1. algoritma program dinamis mundur untuk knapsack kapasitas (W) = 6.

Tahap 1:

$$f_4(y) = \max\{f_4(y),\ p_5 + f_4(y - w_5)\}$$

| $y$ | $\max\{f_4(y),\ p_5 + f_4(y - w_5)\}$ | | Solusi Optimum | |
|---|---|---|---|---|
| | $F_4(y)$ | $50 + f_4(y - 5)$ | $f_3(y)$ | $(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*)$ |
| 0 | 0 | 0 | 0 | (0, 0, 0, 0, 0) |
| 1 | 0 | 0 | 0 | (0, 0, 0, 0, 0) |
| 2 | 0 | 0 | 0 | (0, 0, 0, 0, 0) |
| 3 | 0 | 0 | 0 | (0, 0, 0, 0, 0) |
| 4 | 0 | 0 | 0 | (0, 0, 0, 0, 0) |
| 5 | 0 | 50 | 50 | (1, 0, 0, 0, 0) |
| 6 | 0 | 50 | 50 | (1, 0, 0, 0, 0) |

Tahap 2:

$$f_3(y) = \max\{f_3(y),\ p_4 + f_3(y - w_4)\}$$

| $y$ | | | Solusi Optimum | |
|---|---|---|---|---|
| | $f_3(y)$ | $40 + f_3(y - 4)$ | $f_2(y)$ | $(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*)$ |
| 0 | 0 | $40 + (-\infty) = -\infty$ | 0 | (0, 0, 0, 0, 0) |
| 1 | 0 | $40 + (-\infty) = -\infty$ | 0 | (0, 0, 0, 0, 0) |
| 2 | 0 | $40 + (-\infty) = -\infty$ | 0 | (0, 0, 0, 0, 0) |
| 3 | 0 | $40 + (-\infty) = -\infty$ | 0 | (0, 0, 0, 0, 0) |
| 4 | 0 | $40 + 0 = 40$ | 40 | (0, 0, 0, 1, 0) |
| 5 | 50 | $40 + (-\infty) = -\infty$ | 50 | (0, 0, 0, 0, 1) |
| 6 | 50 | $40 + (-\infty) = -\infty$ | 50 | (0, 0, 0, 0, 1) |

Tahap 3:

$$f_2(y) = \max\{f_2(y),\ p_3 + f_2(y - w_3)\}$$

| $y$ | $\max\{f_2(y),\ p_3 + f_2(y - w_3)\}$ | | Solusi Optimum | |
|---|---|---|---|---|
| | $f_2(y)$ | $15 + f_2(y - 1)$ | $f_1(y)$ | $(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*)$ |
| 0 | 0 | $0 + (-\infty) = -\infty$ | 0 | (0, 0, 0, 0, 0) |
| 1 | 0 | $15 + 0 = 15$ | 15 | (0, 0, 1, 0, 0) |
| 2 | 0 | $15 + 0 = 15$ | 15 | (0, 0, 1, 0, 0) |
| 3 | 0 | $15 + 0 = 15$ | 15 | (0, 0, 1, 0, 0) |
| 4 | 40 | $15 + (-\infty) = -\infty$ | 40 | (0, 0, 0, 1, 0) |
| 5 | 50 | $15 + 40 = 55$ | 55 | (0, 0, 1, 1, 0) |
| 6 | 50 | $15 + 50 = 65$ | 65 | (0, 0, 1, 0, 1) |

Tahap 4:

$$f_1(y) = \max\{f_1(y),\ p_2 + f_1(y - w_2)\}$$

| $y$ | $\max\{f_1(y),\ p_2 + f_1(y - w_2)\}$ | | Solusi Optimum | |
|---|---|---|---|---|
| | $f_1(y)$ | $20 + f_1(y - 2)$ | $f_0(y)$ | $(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*)$ |

