# KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa atas kehendak-Nya Laporan Proyek Kasir Restoran ini dapat terselesaikan. Penulisan karya ini bertujuan untuk menyelesaikan proyek yang telah diberikan oleh Bapak Adnan Rachmat Anom Besari di semester 3 program D4 Teknik Komputer Politeknik Elektronika Negeri Surabaya. Selain untuk menyelesaikan proyek, penulis berkeinginan memperdalam ilmu mengenai penggunaan Java dalam aplikasi.

Dalam penyelesaian laporan ini, penulis banyak mengalami kesulitan terutama disebabkan akan kurangnya pengetahuan mengenai program yang berjalan pada sebuah applikasi. Namun, berkat bimbingan dari berbagai pihak akhirnya laporan ini dapat terselesaikan. Penulis mengucapkan terima kasih kepada :

* Allah SWT
* Bapak Adnan Rachmat Anom Besari selaku Dosen Politeknik Elektronika Negeri Surabaya.

Penulis menyadari laporan ini masih banyak memiliki kekurangan. Oleh karena itu, diharapkan adanya masukan berupa saran dan kritik yang dapat membangun demi kesempurnaan laporan proyek ini. Demikian besar harapan penulis agar laporan proyek ini dapat bermanfaat bagi pembaca, khususnya dalam mempelajari program Java dalam aplikasi.

Surabaya, 07 November 2015

Penulis,

# DAFTAR ISI

[KATA PENGANTAR i](#_Toc434826757)

[DAFTAR ISI ii](#_Toc434826758)

[BAB I 1](#_Toc434826759)

[PENDAHULUAN 1](#_Toc434826760)

[1.1 Latar Belakang 1](#_Toc434826761)

[1.2 Rumusan Masalah 1](#_Toc434826762)

[1.3 Tujuan Penulisan 1](#_Toc434826763)

[BAB II 2](#_Toc434826764)

[TINJAUAN PUSTAKA 2](#_Toc434826765)

[*2.1* *ArrayList* 2](#_Toc434826766)

[*2.2* *Class(Java)* 3](#_Toc434826767)

[*2.3* *Objects(Java)* 4](#_Toc434826768)

[BAB III 6](#_Toc434826769)

[Cara Kerja Program 6](#_Toc434826770)

[*3.1* *UML* 7](#_Toc434826771)

[BAB IV SOURCE CODE & SCREEN SHOOT 8](#_Toc434826772)

[BAB V PENUTUP 19](#_Toc434826773)

[5.1 Analisa 19](#_Toc434826774)

[5.2 Kesimpulan 19](#_Toc434826775)

# 

# BAB I

# PENDAHULUAN

## Latar Belakang

Dewasa ini komputer telah masuk ke segala bidang industri, mulai dari persewaan video, pemesanan tiket sampai restoran juga memakai komputer untuk mengatur pesanan pelanggannya. Hal ini bisa terjadi karena penggunaan komputer lembih efektif daripada menghitung dan mencatat secara manual, sedangkan komputer menyelesaikan segala proses dalam hitungan mili sekon.

Salah satu pengaplikasian komputer pada bidang industri adalah pemakaian komputer untuk kasir restoran. Terdapat sebuah program kasir dalam komputer-komputer kasir dan saling terhubung ke sebuah server database untuk menyimpan data. Namun untuk sekarang, yang akan kami gunakan hanya sebatas program kasir saja dan database dalam bentuk file txt.

## Rumusan Masalah

1. Apa guna “Array List” dalam Java?
2. Bagaimana membuat applikasipada Java?
3. Bagaimana membuat applikasidari *Array List* dalam Java?

## Tujuan Penulisan

Tujuan penulisan laporan ini adalah

1. Dapat mengetahui kegunaan dari Array List dalam menghandle Object
2. Dapat membuat serangkaian Object yang bersifat Manageable atau dapat diatur sesuai kebutuhan
3. Dapat membuat sebuah appilkasi berbasis Java dengan Array List sebagai pengatur Object, serta mampu menerima dan menuliskan Data pada sebuah file berekstensi txt.

# BAB II

# TINJAUAN PUSTAKA

Program kasir adalah sebuah program yang dibuat bedasarkan penggunaan Array List yang digunakan untuk memanage beberapa objek yang masuk ke dalam kategori tertentu. Berikut adalah beberapa penjabaran teori yang kami pakai dalam program ini :

## *ArrayList*

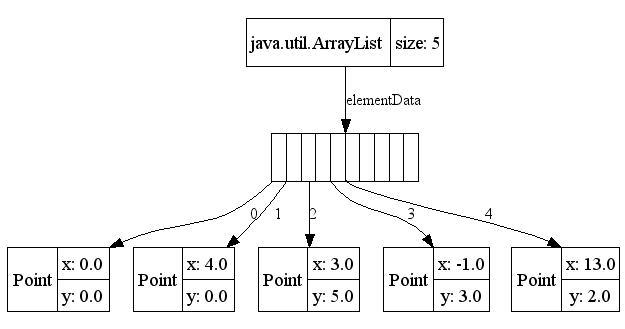
ArrayList adalah sebuah kelas yang dapat melakukan penyimpanan data berupa list objek berbentuk array yang ukurannya dapat berubah secara dinamis sesuai dengan jumlah data yang dimasukkan.

