Lampiran 3 *Source Code* Program

```
import java.util.*;
public class eko_2{
        public static double matriksData[][];
        public static double matriksDataAwal[][];
        public static double permintaan[];
        public static double persediaan[];
        public static boolean cekPermintaan[];
        public static boolean cekPersediaan[];
        public static double garisMatriks[][];
        public static double matriksHasil[][];
        public static boolean cekPP=true;
        public static double tempPermintaan[];
        public static double tempPersediaan[];

        public static void main(String[]args){
                Scanner input = new Scanner(System.in);
                int pilih;
                do{
                        System.out.println ("---------- Program Improved Zero Point Method----------");
                        System.out.println ("1. Mulai Percobaan.");
                        System.out.println ("0. Keluar!");
                        System.out.print ("Masukkan Pilihan Anda: ");
                        pilih=input.nextInt();
```

```
                switch(pilih){
                        case 1:
                                inisialisasiMatriks();
                                printTabel(matriksData, permintaan, persediaan,4,5,"Matriks Data",2,0);
                                kurangiBaris();
                                printTabel(matriksData, permintaan, persediaan,4,5,"Pengurangan Elemen
Baris",2,0);
                                kurangiKolom();
                                printTabel(matriksData, permintaan, persediaan,4,5,"Pengurangan Elemen
Kolom",2,0);
                                cekPermintaanPersediaan();
                                cekPersediaanPermintaan();
                                boolean iterasi=cekIterasi();
                                int jumIterasi=1;
                                while(iterasi==true){
                                        cekPP=true;
                                        System.out.println ("---------- Iterasi "+(jumIterasi++)+" ----------
----------");
                                        tutupGaris();
                                        printTabelCek(matriksData, garisMatriks, permintaan,
persediaan,4,5,"Cek Elemen Kolom dan Baris",2,0);
                                        double temp=cekTerkecilElemen(matriksData, garisMatriks);
                                        kurangiElemen();
                                        printTabel(matriksData, permintaan, persediaan,4,5,"Pengurangan
Elemen\n    Yang Tidak Tertutup Garis",2,0);
```

```
                                                tambahElemen(temp);
                                                printTabel(matriksData, permintaan, persediaan,4,5,"Penambahan
Elemen\n      Yang Tertutup 2 Garis",2,0);

                                                cekPermintaanPersediaan();
                                                cekPersediaanPermintaan();
                                                iterasi=cekIterasi();
                                        }
                                        System.out.println ("------------ Iterasi Berhenti ----------------\n");
                                        hasilAkhir();
                                        printTabelAkhir(matriksHasil, matriksData, permintaan, persediaan,4,5,"Hasil
Akhir",2,0);

                                        double biayaMinimum=biayaMinimum(matriksData);
                                        System.out.println ("Biaya minimum: "+biayaMinimum);

                                }
                }while (pilih!=0);
        }
        public static void inisialisasiMatriks(){
                matriksData = new double[4][5];
                matriksDataAwal = new double[4][5];
                permintaan = new double[5];
                persediaan = new double[4];

                Scanner input = new Scanner (System.in);
                System.out.println ("\n------------ Masukan Biaya ----------------\n");
                for(int i=0;i<4;i++){
                        for(int j=0;j<5;j++){
```

```
                                System.out.print("Masukkan data untuk biaya pada baris "+(i+1)+" dan kolom "+(j+1)+":
");
                                matriksData[i][j]=input.nextDouble();
                                matriksDataAwal[i][j]=matriksData[i][j];
