih_f'`, jika nilai `'pilih_f'` sama dengan 1 maka akan dilanjut dengan menjalankan `'fungsi filter_jarak'`. Jika `'pilih_f'` sama dengan 2 maka akan dilanjut dengan menjalankan `'fungsi filter_rating'`. Jika nilainya tidak sama dengan 1 atau 2, maka akan merekursi / memanggil fungsi `'pilih_filter'` kembali.
 - a. `'filter_jarak(char war2[5][30], int jar2[5], int rat2[5])'`, *mem-passing* parameter `'war[5][30]'`, `'jar[5]'` dan `'rat[5]'` ke parameter `'war2[5][30]'`, `'jar2[5]'` dan `'rat2[5]'`. Setelah itu diminta input jarak terdekat (`f_jarak_a`) dan input jarak terjauh (`f_jarak_b`). Lalu, dilakukan iterasi untuk mengecek apakah jarak warteg berada pada rentang jarak tersebut atau tidak. Jika ya, maka variabel `'ada'` nilainya ditambah dengan 1, jika tidak, maka nilainya tetap 0. Jika ditemukan warteg yang berada pada *range* tersebut, maka *array* `'war2'`, `'jar2'`, `'rat2'` pada iterasi tersebut dimasukkan ke `'warteg_baru'`, `'jarak_baru'` dan `'rating_baru'`. Jika iterasi telah berakhir, maka dilanjut dengan pengecekan apakah `'ada = 0'`? jika ya, maka program akan mencetak "warteg dengan range jarak dari (`f_jarak_a`) km sampai (`f_jarak_b`) km tidak ditemukan". Lalu, mengulang kembali fungsi filter jarak. Jika `'ada'` tidak sama dengan 0, maka selanjutnya akan dicetak *list* warteg yang telah difilter dengan menggunakan *array* `'warteg_baru'`, `'jarak_baru'` dan `'rating_baru'`.
 - b. `'filter_rating(char war2[5][30], int jar2[5], int rat2[5])'`, *mem-passing* parameter `'war[5][30]'`, `'jar[5]'` dan `'rat[5]'` ke parameter `'war2[5][30]'`, `'jar2[5]'` dan `'rat2[5]'`. Setelah itu diminta input *rating* yang diinginkan *user* (`f_rating`). Lalu, dilakukan iterasi untuk mencari warteg yang memiliki *rating* sama dengan `'f_rating'`. Jika ya, maka variabel `'ada'` nilainya ditambah dengan 1, jika tidak, maka nilainya tetap 0. Jika ditemukan warteg yang mempunyai *rating* tersebut, maka *array* `'war2'`, `'jar2'`, `'rat2'` pada iterasi tersebut dimasukkan ke `'warteg_baru'`, `'jarak_baru'` dan `'rating_baru'`. Jika iterasi telah berakhir, maka dilanjut dengan pengecekan apakah `'ada = 0'`? jika ya, maka program akan mencetak "warteg dengan *rating* (`f_rating`) tidak ditemukan". Lalu, mengulang kembali fungsi filter jarak. Jika `'ada'` tidak sama dengan 0, maka selanjutnya akan dicetak *list* warteg yang telah difilter dengan menggunakan *array* `'warteg_baru'`, `'jarak_baru'` dan `'rating_baru'`.
- menjalankan fungsi `'pilih_warteg()'`. Pada fungsi ini user diminta input yaitu memilih warteg yang ingin dipesan dari pilihan warteg yang telah difilter

sebelumnya. Input tersebut lalu masuk ke variabel 'pilih_w'. Lalu, dideklarasikan ulang dengan variabel 'x' dimana 'x = pilih_w-1'. Pengurangan dengan angka 1 tersebut bertujuan untuk penunjukan indeks yang lebih tepat. Hal tersebut dikarenakan program mencetak warteg dengan penomoran dimulai dari 1, sedangkan indeks *array* dimulai dari 0. Sehingga ketika user memilih warteg 1, maka sebenarnya user memilih warteg pada indeks 0. Jika *user* memilih warteg yang tidak ada pada daftar warteg terfilter, maka akan mengulang fungsi 'pilih_warteg()' kembali. Ketika *user* sudah memilih warteg yang ingin dipesan, maka akan dikalkulasi ongkos kirimnya dengan variabel 'ongkir = 2000*jarak_baru[x]'. Lalu, dilanjutkan dengan menjalankan fungsi 'print_menu_(nama warteg)' sesuai dengan warteg yang dipilih oleh *user*. Misal *user* memilih warteg bahari, maka akan dilanjutkan dengan menjalankan fungsi 'print_menu_bahari()'.