Perbedaan paling mendasar antara Array dan ArrayList adalah:

* Untuk menyimpan data dalam array biasa, maka harus mendeklarasikan jumlah elemen maksimal yang bisa menampung. Dengan kata lain jika jumlah datanya fleksibel, maka array tidak bisa digunakan.
* ArrayList dapat menampung sejumlah data secara dinamis, sehingga seberapapun jumlahnya akan ditampung oleh ArrayList tanpa memperhatikan berapa jumlah maksimal elemen yang dapat ditampung.

ArrayList digunakan dalam menyimpan data dalam bentuk objek, sehingga untuk menyimpan data didalam ArrayList maka, buatlah sebuah kelas yang kemudian dijadikan objek yang dapat menyimpan data. ArrayList terdapat pada kelas java.util, sehingga untuk menggunakan ArrayList, maka harus melakukan import java.util.

Berikut adalah skema cara kerja ArrayList :



Gambar diatas menunjukkan bahwa size atau ukuran banyaknya data yang ditampung adalah 5 karena data yang diinputkan ada 5 data, jika ditambahkan data lagi, maka size ArrayList akan berubah secara dinamis sesuai jumlah data. Namun, jumlah data bisa diatasi dengan mengkosongkan jumlah ketika mendeklarasikan ArrayList.

ArrayList dapat menyimpan sekumpulan data yang disimpan dalam satu-kesatuan. Misalkan: menyimpan data mahasiswa berupa NIM, Nama, dan Alamat, maka data tersebut akan disimpan dalam satu-kesatuan array biarpun data tersebut memiliki tipe data berbeda. Berarti ArrayList tersebut menyimpan 3 data variabel yang berbeda dalam satu elemen array.

## *Class(Java)*

Kelas dapat didefinisikan sebagai cetak biru (blueprint) atau prototipe/kerangka yang mendefinisikan variabel-variabel (data) dan method-method (perilaku) umum dari sebuah objek tertentu.

Kelas digunakan untuk mendeklarasikan sebuah variabel yang berupa objek, dan sering disebut dengan referensi objek (objek reference).

Sebagai contoh objek mobil :

* Attribute (State) -> Variable (Member) :
  + Ban, Stir, Pedal Rem, Pedal Gas, Warna, Tahun Produksi dll.
* Behavior -> Method (Fungsi) :
  + Cara menghidupkan mesin,
  + Cara Menjalankan Mobil,
  + Cara Memundurkan mobil.

Perlu diperhatikan bahwa kelas masih bersifat abstrak. Dengan kata lain kelas adalah pola (template) untuk pembuatan objek, dan objek adalah wujud nyata (instance) dari sebuah kelas.

Perbedaan antara Class dan Objeknya adalah Class merupakan konsep dan deskripsi dari suatu objek, sedangkan object merupakan wujud nyata (Instance) dari Class.

Class dapat didefinisikan dengan format :

class NamaKelas {

tipe data1;

tipe data2;

… tipe dataN;

tipe method1(daftar-parameter) {

// kode untuk method1

}

tipe method2(daftar-parameter) {

// kode untuk method2

}

… tipe methodN(daftar-parameter) {

// kode untuk methodN

}

}

Data atau variabel yang didefinisikan di dalam sebuah kelas sering pula disebut dengan instance variable. Nilai dari data-data tersebut selanjutnya akan diakses melalui method method yang ada.

Dengan demikian, method sebenarnya digunakan sebagai antar muka (interface) antara user (pemakai kelas) dan data-data yang terdapat di dalam kelas bersangkutan. Data dan method yang tergabung dalam suatu kelas sering dinamakan dengan class members (anggota-anggota kelas).

Perlu sekali untuk diketahui bahwa setiap objek atau instance dari suatu kelas akan memiliki salinan data sendiri-sendiri.  
Ini artinya, antara objek yang satu dengan yang lainnya, nilai data datanya bisa saja berbeda.

## *Objects(Java)*

Pada saat kita mendefinisikan suatu kelas, kita hanya membuat tipe data baru, bukan membuat objek baru. Untuk membuat objek baru dari tipe kelas yang telah didefinisikan, secara eksplisit kita perlu melakukan dua tahap. Pertama, kita perlu mendeklarasikan variabel yang digunakan sebagai referensi ke objek dari kelas bersangkutan. Kedua, kita perlu menginstansiasi kelas dengan menggunakan operator new dan memasukkan instance-nya ke dalam variabel referensi yang baru saja dideklarasikan.

Berikut ini ilustrasi dua tahap proses tersebut :

// Mendeklarasikan variabel k bertipe Kotak

Kotak k;

// Melakukan instansiasi dan memasukkan referensi ke variabel k  
k = new Kotak();

Baris pertama (Tahap I) berguna untuk mendeklarasikan variabel k yang berperan sebagai referensi ke objek dari tipe Kotak. Pada tahap ini, variabel k akan bernilai null, yang berarti referensi tersebut belum  
menunjuk ke objek aktual manapun. Segala bentuk usaha untuk menggunakan variabel k disaat seperti ini akan menyebabkan kesalahan pada saat proses kompilasi program. Alasannya adalah karene variabel k masih bernilai null.  
Selanjutnya baris kedua (Tahap II) digunakan untuk mengalokasikan sebuah objek baru dan memasukkan referensinya ke dalam variabel k.