                        }
                }
                System.out.println ("\n---------- Masukan Persediaan ----------\n");
                for(int i=0;i<4;i++){
                        System.out.print("Masukkan data untuk persediaan pada baris "+(i+1)+": ");
                        persediaan[i]=input.nextDouble();
                }
                System.out.println ("\n---------- Masukan Permintaan ----------\n");
                for(int j=0;j<5;j++){
                        System.out.print("Masukkan data untuk biaya pada kolom "+(j+1)+": ");
                        permintaan[j]=input.nextDouble();
                }
        }
        public static double cekTerkecilBaris(int idx, double [][]matriks){
                double nilaiTerkecil=1000.0;
                for(int j=0;j<5;j++){
                        if(matriks[idx][j]<nilaiTerkecil){
                                nilaiTerkecil=matriks[idx][j];
                        }
                }
                return nilaiTerkecil;
        }
```

```
public static double cekTerkecilKolom(int idx, double [][]matriks){
        double nilaiTerkecil=1000.0;
        for(int i=0;i<4;i++){
                if(matriks[i][idx]<nilaiTerkecil){
                        nilaiTerkecil=matriks[i][idx];
                }
        }
        return nilaiTerkecil;
}
public static double cekTerkecilElemen(double [][]matriks, double [][]garis){
        double nilaiTerkecil=1000.0;
        for(int i=0;i<4;i++){
                for(int j=0;j<5;j++){
                        if(garis[i][j]!=2){
                                if(matriks[i][j]<nilaiTerkecil){
                                        nilaiTerkecil=matriks[i][j];
                                }
                        }
                }
        }
        return nilaiTerkecil;
}
public static void kurangiBaris(){
        for(int i=0;i<4;i++){
                double temp=cekTerkecilBaris(i, matriksData);
                for(int j=0;j<5;j++){
```

```
                        matriksData[i][j]=matriksData[i][j]-temp;
                }
        }
}

public static void kurangiKolom(){
        for(int j=0;j<5;j++){
                double temp=cekTerkecilKolom(j, matriksData);
                for(int i=0;i<4;i++){
                        matriksData[i][j]=matriksData[i][j]-temp;
                }
        }
}
public static double jumNolKolomPermintaan(int idx, double[][]matriks){
        double temp=0.0;
        for(int i=0;i<4;i++){
                if(matriks[i][idx]==0.0){
                        temp=temp+persediaan[i];
                }
        }
        return temp;
}
public static double jumNolBarisPersediaan(int idx, double[][]matriks){
        double temp=0.0;
        for(int j=0;j<5;j++){
                if(matriks[idx][j]==0.0){
```

```
                    temp=temp+permintaan[j];
                }
            }
            return temp;
        }
        public static String cekNolKolom(int idx, double [][]matriks){
            String temp="";
            int tes=0;
            for(int j=0;j<5;j++){
                if(matriks[idx][j]==0.0){
                    tes++;
                    if(tes==1){
                        temp=temp+"K"+(1+j)+" ";
                        tes++;
                    }else if(tes>1){
                        temp=temp+"+ K"+(1+j)+" ";
                    }
                }
            }
            return temp;
        }
        public static String cekNolBaris(int idx, double [][]matriks){
            String temp="";
            int tes=0;
            for(int i=0;i<4;i++){
                if(matriks[i][idx]==0.0){
```

```
                        tes++;
                        if(tes==1){
                                temp=temp+"B"+(1+i)+" ";
                                tes++;
                        }else if(tes>1){
                                temp=temp+"+ B"+(1+i)+" ";
                        }
                }
        }
        return temp;
}
public static void cekPermintaanPersediaan(){
        cekPermintaan=new boolean[5];
        for(int j=0;j<5;j++){
                double jumlahBaris=jumNolKolomPermintaan(j,matriksData);
                String nolBaris=cekNolBaris(j,matriksData);
                if(cekPP==true){
                        if(permintaan[j]<=jumlahBaris){
                                cekPermintaan[j]=true;