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 20 + (-∞) = -∞ | 0 | (0, 0, 0, 0, 0) |
| 1 | 15 | 20 + (-∞) = -∞ | 15 | (0, 0, 1, 0, 0) |
| 2 | 15 | 20 + 0 = 20 | 20 | (0, 1, 0, 0, 0) |
| 3 | 15 | 20 + 15 = 35 | 35 | (0, 1, 1, 0, 0) |
| 4 | 40 | 20 + (-∞) = -∞ | 40 | (0, 0, 0, 1, 0) |
| 5 | 55 | 20 + (-∞) = -∞ | 55 | (0, 0, 0, 0, 1) |
| 6 | 65 | 20 + (-∞) = -∞ | 65 | (0, 0, 1, 0, 1) |

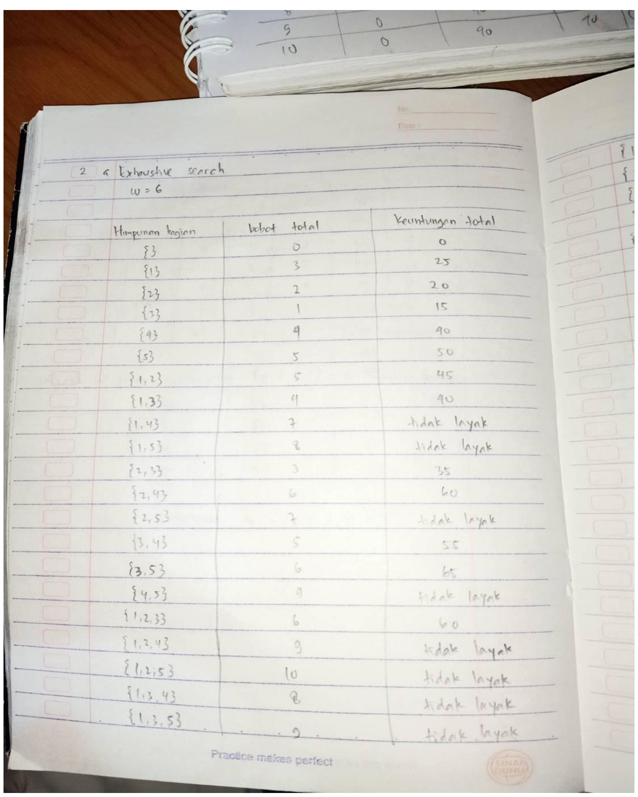
Tahap 5:

$f_0(y) = \max\{f_0(y), p_1 + f_0(y - w_1)\}$

| $y$ | $\max\{f_0(y), p_1 + f_0(y - w_1)\}$ | | Solusi Optimum | |
|---|---|---|---|---|
| | $f_1(y)$ | $20 + f_1(y - 2)$ | $f_0(y)$ | $(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*)$ |
| 0 | 0 | 25 + (-∞) = -∞ | 0 | (0, 0, 0, 0, 0) |
| 1 | 15 | 25 + (-∞) = -∞ | 15 | (0, 0, 1, 0, 0) |
| 2 | 20 | 25 + (-∞) = -∞ | 20 | (0, 1, 0, 0, 0) |
| 3 | 35 | 25 + (-∞) = -∞ | 35 | (0, 1, 1, 0, 0) |
| 4 | 40 | 25 + (-∞) = -∞ | 40 | (0, 0, 0, 1, 0) |
| 5 | 50 | 25 + (-∞) = -∞ | 50 | (0, 0, 0, 0, 1) |
| 6 | 65 | 25 + (-∞) = -∞ | 65 | (0, 0, 1, 0, 1) |

Solusi Optimum X = (0,0,1,0,1) dengan profit sebesar 65.