Pada prakteknya, dua tahap di atas biasanya ditulis dalam satu baris, seperti berikut :

Kotak k = new Kotak();

# BAB III

# Cara Kerja Program

Program ini dimulai dengan menampakan menu login karyawan. Karena ini adalah simulasi dari kasir Restoran dan untuk mempermudah penjelasan, login hanya perlu memasukan nomer ID saja, tidak perlu memasukan Password.

Ketika ID sudah terverifikasi, akan muncul menu utama yang terdiri atas : 1) Cek Ketersediaan Meja, 2)Atur Pesan Makan, 3)Atur Reservasi, 4)Atur Promosi, 5)Atur Item, 6)Lihat Data penjualan dan 7)Keluar.

Program ini menggunakan ArrayList sebagai basisnya untuk menyimpan 3 objek database, yakni meja, pesanan dan menu. Pelanggan akan memesan sebuah meja untuk makan dengan jumlah kursi tertentu, barulah bisa memesan makanan. Langkah yang harus dilakukan kasir adalah menekan menu pertama, memilih meja, kemudian memilih menu kedua untuk memesan makanan.

Apabila hendak mengatur pemesanan sebuah meja lebih awal, dapat memakai menu ketiga, yakni Reservasi. Setelah memakai menu ini, meja yang disimpan oleh Reservasi tidak akan bisa dipilih oleh menu pertama.

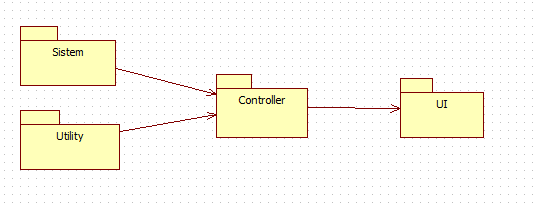
Menu keempat dan kelima mengatur promosi dan item menu yang ada. Promosi sedikit berbeda dengan menu karena terdiri atas beberapa item dan harga “spesial”, sedangkan item menu adalah hidangan makanan, minuman atau dessert yang tersedia untuk dipesan. Dalam promosi dan atur item sendiri terdapat berbagai menu untuk menambah, menghapus atau mengubah salah satu item.

Apabila pelanggan selesai menyantap makanan, kembali ke menu kedua, pilih sub menu keempat, “Bayar Makanan” untuk mencetak Nota pembayaran.

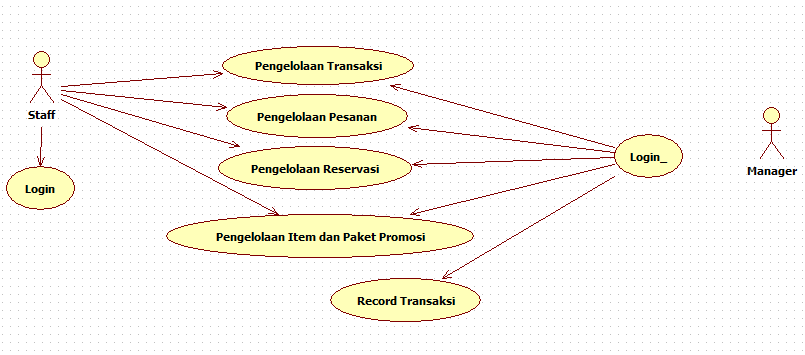
Ada satu menu terakhir di menu utama, yakni “Lihat Data Penjualan”. Menu ini hanya bisa diaksess oleh Admin saja, yakni staff yang berpangkat “Manager”. Dalam simulasi ini, pangkat manager adalah staff ID 2. Jika dicoba masuk dengan profil staff biasa, maka akan ditolak oleh program.

## *UML*

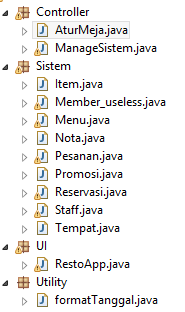
Berikut adalah serangkaian gambar yang menggambarkan rangkaian UML dari setiap Package :



Class-class pada sistem dan Utility dikendalikan oleh class-class di dalam Package Controller. Packaged UI mengatur interaksi antara komputer dan pemakainya, dalam interaksi inilah berbagai method dipanggil dari Controller.