- a. fungsi 'print_menu_(nama warteg())', semua fungsi untuk print menu ini memiliki kondisi yang sama, dimana akan mencetak menu dari warteg tersebut. Untuk menu bahari, maka akan dicetak menu dari *array* 'menu[]' dan 'harga[]' dengan indeks dari 0 sampai 3 (berdasar tabel referensi menu dan harga). Untuk menu shinta dicetak dari indeks 4 sampai 7. Untuk menu samudra dicetak dari indeks 8 sampai 11. Untuk menu tegal dicetak dari indeks 12 sampai 15. Untuk menu makmur dicetak dari indeks 16 sampai 19. Setelah itu, *user* memilih menu yang ingin dipesan. Jika *user* memilih menu yang tidak ada pada daftar menu tersebut maka akan ada pesan error dan program akan menjalankan kembali fungsi 'print_menu_(nama warteg())'.
 - b. Pada fungsi 'print_menu_(nama warteg())', *user* akan diminta untuk memilih menu yang diinginkan sesuai dengan menu yang disediakan dan jumlah pemesanannya. Menu yang dipesan oleh *user* akan disalin ke dalam *array* 'fmenu[]' sementara harga menu tersebut akan disalin ke dalam *array* 'fharga[]'. Jika sudah, *user* dapat memesan kembali menu lain yang disediakan maupun langsung masuk ke pembayaran.
 - c. Pada fungsi yang sama apabila *user* memilih untuk masuk ke pembayaran maka program akan mencetak struk yang berisikan menu menu yang dipesan, jumlah pemesanan masing masing menu, harga satuan, harga total, ongkos kirim, dan total biaya yang harus dibayarkan.
- Menjalankan fungsi 'ulang()'. Pada tahap ini *user* telah berhasil memesan makanan. Selanjutnya *user* dapat memilih untuk memesan makanan lagi dari warteg lain ataupun mengakhiri program. Apabila *user* memutuskan untuk melakukan pemesanan kembali, maka fungsi 'ulang()' akan mereset variabel 'k' agar pengulangannya stabil, variabel 'total' agar harga totalnya tidak dijumlahkan dengan harga total sebelumnya, variabel 'ada' agar kondisi pencarian di-reset, *array* 'jarak_baru[]' dan *array* 'rating_baru[]' agar hasil filternya tidak dipengaruhi oleh hasil pemfilteran sebelumnya.

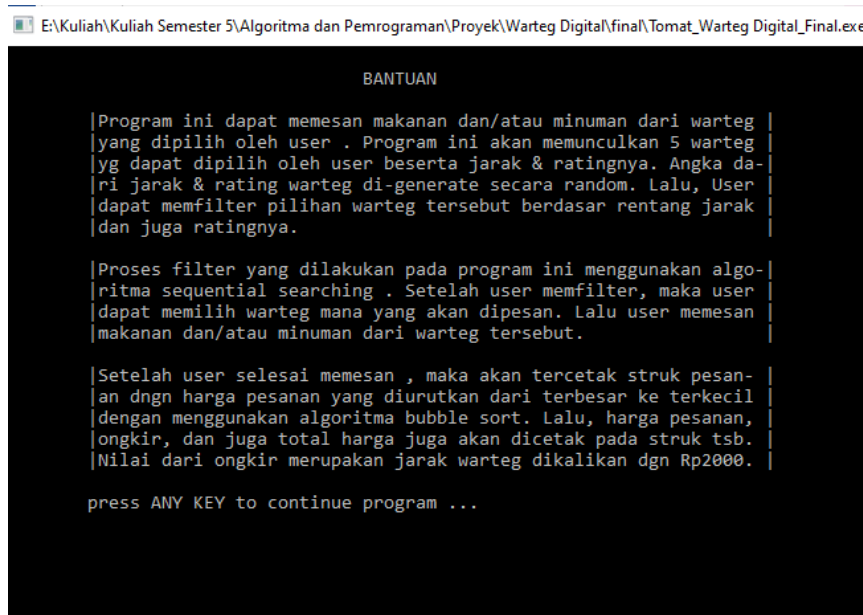
IV. Print Screen Output Program

Berikut ini tampilan ketika menjalankan program.