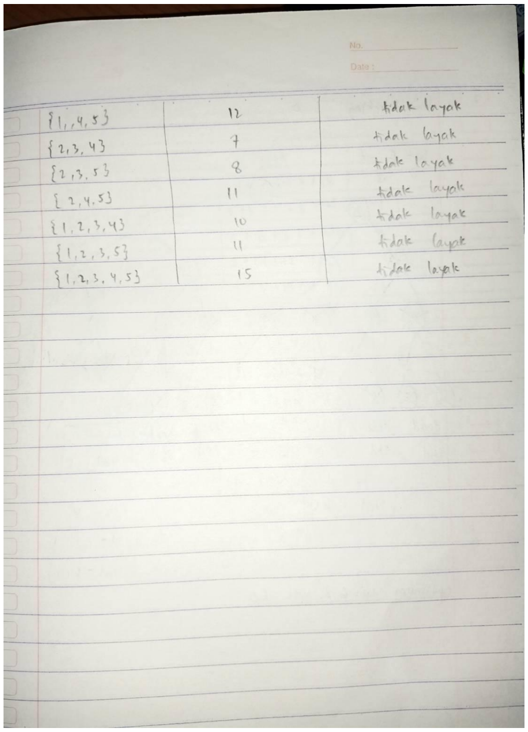
2. A. exhaustive search



2 a Exhaustive search

w = 6

| Himpunan bagian | bobot total | keuntungan total |
|---|---|---|
| {} | 0 | 0 |
| {1} | 3 | 25 |
| {2} | 2 | 20 |
| {3} | 1 | 15 |
| {4} | 4 | 40 |
| {5} | 5 | 50 |
| {1,2} | 5 | 45 |
| {1,3} | 4 | 40 |
| {1,4} | 7 | tidak layak |
| {1,5} | 8 | tidak layak |
| {2,3} | 3 | 35 |
| {2,4} | 6 | 60 |
| {2,5} | 7 | tidak layak |
| {3,4} | 5 | 55 |
| {3,5} | 6 | 65 |
| {4,5} | 9 | tidak layak |
| {1,2,3} | 6 | 60 |
| {1,2,4} | 9 | tidak layak |
| {1,2,5} | 10 | tidak layak |
| {1,3,4} | 8 | tidak layak |
| {1,3,5} | 9 | tidak layak |



No.

Date :

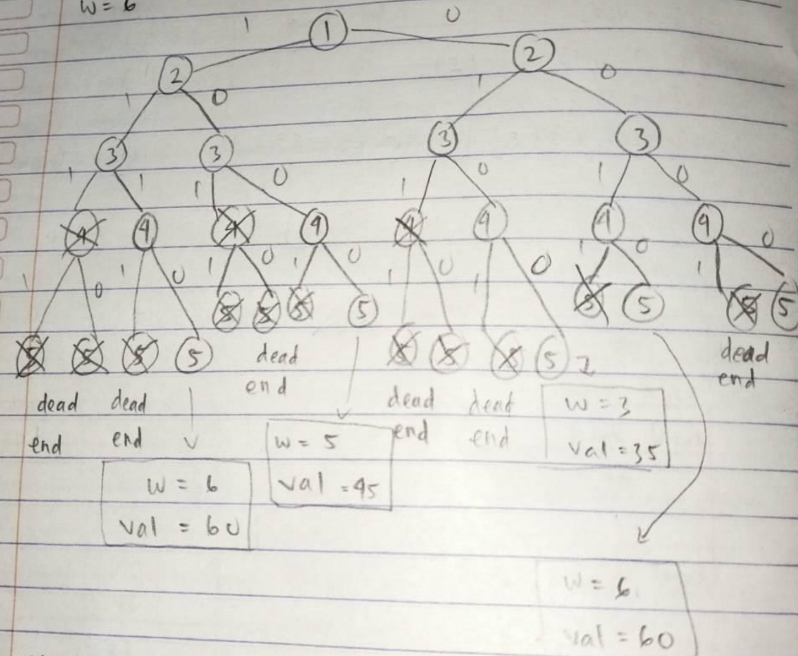
| | | |
|---|---|---|
| {1, ,4, 5} | 12 | tidak layak |
| {2,3, 4} | 7 | tidak layak |
| {2,3, 5} | 8 | tidak layak |
| {2,4, 5} | 11 | tidak layak |
| {1, 2, 3, 4} | 10 | tidak layak |
| {1, 2, 3, 5} | 11 | tidak layak |
| {1, 2, 3, 4, 5} | 15 | tidak layak |