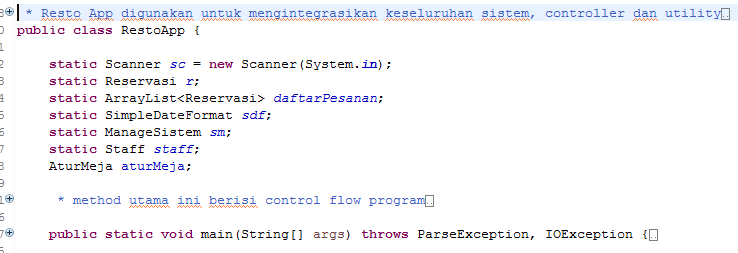
**Daftar Class dalam program :**

****

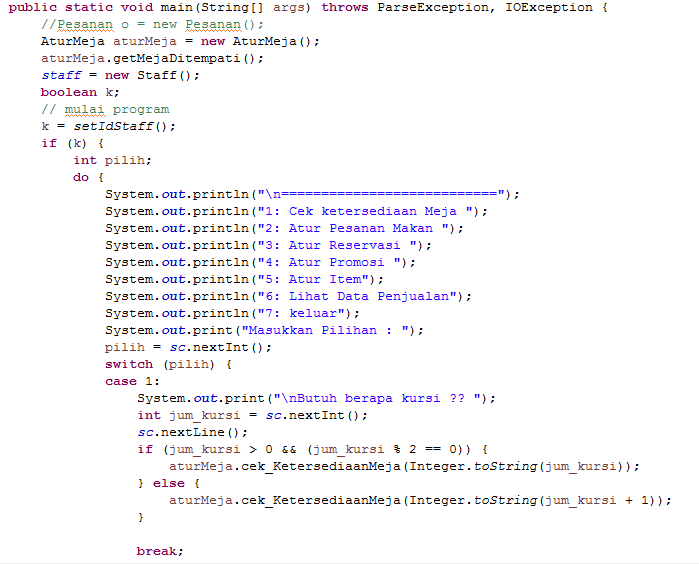
# BAB IV SOURCE CODE & SCREEN SHOOT

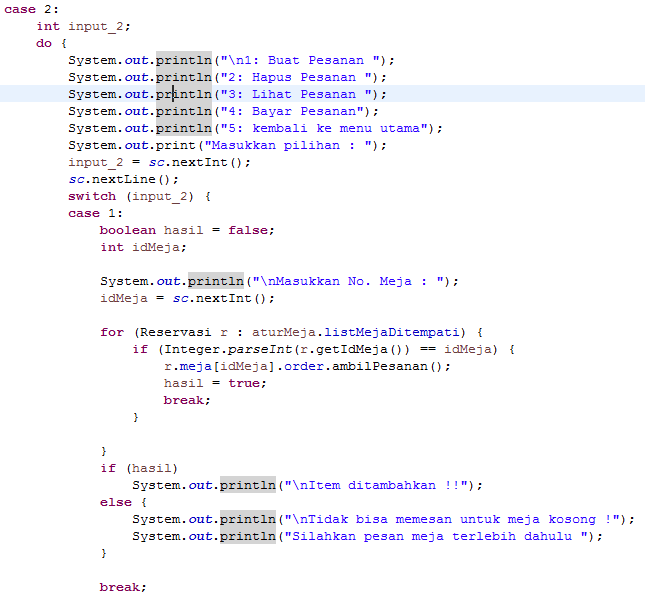
\*Untuk menghemat tempat, semua fungsi Get-Set tidak akan ditampilkan pada screenshots.