                                System.out.println ("K"+(1+j)+" <= "+nolBaris);

                        }
                        else{
                                cekPermintaan[j]=false;
                                System.out.println ("K"+(1+j)+" > "+nolBaris);
                                cekPP=false;
```

```
                    }
                }else if(cekPP==false){
                    if(permintaan[j]<=jumlahBaris){
                        cekPermintaan[j]=true;
                        System.out.println ("K"+(1+j)+" <= "+nolBaris);

                    }
                    else{
                        cekPermintaan[j]=true;
                        System.out.println ("K"+(1+j)+" > "+nolBaris);
                    }
                }

            }
        }
        public static void cekPersediaanPermintaan(){
            cekPersediaan=new boolean[4];
            for(int i=0;i<4;i++){
                double jumlahKolom=jumNolBarisPersediaan(i,matriksData);
                String nolKolom=cekNolKolom(i,matriksData);
                if(cekPP==true){
                    if((persediaan[i]<=jumlahKolom)){
                        cekPersediaan[i]=true;
                        System.out.println ("B"+(1+i)+" <= "+nolKolom);

                    }
```

```
                    else{
                        cekPersediaan[i]=false;
                        System.out.println ("B"+(1+i)+" > "+nolKolom);
                        cekPP=false;
                    }
                }else if(cekPP==false){
                    if((persediaan[i]<=jumlahKolom)){
                        cekPersediaan[i]=true;
                        System.out.println ("B"+(1+i)+" <= "+nolKolom);
                    }
                    else{
                        cekPersediaan[i]=true;
                        System.out.println ("B"+(1+i)+" > "+nolKolom);
                    }
                }
            }
        }
    public static boolean cekGaris(int idI,int idJ, double [][]matriks){
            boolean temp=false;
            int temp1=0;
            for(int i=0;i<4;i++){
                if(matriks[i][idJ]==1.0 && cekPersediaan[i]==false ) temp1++;
            }
            for(int j=0;j<5;j++){
                if(matriks[idI][j]==1.0 && cekPermintaan[j]==false ) temp1++;
            }
```

```
        if(temp1>=1) temp=true;
        return temp;
    }
    public static boolean cekGaris2(int idI,int idJ, double [][]matriks){
        boolean temp=false;
        int temp1=0, temp2=0;
        for(int i=0;i<4;i++){
            if(matriks[i][idJ]==2.0) temp1++;
        }
        for(int j=0;j<5;j++){
            if(matriks[idI][j]==2.0) temp2++;
        }
        if(temp1>=3 || temp2>=4) temp=true;
        return temp;
    }
    public static boolean cekGarisMendatar(int idI, double[][]matriks){
        boolean temp=true;
        for(int j=0;j<5;j++){
            if(matriks[idI][j]==0.0){
                temp=false;
                break;
            }
        }
        return temp;
    }
```

```
public static boolean cekGarisTegak(int idJ, double[][]matriks){
        boolean temp=true;
        for(int i=0;i<4;i++){
                if(matriks[i][idJ]==0.0){
                        temp=false;
                        break;
                }
        }
        return temp;
}

public static void ubahDatar(int idI){
        for(int j=0;j<5;j++){
                if(garisMatriks[idI][j]==1.0) garisMatriks[idI][j]++;
        }
}

public static void ubahTegak(int idJ){
        for(int i=0;i<4;i++){
                if(garisMatriks[i][idJ]==1.0) garisMatriks[i][idJ]++;
        }
}
public static void tutupGaris(){
        garisMatriks=new double[4][5];
        for(int i=0;i<4;i++){
                for(int j=0;j<5;j++){
```

```
                        if(cekPersediaan[i]==true || matriksData[i][j]==0.0){
                                garisMatriks[i][j]++;
                        }
                }
        }
        //printTabel(garisMatriks, permintaan, persediaan,4,5,"1",2,0);
        for(int j=0;j<5;j++){
                for(int i=0;i<4;i++){
                        if(cekPermintaan[j]==true || matriksData[i][j]==0.0){
                                garisMatriks[i][j]++;
                        }
                }
        }
        //printTabel(garisMatriks, permintaan, persediaan,4,5,"2",2,0);
        double temp[][]=new double[4][5];
        for(int i=0;i<4;i++){