B. Backtracking

No.
Date :

2 b backtracking 1. A

W = 6



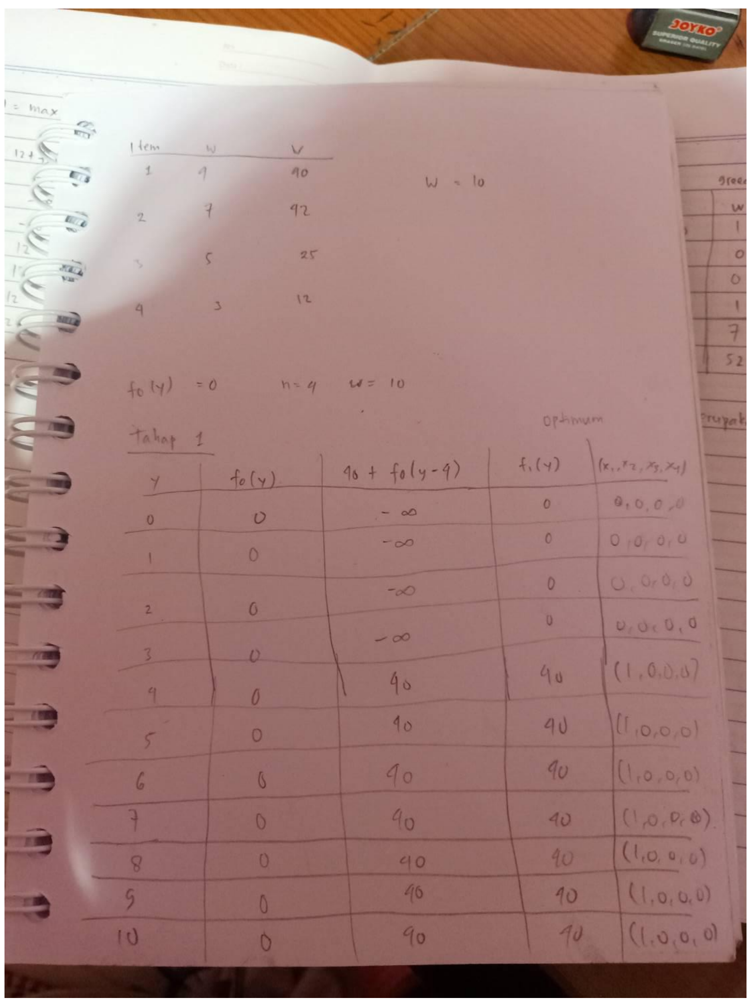

optimum w = 6 & val = 60

3. Algoritma program dinamis maju dengan kapasitas w = 10.

| Item | weight | Value | value/weight |
|---|---|---|---|
| 1 | 4 | 40 | 10 |
| 2 | 7 | 42 | 6 |
| 3 | 5 | 25 | 5 |
| 4 | 3 | 12 | 4 |

Gambar 1.B.

Tahap 1 :



| Item | w | v |
|---|---|---|
| 1 | 4 | 40 |
| 2 | 7 | 42 |
| 3 | 5 | 25 |
| 4 | 3 | 12 |

W = 10

$f_0(y) = 0$ n = 4 w = 10

Tahap 1

| y | $f_0(y)$ | $40 + f_0(y-4)$ | $f_1(y)$ | optimum $(x_1, x_2, x_3, x_4)$ |
|---|---|---|---|---|
| 0 | 0 | $-\infty$ | 0 | 0, 0, 0, 0 |
| 1 | 0 | $-\infty$ | 0 | 0, 0, 0, 0 |
| 2 | 0 | $-\infty$ | 0 | 0, 0, 0, 0 |
| 3 | 0 | $-\infty$ | 0 | 0, 0, 0, 0 |
| 4 | 0 | 40 | 40 | (1, 0, 0, 0) |
| 5 | 0 | 40 | 40 | (1, 0, 0, 0) |
| 6 | 0 | 40 | 40 | (1, 0, 0, 0) |
| 7 | 0 | 40 | 40 | (1, 0, 0, 0) |
| 8 | 0 | 40 | 40 | (1, 0, 0, 0) |
| 9 | 0 | 40 | 40 | (1, 0, 0, 0) |
| 10 | 0 | 40 | 40 | (1, 0, 0, 0) |