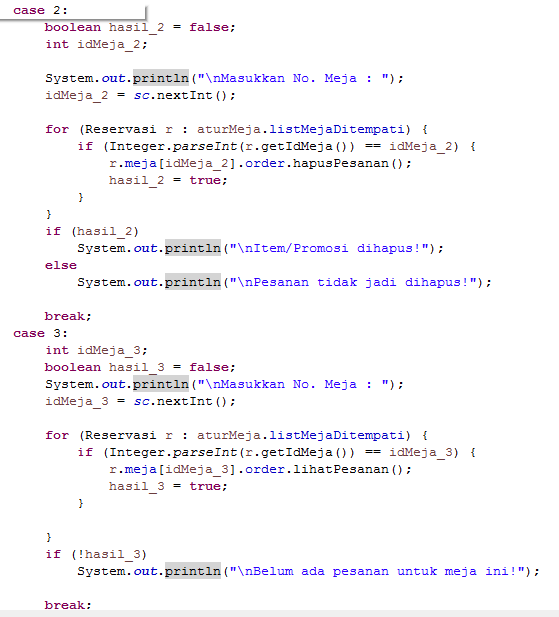
**RestoApp.Java**

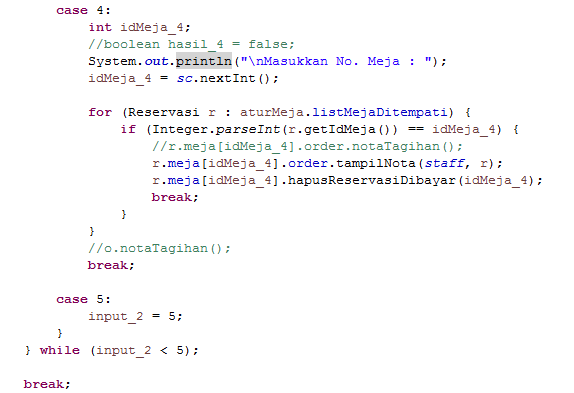
****

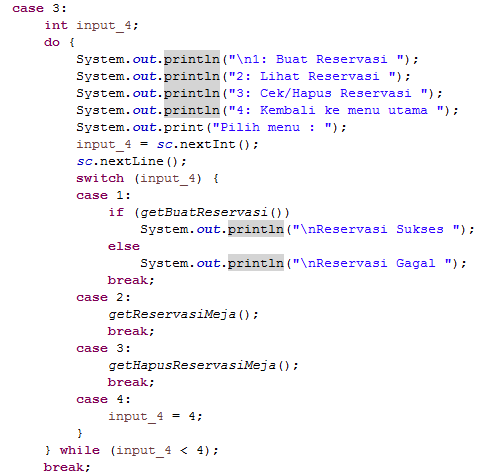
**Main Method**

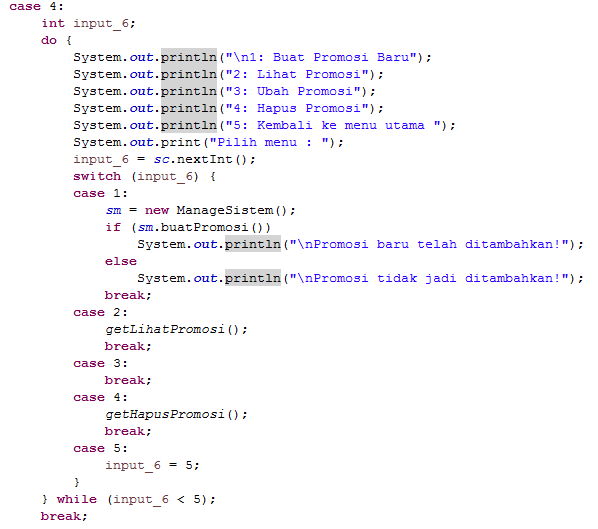
****

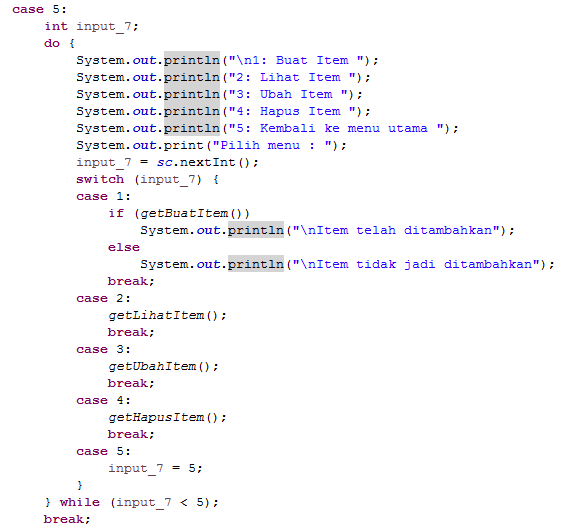
****

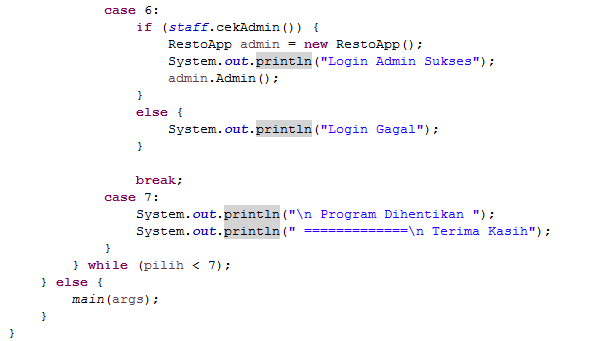
****

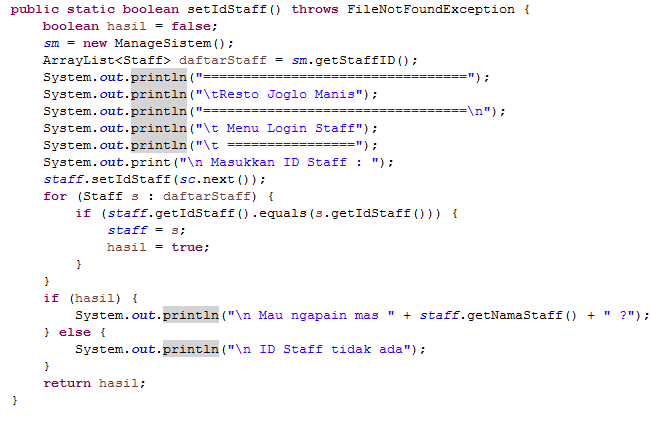
****

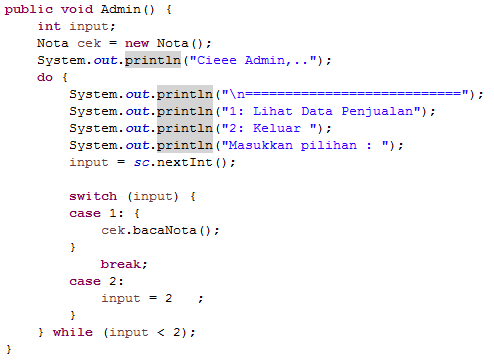
****

****

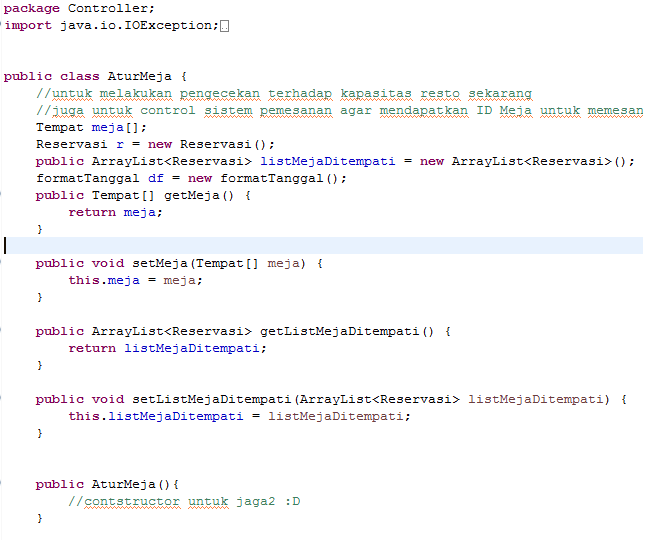
****

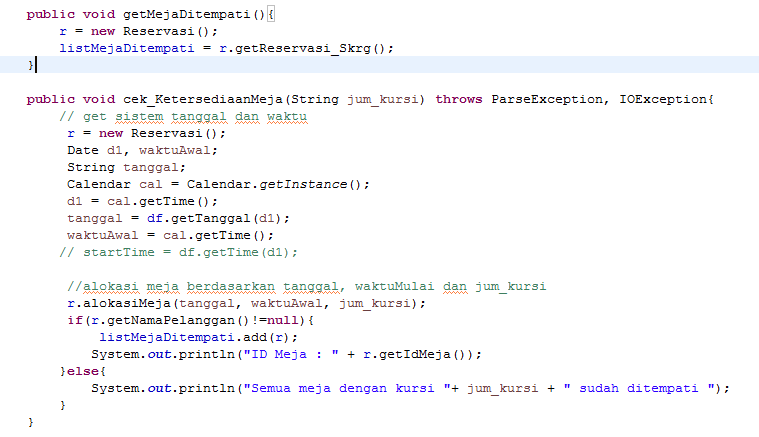
****

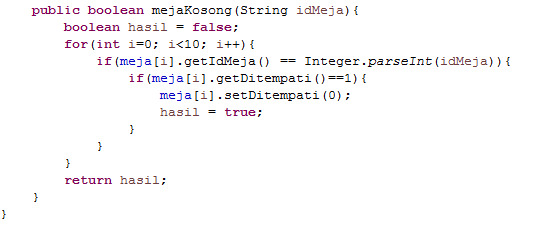
****

****

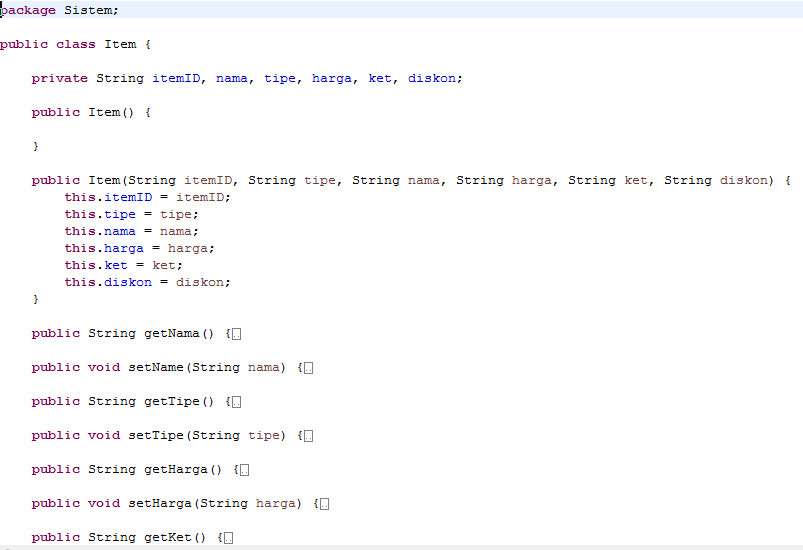
**Atur Meja.Java**

****

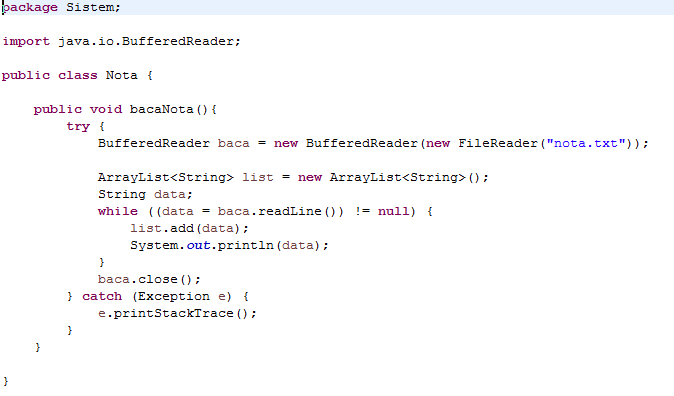
