**LAPORAN HASIL PRAKTIKUM ALGORITMA DAN STRUKTUR DATA**

**JOBSHEET 8**



MUHAMMAD AMMAR HAFIZH

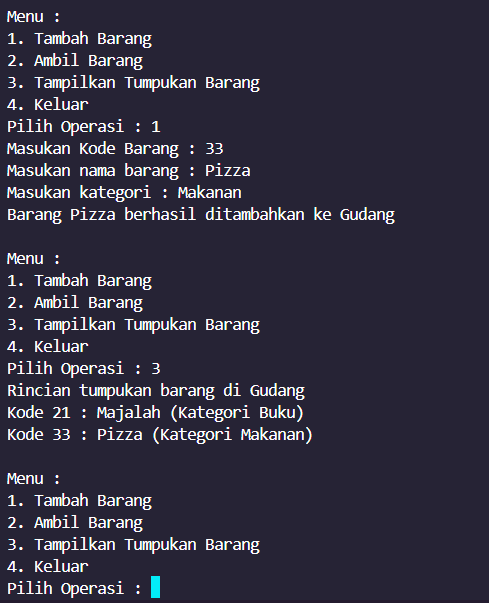
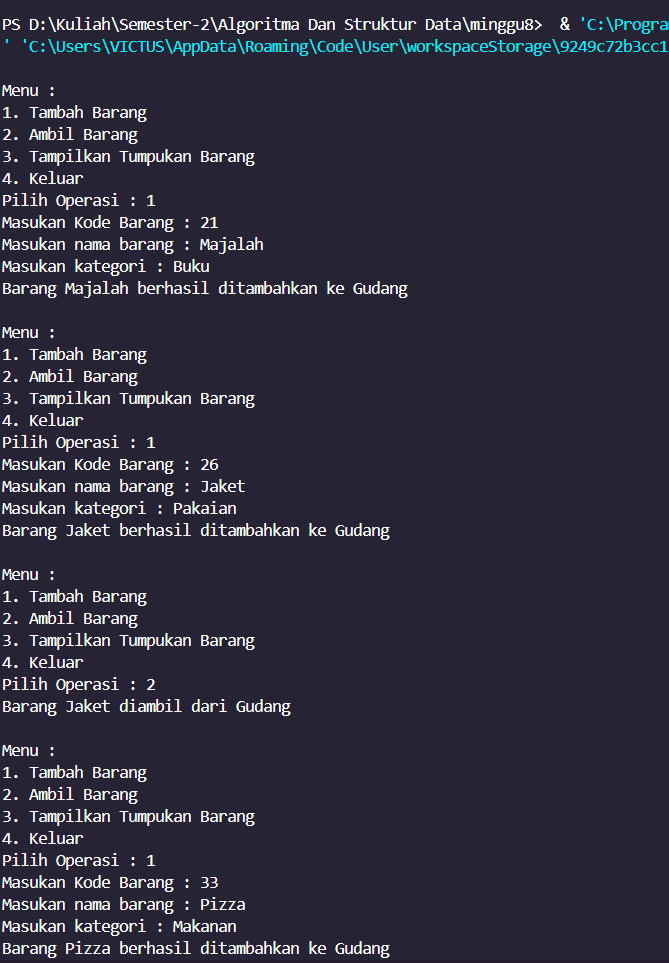
(2341720074)