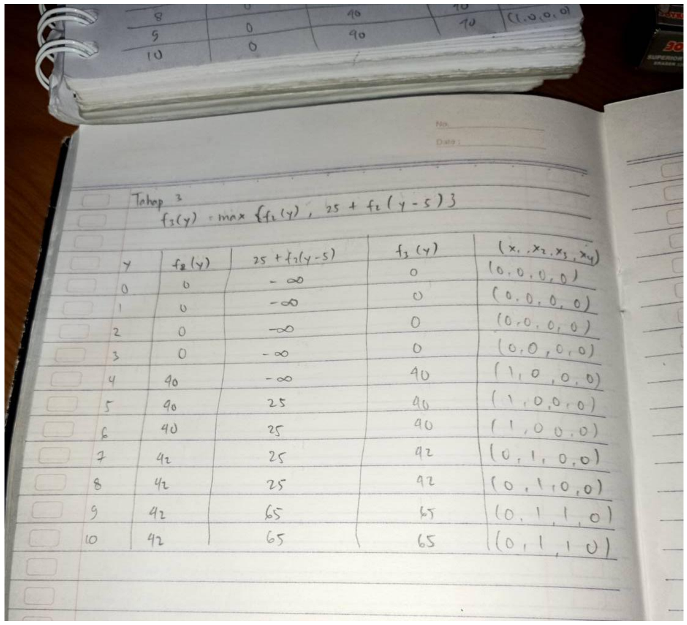
Tahap 2:

Tahap 2

$$f_2(y) = \max\{f_1(y),\ 42 + f_1(y-7)\}$$

| $y$ | $f_1(y)$ | $42 + f_1(y-7)$ | $f_2(y)$ | $(x_1, x_2, x_3, x_4)$ |
|---|---|---|---|---|
| 0 | 0 | $-\infty$ | 0 | (0,0,0,0) |
| 1 | 0 | $-\infty$ | 0 | (0,0,0,0) |
| 2 | 0 | $-\infty$ | 0 | (0,0,0,0) |
| 3 | 0 | $-\infty$ | 0 | (0,0,0,0) |
| 4 | 40 | $-\infty$ | 40 | (1,0,0,0) |
| 5 | 40 | $-\infty$ | 40 | (1,0,0,0) |
| 6 | 40 | $-\infty$ | 40 | (1,0,0,0) |
| 7 | 40 | 42 | 42 | (0,1,0,0) |
| 8 | 40 | 42 | 42 | (0,1,0,0) |
| 9 | 40 | 42 | 42 | (0,1,0,0) |
| 10 | 40 | 42 | 42 | (0,1,0,0) |

Tahap 3 :