****

****

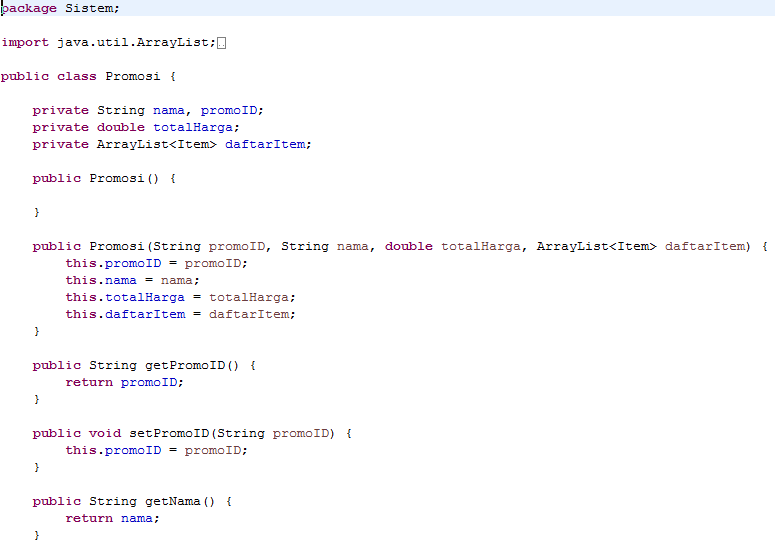
**Item.Java**

****

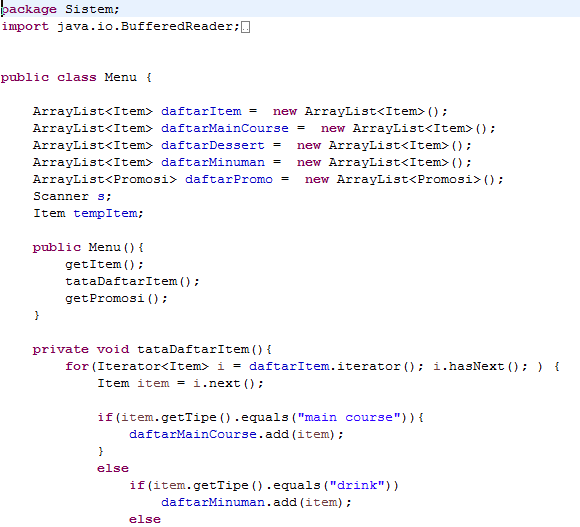
**Nota.Java**

****

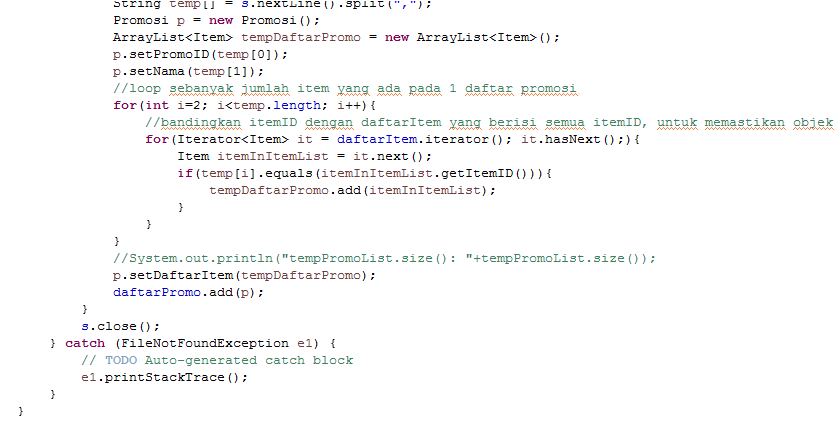
**Promosi.Java**

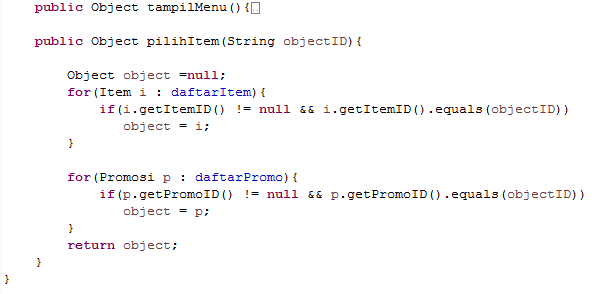
****

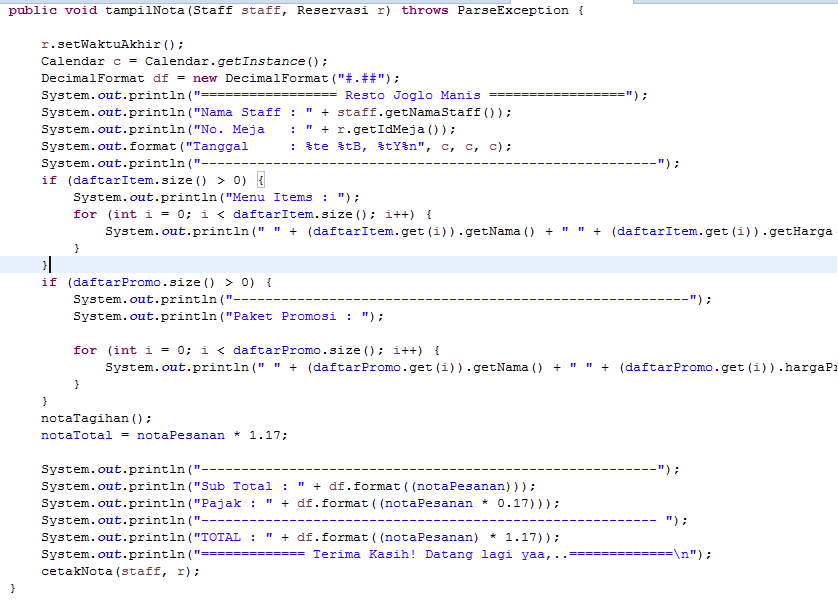
**Menu.Java**

****

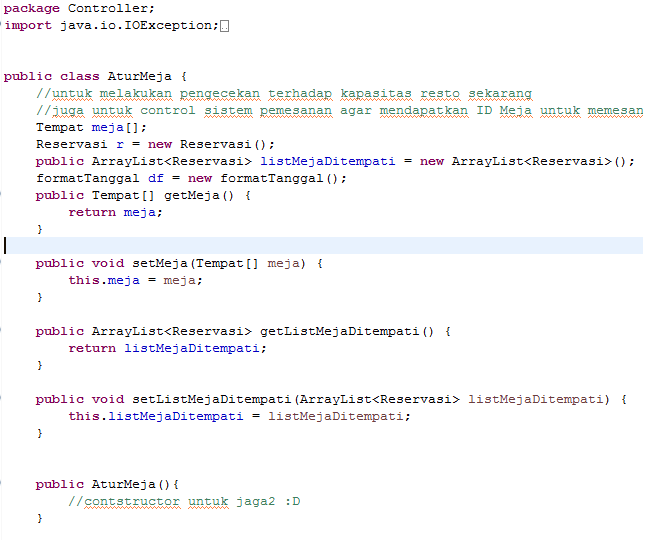
****

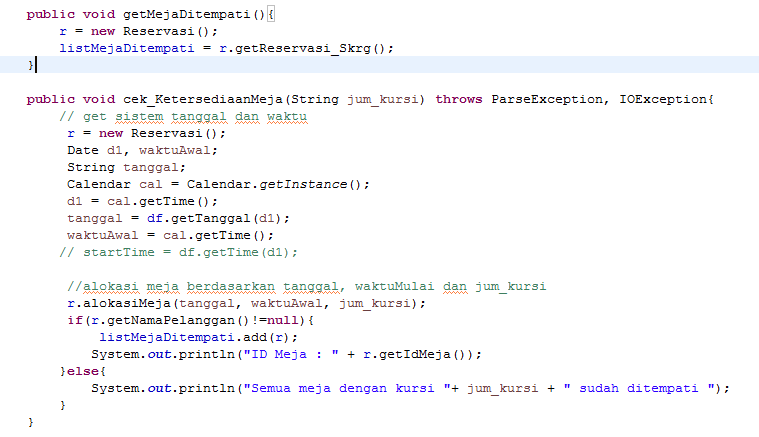
****

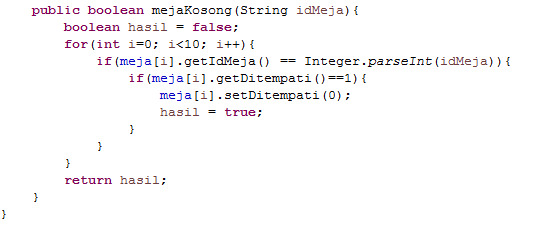
****

****

**AturMeja.Java**

****

****

****

# BAB V PENUTUP

## Analisa

Program ini bekerja dengan menggunakan Array List untuk mengatur objek-objek yang dibuat bedasarkan class-class tertentu. Class-class yang dipakai sebagai dasaran objek Array List adalah : Item, Promosi dan Staff.