D-IV TEKNIK INFORMATIKA – 1E

**Jurusan Teknologi Informasi**

**Politeknik Negeri Malang**

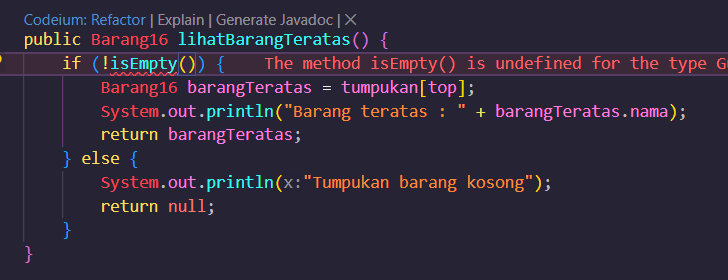
**Hasil Praktikum Percobaan 1**

****

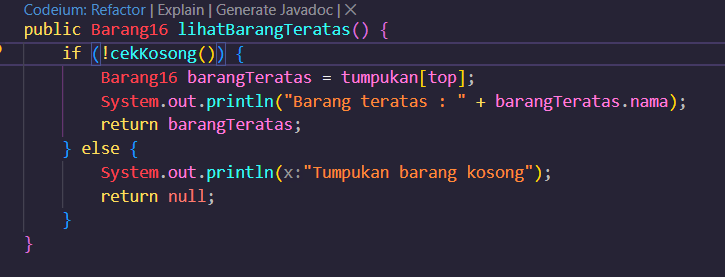
**Pertanyaan Percobaan 1**

1. Lakukan perbaikan pada kode program, sehingga keluaran yang dihasilkan sama dengan verifikasi hasil percobaan! Bagian mana saja yang perlu diperbaiki?

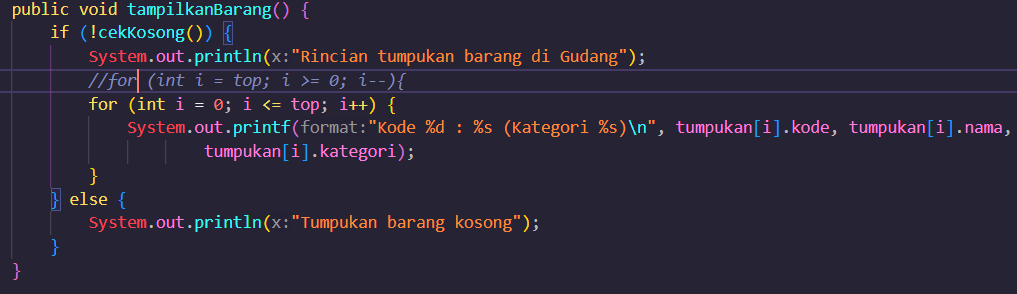
* Before



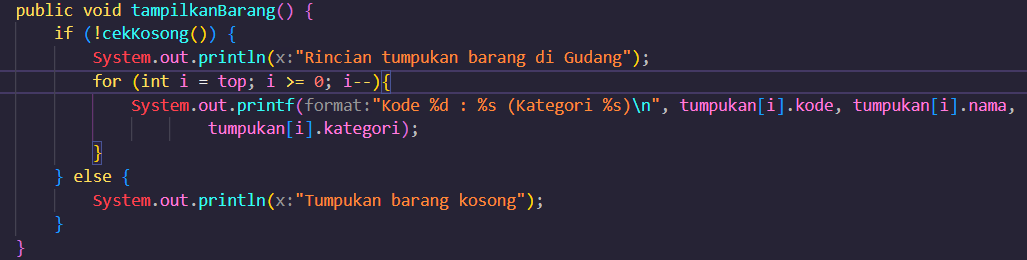
* After



* Before

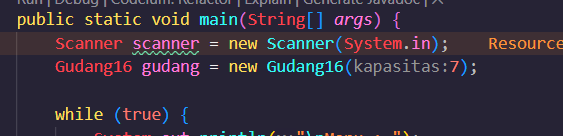


* After



2. Berapa banyak data barang yang dapat ditampung di dalam tumpukan? Tunjukkan potongan kode programnya!

* Size pada stack Gudang adalah 7

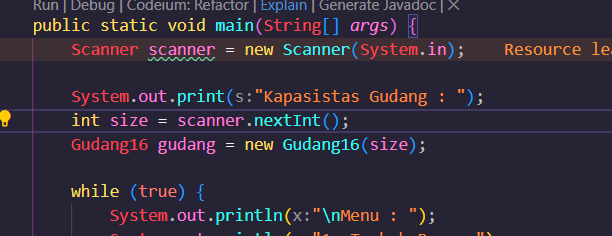


3. Mengapa perlu pengecekan kondisi !cekKosong() pada method tampilkanBarang? Kalau kondisi tersebut dihapus, apa dampaknya?