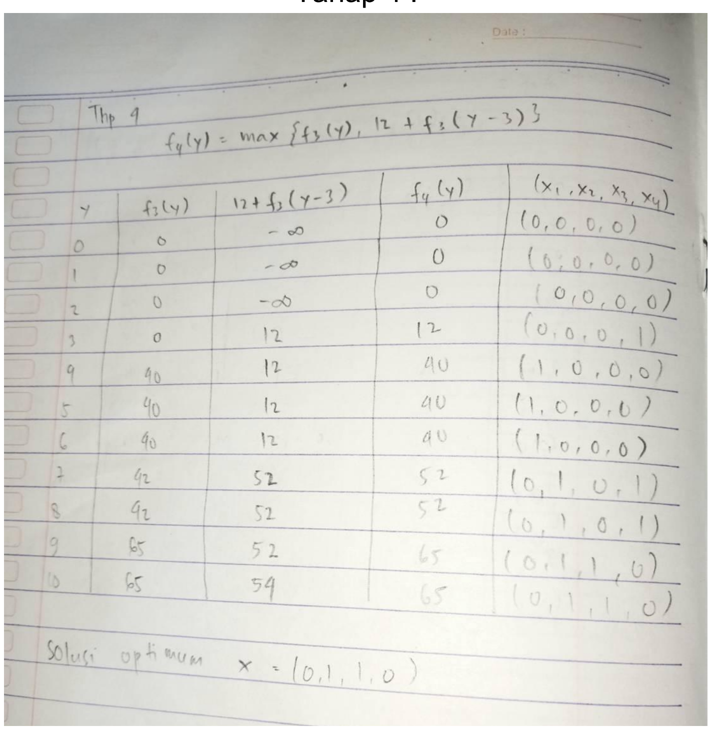


Tahap 3

$f_3(y) = \max\{f_2(y),\ 25 + f_2(y-5)\}$

| $y$ | $f_2(y)$ | $25 + f_2(y-5)$ | $f_3(y)$ | $(x_1, x_2, x_3, x_4)$ |
|---|---|---|---|---|
| 0 | 0 | $-\infty$ | 0 | (0, 0, 0, 0) |
| 1 | 0 | $-\infty$ | 0 | (0, 0, 0, 0) |
| 2 | 0 | $-\infty$ | 0 | (0, 0, 0, 0) |
| 3 | 0 | $-\infty$ | 0 | (0, 0, 0, 0) |
| 4 | 40 | $-\infty$ | 40 | (1, 0, 0, 0) |
| 5 | 40 | 25 | 40 | (1, 0, 0, 0) |
| 6 | 40 | 25 | 40 | (1, 0, 0, 0) |
| 7 | 42 | 25 | 42 | (0, 1, 0, 0) |
| 8 | 42 | 25 | 42 | (0, 1, 0, 0) |
| 9 | 42 | 65 | 65 | (0, 1, 1, 0) |
| 10 | 42 | 65 | 65 | (0, 1, 1, 0) |

Tahap 4 :



Thp 4

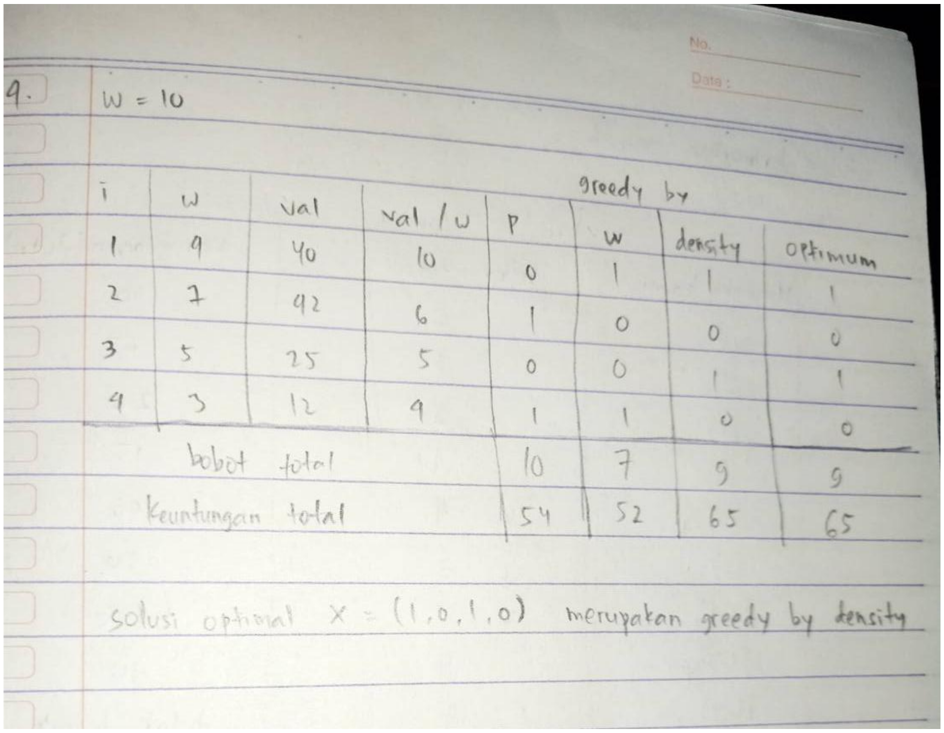
$f_4(y) = \max\{f_3(y),\ 12 + f_3(y-3)\}$