Class Atur Meja mengatur keberadaan Meja-meja, ketersediaan dan juga menu-menu yang dipesan. Objek yang dihasilkan dari class ini akan disimpan pada RestoApp dan digunakan sebagai acuan untuk ArrayList yang mendata meja-meja yang ada.

Class yang paling sering dipakai selain Main class adalah ManageSistem. Pada class ManageSistem terdapat berbagai metode get-set yang menarik dan mengatur objek-objek pada class-class package Utility dan package Sistem.

Main Class Program, RestoApp mengatur interaksi antar pengguna dan komputer. Dalam class ini terdeklarasi beberapa ArrayList untuk mewakili item-item menu, meja dan promosi. Semua Array tersebut diisi melalui method yang memanggil ke controller, lalu ke package sistem atau utility dan terakhir tersampai ke class masing-masing.

Kekurangan dari program ini adalah belum bisa menampilkan jumlah penjualan untuk masing-masing item, tidak mengaplikasikan fitur membership dan penulisan program yang rumit, sehingga susah dimengerti oleh pemerogram lain.

## Kesimpulan

Dengan adanya program ini di PC kasir restoran, maka staff akan dapat menginput pesanan bedasarkan meja, menambahkan, mengurangi atau mengubah menu,melakukan reservasi hanya dengan menginput di komputer dan melihat Nota-nota yang telah dicetak oleh mesin, khusus untuk Manager restoran.