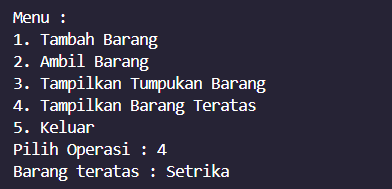
* Jika tidak melakukan pengecekan kondisi kosong pada stack bisa jadi stack itu sedang di posisi kosong dan akan terjadi error jika ingin mengambil barang karena tidak ada data apapun dalam stack.

4. Modifikasi kode program pada class Utama sehingga pengguna juga dapat memilih operasi lihat barang teratas, serta dapat secara bebas menentukan kapasitas gudang!

* Size Gudang

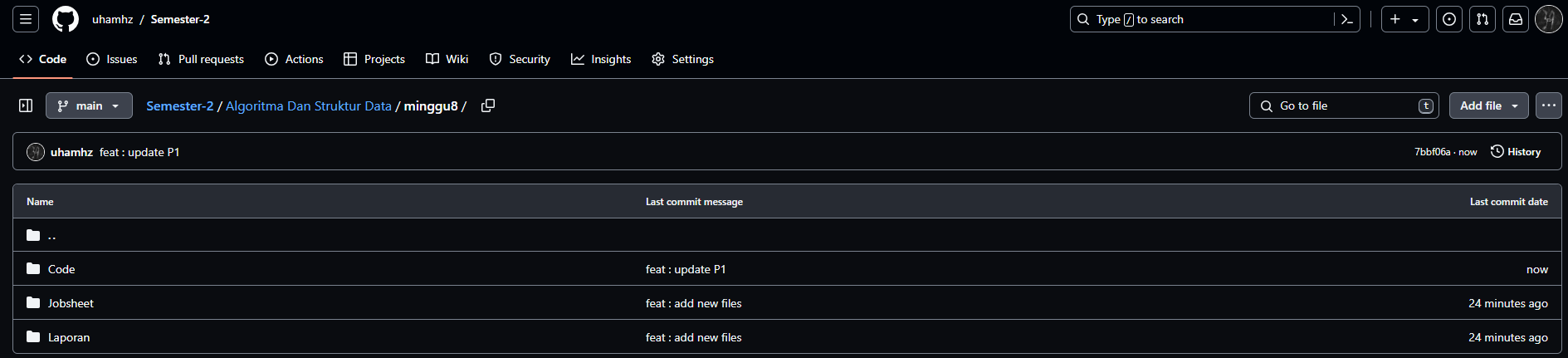


* Case 4 Barang Teratas

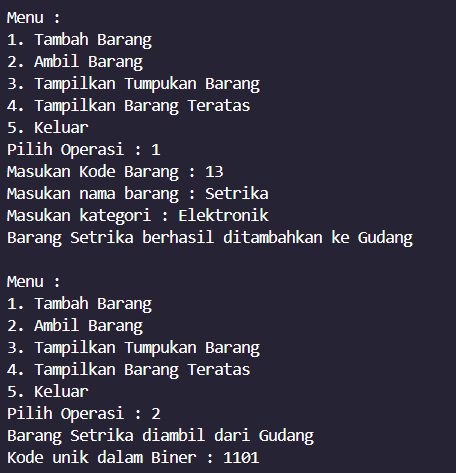


5. Commit dan push kode program ke Github

* <https://github.com/uhamhz/Semester-2/tree/main/Algoritma%20Dan%20Struktur%20Data/minggu8>



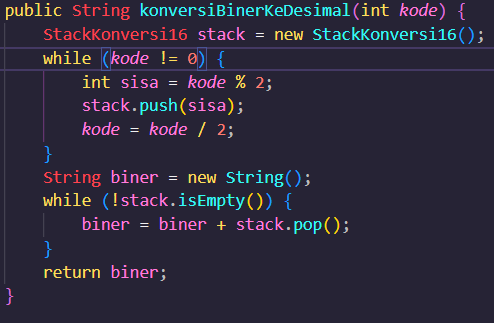
**Hasil Praktikum Percobaan 2**

****

**Pertanyaan Percobaan 2**

1. Pada method konversiDesimalKeBiner, ubah kondisi perulangan menjadi while (kode != 0), bagaimana hasilnya? Jelaskan alasannya!

* Hasilnya sama saja, karena kondisi mengingkan kode tidak sama dengan 0 dan kondisi sebelumnya menginginkan kode lebih dari 0.