| $y$ | $f_3(y)$ | $12 + f_3(y-3)$ | $f_4(y)$ | $(x_1, x_2, x_3, x_4)$ |
|---|---|---|---|---|
| 0 | 0 | $-\infty$ | 0 | (0, 0, 0, 0) |
| 1 | 0 | $-\infty$ | 0 | (0, 0, 0, 0) |
| 2 | 0 | $-\infty$ | 0 | (0, 0, 0, 0) |
| 3 | 0 | 12 | 12 | (0, 0, 0, 1) |
| 4 | 40 | 12 | 40 | (1, 0, 0, 0) |
| 5 | 40 | 12 | 40 | (1, 0, 0, 0) |
| 6 | 40 | 12 | 40 | (1, 0, 0, 0) |
| 7 | 42 | 52 | 52 | (0, 1, 0, 1) |
| 8 | 42 | 52 | 52 | (0, 1, 0, 1) |
| 9 | 65 | 52 | 65 | (0, 1, 1, 0) |
| 10 | 65 | 54 | 65 | (0, 1, 1, 0) |

Solusi optimum x = (0, 1, 1, 0)

Solusi optimum x =(0, 1, 0, 1) dengan profit sebesar 65.

4, greedy



9. W = 10

| i | w | val | val / w | greedy by p | w | density | optimum |
|---|---|---|---|---|---|---|---|
| 1 | 9 | 40 | 10 | 0 | 1 | 1 | 1 |
| 2 | 7 | 42 | 6 | 1 | 0 | 0 | 0 |
| 3 | 5 | 25 | 5 | 0 | 0 | 1 | 1 |
| 4 | 3 | 12 | 4 | 1 | 1 | 0 | 0 |
| bobot total | | | | 10 | 7 | 9 | 9 |
| keuntungan total | | | | 54 | 52 | 65 | 65 |

solusi optimal x = (1,0,1,0) merupakan greedy by density

Solusi optimal x = (1, 0, 1, 0) yang merupakan greedy by density.

5. Solusi 8-ratu :

a. Brute force

Pada setiap baris hanya perlu menempatkan ratu pada setiap kolom yang berbeda, dalam vektor 8-tuple:

X = (x1 , x2 , ... , x8)

Jumlah permutasi bilangan 1 sampai 8 adalah P(1, 8) = 8! = 40.320 buah.

<u>procedure Ratu</u>

{Mencari semua solusi penempatan delapan ratu pada petak-petak papan catur yang berukuran 8 x 8 }

<u>Deklarasi</u>

```
X : vektor_solusi

n,i : integer


    Algoritma:

n←40320{ Jumlah permutasi (1, 2, ..., 8) }

i←1

repeat

    X←Permutasi(8) { Bangn permutasi (1, 2, ..., 8) }

        { periksa apakah X merupakan solusi }

    if Solusi(X) then

        TulisSolusi(X)

    endif

    i←i+1 { ulangi untuk permutasi berikutnya }

until i > n
```

b. Backtracking

i. Inisialisasi x[1], x[2], ..., x[N] dengan 0

```
for i←N to n do

    x[i] ← 0

endfor2.
```

ii. Kemudian panggil prosedur N_RATU_R

procedure N_RATU_I(input N:integer)

{ Mencetak semua solusi penempatan N buah ratu pada petak papan catur N x N tanpa melanggar kendala; versi iteratif

Masukan: N = jumlah ratu

Keluaran: semua solusi x = (x[1], x[2], ..., x[N]) dicetak ke layar.