1. Program dijalankan.



2. Untuk menampilkan bantuan pengguna dapat memberikan input 2, setelah itu akan dilanjutkan dengan menjalankan program.



3. Untuk menjalankan program pengguna dapat memberi input 1.

```

-----
Warteg yang ada di sekitarmu:

|no|warteg      |jarak(km)|rating|
|--|-----|-----|-----|
|1|bahari      |2        |1     |
|2|shinta      |4        |5     |
|3|samudra    |7        |3     |
|4|tegal      |3        |5     |
|5|makmur     |2        |3     |

filter warteg berdasar:
(1) jarak
(2) rating

pilih:

```

4. Apabila pengguna memilih filter berdasarkan jarak maka pengguna akan diminta untuk menginput jarak yang diinginkan.

```

-----
Warteg yang ada di sekitarmu:

|no|warteg      |jarak(km)|rating|
|--|-----|-----|-----|
|1|bahari      |2        |1     |
|2|shinta      |4        |5     |
|3|samudra    |7        |3     |
|4|tegal      |3        |5     |
|5|makmur     |2        |3     |

filter warteg berdasar:
(1) jarak
(2) rating

pilih: 1
-----

input jarak terdekat : 3
input jarak terjauh  : 5

```

maka akan muncul nama nama warteg yang memenuhi kriteria filter yang diinginkan. user dapat memilih warteg yang diinginkan dengan memberikan input nomor wartegnya.

```

List warteg setelah difilter:

|no|warteg      |jarak(km)|rating|
|--|-----|-----|-----|
|2|shinta      |4        |5     |
|4|tegal      |3        |5     |

pilih warteg yang ingin dipesan:

```

5. Jika pengguna memberi input 2. maka program akan menampilkan menu warteg “Shinta”.

- ```

----- MENU SHINTA -----
menu shinta: harga:
1. ikan tongkol 6000
2. telur ceplok 4000
3. sop tahu 3000
4. es teh 2000

Pilih menu yang ingin dipesan : 1
Jumlah pemesanan : 2

1. Pesan menu lain
2. Masuk ke Pembayaran
Masukan pilihan :

```

```

menu shinta: harga:
1. ikan tongkol 6000
2. telur ceplok 4000
3. sop tahu 3000
4. es teh 2000

Pilih menu yang ingin dipesan : 3
Jumlah pemesanan : 1

1. Pesan menu lain
2. Masuk ke Pembayaran
Masukan pilihan : 1

menu shinta: harga:
1. ikan tongkol 6000
2. telur ceplok 4000
3. sop tahu 3000
4. es teh 2000

Pilih menu yang ingin dipesan : 4
Jumlah pemesanan : 1

1. Pesan menu lain
2. Masuk ke Pembayaran
Masukan pilihan :

```

7. Masuk ke pembayaran (input 2) jika pemesanan sudah selesai, dengan begitu program akan menampilkan struk pembayaran dengan kondisi harga pesanan telah di-sorting dan ongkos kirim telah dihitung (jarak warteg shinta = 4 km \* 2000 rupiah = 8000 rupiah).

E:\Kuliah\Kuliah Semester 5\Algoritma dan Pemrograman\Proyek\Warteg Digital\final\Tomat\_Warteg Digital\_Fi

```

Cetak struk:

 WARTeg SHINTA
=====
ikan tongkol 2 6000 12000
sop tahu 1 3000 3000
es teh 1 2000 2000

=====
Total Pesanan 17000
Ongkos Kirim 8000
=====
Total 25000
=====

Program telah selesai
1. Lakukan pembayaran dan memesan lagi
2. Lakukan pembayaran dan tutup program
Pilih:
```

8. Pengguna dapat memilih untuk selesaikan pembayaran dan memesan makanan dari warteg lain dengan memberikan input 1.

E:\Kuliah\Kuliah Semester 5\Algoritma dan Pemrograman\Proyek\Warteg Digital\final\Tomat\_Warteg Digital\_Final.exe

```

Warteg yang ada di sekitarmu:

no	warteg	jarak(km)	rating
1	bahari	2	1
2	shinta	4	5
3	samudra	7	3
4	tegal	3	5
5	makmur	2	3

filter warteg berdasar:
(1) jarak
(2) rating

pilih:
```

9. Apabila pengguna memilih filter berdasarkan rating maka pengguna akan diminta untuk menginputkan rating yang diinginkan.