                for(int j=0;j<5;j++){
                        temp[i][j]=garisMatriks[i][j];
                }
        }
        for(int i=0;i<4;i++){
                for(int j=0;j<5;j++){
                        boolean cekTdkTerpenuhi=cekGaris(i,j,temp);
                        if(cekTdkTerpenuhi==true && matriksData[i][j] != 0.0 ){
                                garisMatriks[i][j]--;
                        }
```

```
            }
        }
        //printTabel(garisMatriks, permintaan, persediaan,4,5,"3",2,0);
        for(int i=0;i<4;i++){
            for(int j=0;j<5;j++){
                temp[i][j]=garisMatriks[i][j];
            }
        }

        for(int i=0;i<4;i++){
            for(int j=0;j<5;j++){
                boolean cekTdkTerpenuhi2=false;
                if(temp[i][j]==1.0){
                    cekTdkTerpenuhi2=cekGaris2(i,j,garisMatriks);
                    System.out.println (i+" "+j+" "+cekTdkTerpenuhi2);
                    if(cekTdkTerpenuhi2==true) {
                        garisMatriks[i][j]++;
                    }
                }
            }
        }

        for(int i=0;i<4;i++){
            for(int j=0;j<5;j++){
                temp[i][j]=garisMatriks[i][j];
            }
```

```
            }

            //printTabel(garisMatriks, permintaan, persediaan,4,5,"4",2,0);
            for(int i=0;i<4;i++){
                  for(int j=0;j<5;j++){
                        boolean cekTdkTerpenuhi3=false;
                        if(matriksData[i][j]==0.0){
                              cekTdkTerpenuhi3=cekGaris(i,j,temp);
                              if(cekTdkTerpenuhi3==false){
                                    boolean cekDatar=cekGarisMendatar(i,temp);
                                    boolean cekTegak=cekGarisTegak(j,temp);
                                    //System.out.println (i+" "+j+" "+cekDatar+" "+cekTegak);
                                    if(cekDatar==true && cekTegak==false){
                                          ubahDatar(i);
                                    }else if(cekTegak==true && cekDatar==false){
                                          ubahTegak(j);
                                    }
                              }
                        }
                  }
            }
            //printTabel(garisMatriks, permintaan, persediaan,4,5,"5",2,0);
      }

      public static void kurangiElemen(){
            double temp=cekTerkecilElemen(matriksData, garisMatriks);
```

```
            for(int i=0;i<4;i++){
                for(int j=0;j<5;j++){
                    if(garisMatriks[i][j]!=2){
                        matriksData[i][j]=matriksData[i][j]-temp;
                    }
                }
            }
        }

    public static boolean jumlahGaris(int idI, int idJ, double[][]garis){
            boolean temp=false;
            int baris=0, kolom=0;
            for(int i=0;i<4;i++){
                if(garisMatriks[i][idJ]==2.0) baris++;
            }
            for(int j=0;j<5;j++){
                if(garisMatriks[idI][j]==2.0) kolom++;
            }
            if(baris==4 && kolom==5) temp=true;
            return temp;
    }

    public static void tambahElemen(double temp){
            for(int i=0;i<4;i++){
                for(int j=0;j<5;j++){
                    boolean temp2=jumlahGaris(i,j,garisMatriks);
```

```
                        if(temp2==true){
                            matriksData[i][j]=matriksData[i][j]+temp;
                        }
                    }
                }
            }

    public static boolean cekIterasi(){
            boolean temp=false;
            for(int i=0;i<4;i++){
                if(cekPersediaan[i]==false){
                    temp=true;
                    break;
                }
            }
            for(int j=0;j<5;j++){
                if(cekPermintaan[j]==false){
                    temp=true;
                    break;
                }
            }
            return temp;
    }

    public static double cariNilaiTertinggi(double temp, double [][]matriks){
            double nilaiTertinggi=-1000;
```

```
		for(int i=0;i<4;i++){
			for(int j=0;j<5;j++){
				if(matriks[i][j]<temp && matriks[i][j] > 0.0){
					if(matriks[i][j]>=nilaiTertinggi){
						nilaiTertinggi=matriks[i][j];
					}
				}
			}
		}
		return nilaiTertinggi;
}

public static boolean cekIterasi2(){