}

Deklarasi

k : integer

Algoritma:

k←1{mulai pada baris catur ke-1}

x[1]←0 {inisialisasi kolom dengan 0}

while k > 0 do

x[k]←x[k]+1 {pindahkan ratu ke kolom berikutnya}

while (x[k] ≤ N) and (not TEMPAT(k)) do

{periksa apakah ratu dapat ditempatkan pada kolom x[k]}

x[k]:=x[k] + 1

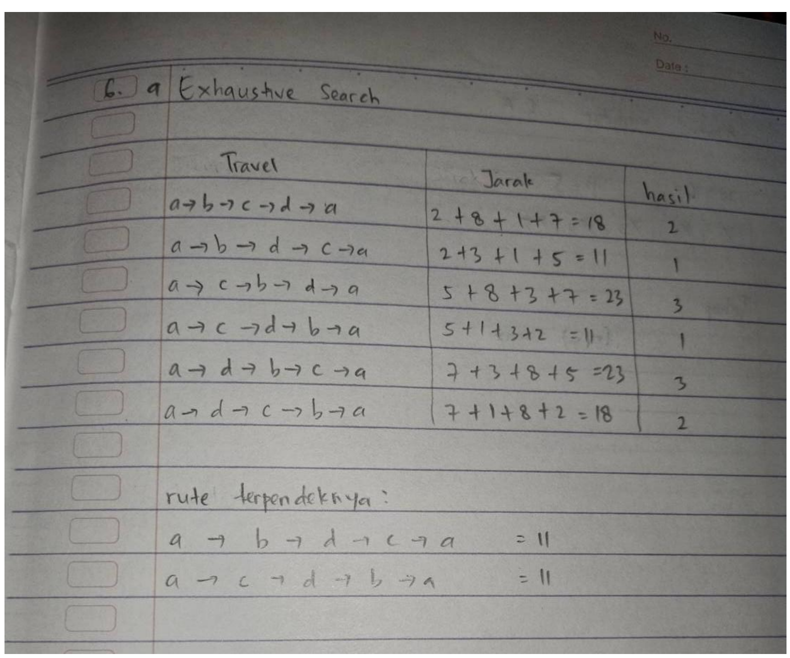
endwhile

{x[k] > n or TEMPAT(k) }

if x[k]≤ n then{ kolom penempatan ratu ditemukan }

```
            if k=N then{ apakah solusi sudah lengkap?}
                CetakSolusi(x,N){ cetak solosi}
            else
                k←k+1{pergi ke baris berikutnya}
                x[k]←0 {inisialisasi kolom dengan 0}
            endif
        else
            k←k-1 { runut-balik ke baris sebelumnya}
        endif
    endwhile
    { k = 0 }
```

6. A. Exhaustive search


6. a Exhaustive Search

| Travel | Jarak | hasil |
|---|---|---|
| a→b→c→d→a | 2+8+1+7 = 18 | 2 |
| a→b→d→c→a | 2+3+1+5 = 11 | 1 |
| a→c→b→d→a | 5+8+3+7 = 23 | 3 |
| a→c→d→b→a | 5+1+3+2 = 11 | 1 |
| a→d→b→c→a | 7+3+8+5 = 23 | 3 |
| a→d→c→b→a | 7+1+8+2 = 18 | 2 |

rute terpendeknya:

a → b → d → c → a = 11

a → c → d → b → a = 11

b. Greedy algorithm

Dengan greedy algorithm mencari yang terdekat terlebih dahulu:

a. Kota A terhubung dengan kota B (2) dan C (5), lalu pilih jarak terpendek yaitu ke Kota B (2).

b. Dari Kota B terhubung dengan Kota C (8) dan D (3), pilih jarak terpendek yaitu kota D ( 3)

c. Kemudian dari Kota D terhubung dengan Kota A (7) dan kota C (1), dikarenakan kota A adalah kota titik awal, maka pergi ke C dahulu karena belum di kunjungi maka jarak ke kota C (1)

d. Dan setelah di kota C karena seluruh kota sudah dikunjungi, kembali ke kota A (5).

e. Solusi optimum dengan greedy yaitu A – B – D – C – A dengan total 11.