E:\Kuliah\Kuliah Semester 5\Algoritma dan Pemrograman\Proyek\Warteg Digital\final\Tomat\_Warteg Digital\_Final.exe

```

Warteg yang ada di sekitarmu:

no	warteg	jarak(km)	rating
1	bahari	2	1
2	shinta	4	5
3	samudra	7	3
4	tegal	3	5
5	makmur	2	3

filter warteg berdasar:
(1) jarak
(2) rating

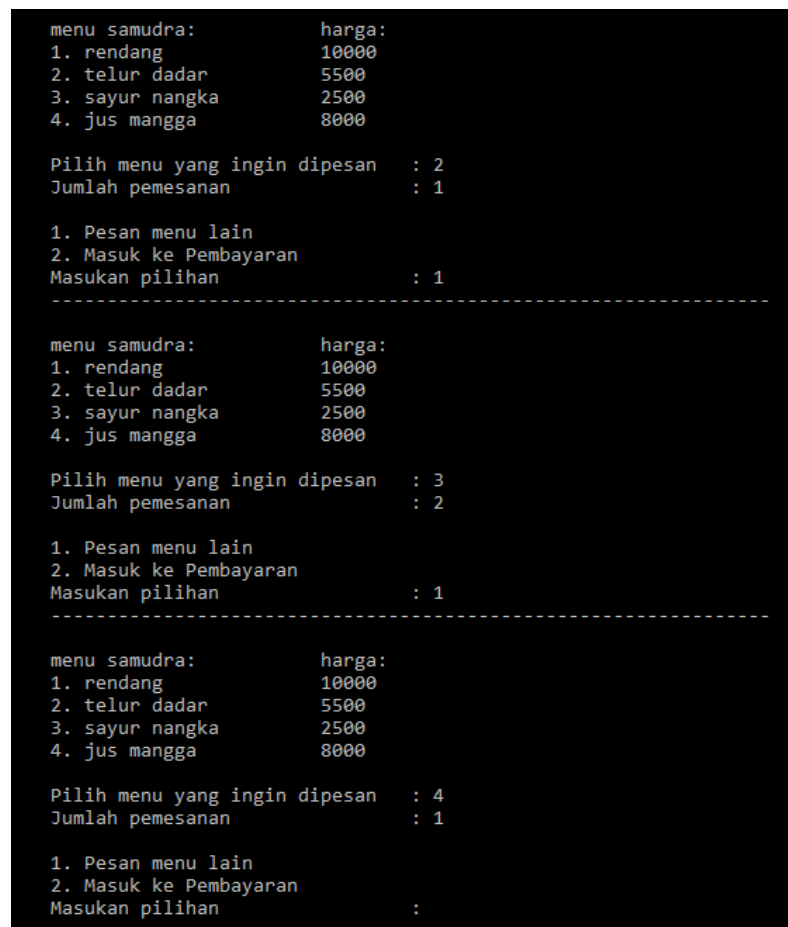
pilih: 2

input rating warteg yang ingin dicari: 3
```

E:\Kuliah\Kuliah Semester 5\Algoritma dan Pemrograman\Proyek\Warteg Digital\final\Tomat\_Warteg Digital\_Final.exe

Lalu, misal user memilih warteg samudra dan ingin memesan 1 telur dadar, 2 sayur nangka dan 1 jus mangga.

E:\Kuliah\Kuliah Semester 5\Algoritma dan Pemrograman\Proyek\Warteg Digital\final\Tomat\_Warteg Digital\_Final.exe



Lalu, melakukan pembayaran dari total harga pesanan dan juga ongkir (jarak warteg samudra 7 km \* 2000 rupiah = 14000 rupiah).

```
E:\Kuliah\Kuliah Semester 5\Algoritma dan Pemrograman\Proyek\Warteg Digital\final\Tomat_Warteg Digital_Final.exe

Cetak struk:

 WARTEG SAMUDRA
=====
jus mangga 1 8000 8000
telur dadar 1 5500 5500
sayur nangka 2 2500 5000
=====
Total Pesanan 18500
Ongkos Kirim 14000
=====
Total 32500
=====

Program telah selesai
1. Lakukan pembayaran dan memesan lagi
2. Lakukan pembayaran dan tutup program
Pilih:
```

11. Setelah selesai memesan, user dapat menyelesaikan pembayaran dan menutup program dengan memberi input 2.