		boolean temp=false;int cek=0;
		for(int i=0;i<4;i++){
			if(tempPersediaan[i]==0){
				cek++;
			}
		}
		for(int j=0;j<5;j++){
			if(tempPermintaan[j]==0){
				cek++;
			}
		}
		if(cek==9){
			temp=true;
```

```
		}else{
			temp=false;
		}
		return temp;
	}

	public static void tempPermintaanPersediaan(){
		tempPersediaan=new double[4];
		tempPermintaan=new double[5];
		for(int i=0;i<4;i++){
			tempPersediaan[i]=persediaan[i];
		}
		for(int j=0;j<5;j++){
			tempPermintaan[j]=permintaan[j];
		}
	}

	public static boolean cekNolSatuDatar(int idI, double[][]matriks){
		boolean temp=false; int cek=0;
		for(int j=0;j<5;j++){
			if(matriks[idI][j]==0.0) cek++;
		}
		if(cek==1) temp=true;
		return temp;
	}
```

```
public static boolean cekNolSatuTegak(int idJ, double[][]matriks){
        boolean temp=false; int cek=0;
        for(int i=0;i<4;i++){
                if(matriks[i][idJ]==0.0) cek++;
        }
        if(cek==1) temp=true;
        return temp;
}

public static int cekNolDatar(int idI, double[][]matriks){
        int cek=0;
        for(int j=0;j<5;j++){
                if(matriks[idI][j]==0.0){
                        cek=j;
                        break;
                }
        }
        return cek;
}

public static int cekNolTegak(int idJ, double[][]matriks){
        int cek=0;
        for(int i=0;i<4;i++){
                if(matriks[i][idJ]==0.0){
                        cek=i;
                        break;
```

```
                }
            }
            return cek;
    }


    public static void pengisianElemenNol(int idI, int idJ, double[][]matriks, double[][]matriksHasil){
            boolean cekNolSatuDatar=cekNolSatuDatar(idI,matriks);
            boolean cekNolSatuTegak=cekNolSatuTegak(idJ,matriks);

            if(cekNolSatuDatar==true){
                int cekNolDatar=cekNolDatar(idI,matriksHasil);
                if(tempPermintaan[cekNolDatar]<=tempPersediaan[idI]){
                    matriksHasil[idI][cekNolDatar]=tempPermintaan[cekNolDatar];
                    tempPermintaan[cekNolDatar]=tempPermintaan[cekNolDatar]-
matriksHasil[idI][cekNolDatar];
                    tempPersediaan[idI]=tempPersediaan[idI]-matriksHasil[idI][cekNolDatar];
                }else{
                    matriksHasil[idI][cekNolDatar]=tempPersediaan[idI];
                    tempPersediaan[idI]=tempPersediaan[idI]-matriksHasil[idI][cekNolDatar];
                    tempPermintaan[cekNolDatar]=tempPermintaan[cekNolDatar]-
matriksHasil[idI][cekNolDatar];
                }
            }
            if(cekNolSatuTegak==true){
```

```
                int cekNolTegak=cekNolTegak(idJ,matriksHasil);
                if(tempPermintaan[idJ]<=tempPersediaan[cekNolTegak]){
                        matriksHasil[cekNolTegak][idJ]=tempPermintaan[idJ];
                        tempPermintaan[idJ]=tempPermintaan[idJ]-matriksHasil[cekNolTegak][idJ];
                        tempPersediaan[cekNolTegak]=tempPersediaan[cekNolTegak]-
matriksHasil[cekNolTegak][idJ];
                }else{
                        matriksHasil[cekNolTegak][idJ]=tempPersediaan[cekNolTegak];
                        tempPermintaan[idJ]=tempPermintaan[idJ]-matriksHasil[cekNolTegak][idJ];
                        tempPersediaan[cekNolTegak]=tempPersediaan[cekNolTegak]-
matriksHasil[cekNolTegak][idJ];
                }
        }
        if(cekNolSatuDatar==false){
                int cekNolDatar=cekNolDatar(idI,matriksHasil);
                for(int j=cekNolDatar;j<5;j++){
                        boolean cekNolSatuTegak2=cekNolSatuTegak(j,matriks);
                        if(cekNolSatuTegak2==true){
                                if(matriksHasil[idI][cekNolDatar]==0.0 &&
tempPermintaan[cekNolDatar]<=tempPersediaan[idI]){
                                        matriksHasil[idI][cekNolDatar]=tempPermintaan[cekNolDatar];
                                        tempPersediaan[idI]=tempPersediaan[idI]-
matriksHasil[idI][cekNolDatar];
                                        tempPermintaan[cekNolDatar]=tempPermintaan[cekNolDatar]-
matriksHasil[idI][cekNolDatar];
                                }else if(matriksHasil[idI][cekNolDatar]==0.0 &&
```

```
tempPermintaan[cekNolDatar]>tempPersediaan[idI]){
                                                matriksHasil[idI][cekNolDatar]=tempPersediaan[idI];
                                                tempPermintaan[cekNolDatar]=tempPermintaan[cekNolDatar]-