2. Jelaskan alur kerja dari method konversiDesimalKeBiner!

* Sebelum dikonversi kode barang akan dicek terlebih dahulu bukan sama dengan 0

Dan setelah itu kode barang dimodulus 2 dan hasilnya dimasukan ke dalam stack dan setelah itu kode dibagi 2 dan terus melakukan perulangan seperti itu sampai kode sama dengan 0 sehabis kode sisa tadi dimasukan ke dalam stake, stake akan mengeluarkan dari atas dan akan menjadi kode biner.

**Hasil Praktikum Percobaan 3**

****

**Pertanyaan Percobaan 3**

1. Pada method derajat, mengapa return value beberapa case bernilai sama? Apabila return value diubah dengan nilai berbeda-beda setiap case-nya, apa yang terjadi?

* Karena ada beberapa operator yang sederajat dan apabila diubah dengan nilai yang berbeda beda akan menghasilkan postfix yang tidak benar.

2. Jelaskan alur kerja method konversi!

* Pada konversi dari infix ke postfix dibuat kolom string yang masih kosong dan dilakukan perulangan sampai n (total) jika yang ditemukan adalah operand maka akan di pindahkan ke string dan apabila buka kurung yang ditemukan akan dimasukan ke stack jika tutup kurung yang ditemukan maka tumpukan yang di stack akan di keluarkan sampai ketemu buka kurung dan jika yang ditemukan operator maka akan dibandingkan terlebih dahulu dengan stack teratas apabila lebih dari derajat pada stack teratas maka akan dipindahkan ke stack.

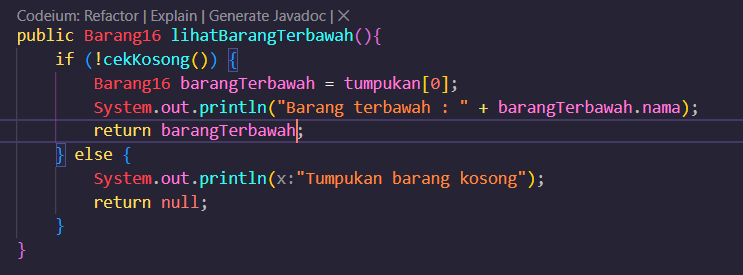
3. Pada method konversi, apa fungsi dari potongan kode berikut?

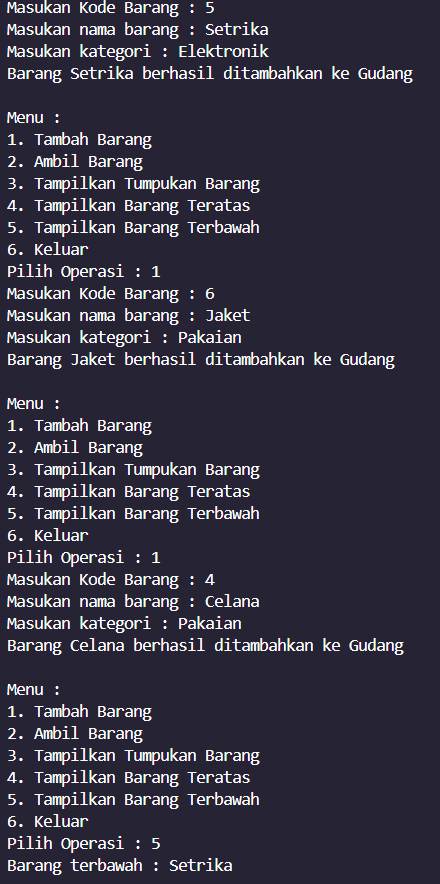
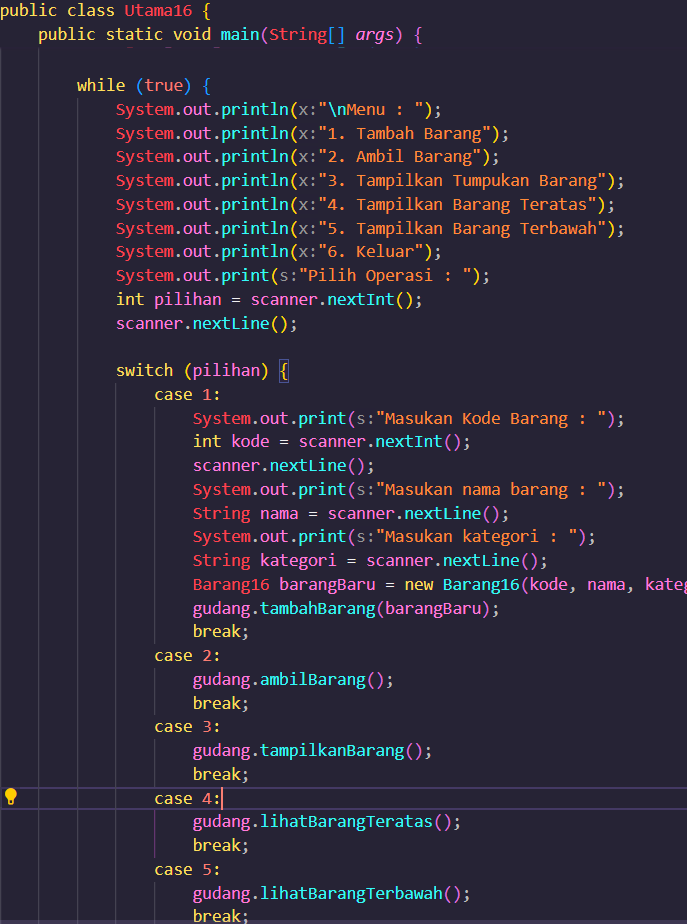
****

