

Program MESIN JAJA

```
{ Program Mesin Jaja menyimpan data sejumlah n integer baki dalam matriks matriksBaki.  
  
Untuk setiap baki disimpan: nomor baki, harga barang, jumlah barang.  
  
Program akan meminta input matriksBaki untuk setup,  
  
lalu meminta nomor baki yang barangnya akan pelanggan beli sampai semua barang dalam Mesin Jaja habis. }
```

KAMUS

```
kondisi, n, count, i: integer  
  
matriksBaki: array [1..3] of array [1..n] of integer  
  
procedure modusLoading (input n: integer; output matriksBaki: array [1..3] of array [1..n] of integer)  
    {untuk men-setup Mesin Jaja}  
  
procedure modusOperation (input x: integer [1..n]; input/output matriksBaki: array [1..3] of array [1..n] of integer)  
    {untuk mengurangi total barang ke-x dengan 1 jika totalnya tidak 0}
```

ALGORITMA

```
kondisi ← 0  
  
while kondisi = 0 do {ketika salah input, akan meminta input terus sampai inputnya benar}  
    output("banyaknya baki = ")  
    input(n) {meminta banyaknya n baki, I.S.}  
    if n > 0 and n ≤ 40 then  
        modusLoading(n,matriksBaki) {mengisi matriksBaki, I.S.}  
        kondisi ← 1  
    else  
        output("hanya tersedia 1 s.d. 40 baki")  
  
count ← 0  
  
i traversal [1..n] {menjumlahkan total barang yang tersisa}  
    count ← count + matriksBaki[i][3]  
  
while count ≠ 0 do  
    output("Pilih barang ke-x yang akan dibeli")  
    input(x) {meminta untuk memilih barang mana yang akan dibeli, I.S.}  
    count ← count - matriksBaki[x][3] {mengurangi count dengan total barang x sebelumnya}  
    modusOperation(x,matriksBaki)  
    count ← count + matriksBaki[x][3] {menambahkan count dengan total barang x setelah dibeli  
    (tidak mengurangi count dengan 1 langsung untuk menghindari jika barang ke-x sudah habis (0))}  
    output("SOLD OUT") {jika semua barang di Mesin Jaja sudah habis, F.S.}
```

```
procedure modusLoading (input n: integer; output matriksBaki: array [1..3] of array [1..n] of integer)  
{ I.S. n (banyak) baki yang terdefinisi, menginput setiap element dari  
    matriksBaki ([nomor baki, harga barang, jumlah barang] sebanyak n baki)  
    F.S. matriksBaki yang sudah terisi }
```

KAMUS LOKAL

```
i, condishong : integer
```

ALGORITMA

```
output("SETUP MESIN JAJA")  
  
i traversal [1..n]  
    output("Untuk baki ke-",i,":")  
    matriksBaki[i][1] ← i {memasukkan nomor baki ke matriksBaki[i][1]}  
  
    output("harga barang ke-",i," = ")  
    condition ← 0  
    while condishong = 0 do {untuk validasi, saat kondisinya belum terpenuhi, akan me-loop}  
        input(matriksBaki[i][2])  
        if matriksBaki[i][2] mod 5000 = 0 then  
            condishong ← 1  
        else  
            output("*Masukkan tidak valid! Harga barang harus kelipatan 5000.")  
  
    output("jumlah barang ke-",i," = ")  
    while condishong = 1 do {untuk validasi}  
        input(matriksBaki[i][3])  
        if matriksBaki[i][3] ≥ 0 or matriksBaki[i][3] ≤ 10 then  
            condishong ← 2  
        else  
            output("*Masukkan tidak valid! Jumlah barang harus di antara 0 s.d. 10.")
```

```
procedure modusOperation (input x: integer [1..n]; input/output matriksBaki: array [1..3] of array [1..n] of integer)  
{ I.S. barang ke-x yang akan dibeli dari matriksBaki  
    F.S. matriksBaki yang matriksBaki[x][3]-nya sudah dikurangi 1, jika > 0}
```

KAMUS LOKAL

ALGORITMA

```
if matriksBaki[x][3] > 0 then  
    matriksBaki[x][3] ← matriksBaki[x][3] - 1
```