matriksHasil[idI][cekNolDatar];
                                                tempPersediaan[idI]=tempPersediaan[idI]-
matriksHasil[idI][cekNolDatar];
                                        }
                                }//else if(cekNolSatuTegak2==false)

                                cekNolDatar++;
                        }
                }
                if(cekNolSatuTegak==false){
                        int cekNolTegak=cekNolTegak(idJ,matriksHasil);
                        for(int i=cekNolTegak;i<4;i++){
                                boolean cekNolSatuDatar2=cekNolSatuDatar(i,matriks);
                                if(cekNolSatuDatar2==true){
                                        if(matriksHasil[cekNolTegak][idJ]==0.0 &&
tempPermintaan[idJ]<=tempPersediaan[cekNolTegak]){
                                                matriksHasil[cekNolTegak][idJ]=tempPermintaan[idJ];
                                                tempPermintaan[idJ]=tempPermintaan[idJ]-
matriksHasil[cekNolTegak][idJ];
                                                tempPersediaan[cekNolTegak]=tempPersediaan[cekNolTegak]-
matriksHasil[cekNolTegak][idJ];
                                        }else if(matriksHasil[cekNolTegak][idJ]==0.0 &&
tempPermintaan[idJ]>tempPersediaan[cekNolTegak]){
```

```
                                                matriksHasil[cekNolTegak][idJ]=tempPersediaan[cekNolTegak];
                                                tempPermintaan[idJ]=tempPermintaan[idJ]-
matriksHasil[cekNolTegak][idJ];
                                                tempPersediaan[cekNolTegak]=tempPersediaan[cekNolTegak]-
matriksHasil[cekNolTegak][idJ];
                                        }
                                }
                                cekNolTegak++;
                        }
                }
        }

        public static void hasilAkhir(){
                matriksHasil=new double[4][5];
                for(int i=0;i<4;i++){
                        for(int j=0;j<5;j++){
                                if(matriksData[i][j]!=0.0) matriksHasil[i][j]=0.1;
                        }
                }
                tempPermintaanPersediaan();
                boolean cekIterasi2=cekIterasi2();
                double nilaiMax=cariNilaiTertinggi(1000,matriksData);
                int tes=0;
                while(cekIterasi2==false){
                        int idI=0, idJ=0, cek=0;
                        for(int i=0;i<4;i++){
```

```
                                for(int j=0;j<5;j++){
                                        if(matriksData[i][j]==nilaiMax && cekIterasi2==false){
                                                idI=i;idJ=j;
                                                //System.out.println (nilaiMax+" "+i+" "+j);
                                                pengisianElemenNol(idI,idJ,matriksHasil,matriksHasil);
                                                //printTabel(matriksHasil, tempPermintaan, tempPersediaan,4,5,"Hasil
Akhir",2,0);
                                                cekIterasi2=cekIterasi2();
                                        }
                                }
                        }
                        nilaiMax=cariNilaiTertinggi(nilaiMax,matriksData);
                }
        }

        public static double biayaMinimum(double[][]matriks){
                double biayaMinimum=0.0;
                for(int i=0;i<4;i++){
                        for(int j=0;j<5;j++){
                                if(matriksHasil[i][j]!=0.1){
                                        biayaMinimum=biayaMinimum+(matriksHasil[i][j]*matriksDataAwal[i][j]);
                                }
                        }
                }
                return biayaMinimum;
        }
```

```
	private  static void printTabel(double[][] tabel,double permintaan[], double persediaan[], int tinggi,int lebar,String
header,int angkaDiDepanKoma, int angkaDiBelakangKoma){
		String
format="%"+Integer.toString(angkaDiDepanKoma)+"."+Integer.toString(angkaDiBelakangKoma)+"f%s";
		System.out.println ("                "+header);
		System.out.println ("-----------------------------------------------------------");
		System.out.println("|                  |            |      |      |       |              |Persediaan|");