* C sama dengan infix bagian ke-i yang dimasukan dalam main postfix yang sudah ditrim atau dipisah-pisah (dijadikan char)

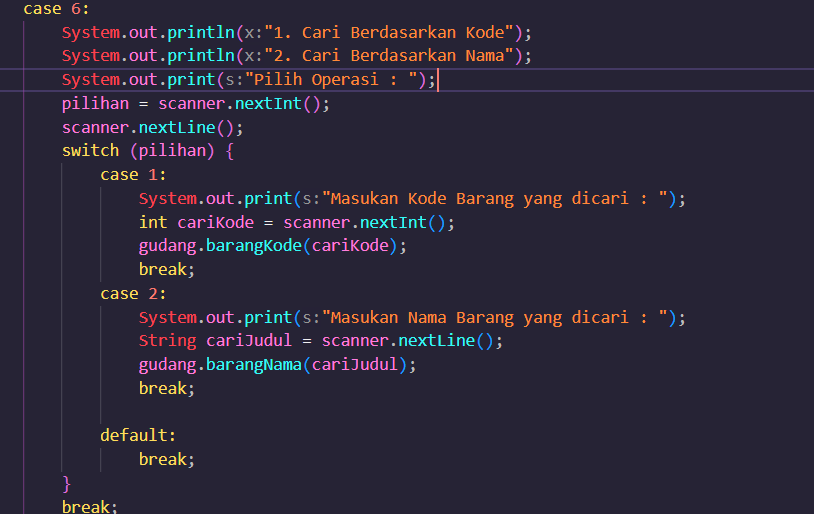
**Latihan Praktikum**

* **Method lihatBarangTerbawah**

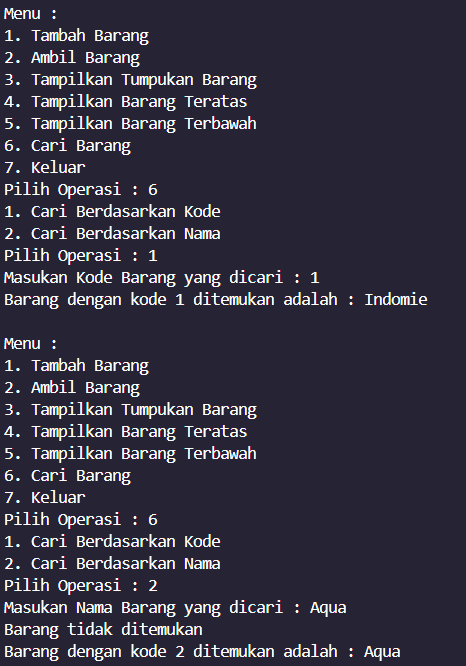
****

****

* **Method cariBarang**



****

****