		System.out.println ("-----------------------------------------------------------");
		for (int baris=0;baris<tinggi;baris++){
			System.out.print("|     |");
			for (int kolom=0;kolom<lebar;kolom++){
				System.out.printf(format,tabel[baris][kolom],"  |  ");
			}
			System.out.printf(format,persediaan[baris],"  |  ");
			System.out.println ();
		}
		System.out.println ("-----------------------------------------------------------");
		System.out.print("|Permintaan|");
		for (int kolom=0;kolom<lebar;kolom++){
			System.out.printf(format,permintaan[kolom],"  |  ");
		}
		System.out.println ();
		System.out.println ("-----------------------------------------------------------");
		System.out.println ();
	}
```

```
        private  static void printTabelCek(double[][] tabel,double[][] tabel2, double permintaan[], double persediaan[], int
tinggi,int lebar,String header,int angkaDiDepanKoma, int angkaDiBelakangKoma){
                String
format="%"+Integer.toString(angkaDiDepanKoma)+"."+Integer.toString(angkaDiBelakangKoma)+"f%s";
                String
format2="."+"%"+Integer.toString(angkaDiDepanKoma)+"."+Integer.toString(angkaDiBelakangKoma)+"f%s";
                System.out.println ("              "+header);
                System.out.println ("-----------------------------------------------");
                System.out.println("|              |     |     |     |     |Persediaan|");
                System.out.println ("-----------------------------------------------");
                for (int baris=0;baris<tinggi;baris++){
                        System.out.print("|     |");
                        for (int kolom=0;kolom<lebar;kolom++){
                                if(tabel2[baris][kolom]==2){
                                        System.out.printf(format2,tabel[baris][kolom]," | ");
                                }else{
                                        System.out.printf(format,tabel[baris][kolom]," | ");
                                }
                        }
                        System.out.printf(format,persediaan[baris]," | ");
                        System.out.println ();
                }
                System.out.println ("-----------------------------------------------");
                System.out.print("|Permintaan|");
                for (int kolom=0;kolom<lebar;kolom++){
```

```
                System.out.printf(format,permintaan[kolom]," | ");
            }
            System.out.println ();
            System.out.println ("-------------------------------------------------");
            System.out.println ();
        }

        private  static void printTabelAkhir(double[][] tabel,double[][] tabel2, double permintaan[], double persediaan[], int
tinggi,int lebar,String header,int angkaDiDepanKoma, int angkaDiBelakangKoma){
            String
format="%"+Integer.toString(angkaDiDepanKoma)+"."+Integer.toString(angkaDiBelakangKoma)+"f%s";
            String
format2="("+"%"+Integer.toString(angkaDiDepanKoma)+"."+Integer.toString(angkaDiBelakangKoma)+"f%s";
            System.out.println ("            "+header);
            System.out.println ("-------------------------------------------------");
            System.out.println("|           |       |   |   |       |Persediaan|");
            System.out.println ("-------------------------------------------------");
            for (int baris=0;baris<tinggi;baris++){
                System.out.print("|    |");
                for (int kolom=0;kolom<lebar;kolom++){
                    if(tabel2[baris][kolom]==0.0){
                        System.out.printf(format2,tabel[baris][kolom]," | ");
                    }else{
                        System.out.printf(format,tabel[baris][kolom]," | ");
                    }
                }
```

```
                    System.out.printf(format,persediaan[baris]," | ");
                    System.out.println ();
                }
                System.out.println ("-------------------------------------------------");
                System.out.print("|Permintaan|");
                for (int kolom=0;kolom<lebar;kolom++){
                    System.out.printf(format,permintaan[kolom]," | ");
                }
                System.out.println ();
                System.out.println ("-------------------------------------------------");
                System.out.println ();
        }
}
```

UNIVERSITAS BRAWIJAYA