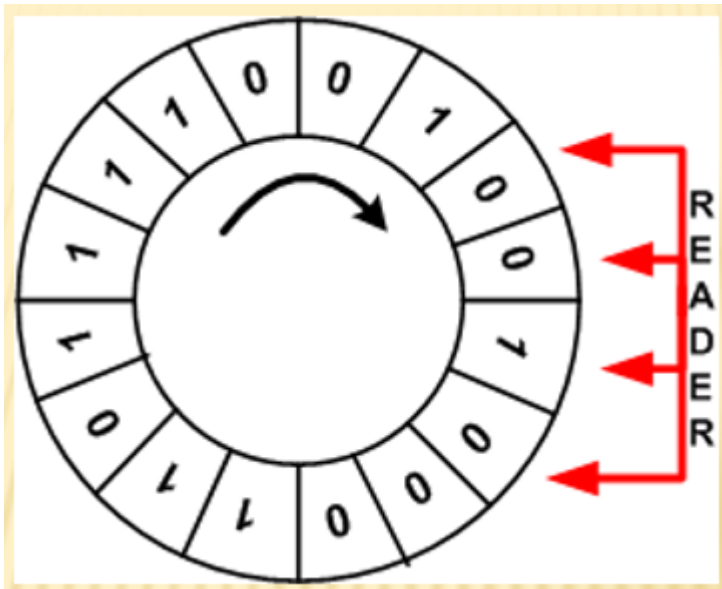


TUGAS MINGGUAN 5

Nama : Muhammad Daffa Rizky Sutrisno

NRP : 5025231207

Teleprinter :



1. Posisi sebuah rotating drum semestinya dpt dikenali dr signal biner yg dihasilkan melalui pembacaan reader. Gambar di atas adalah contoh desain rotating drum yg tidak efisien krn pembacaan per 4-digit biner akan dpt menghasilkan susunan string yg sama. Buatlah desain rotating drum yg lebih efisien jk digunakan sebuah piringan yg memiliki $2r$ permukaan baca dg r reader. Asumsikan $r = 4$.

Desain rotating drum yang efisien harus memastikan bahwa setiap pembacaan oleh 4 reader menghasilkan string biner unik tanpa pengulangan, sehingga dapat dengan mudah mengenali posisi drum. Hal ini dapat dicapai dengan menggunakan prinsip kode siklik (cyclic code), seperti **de Bruijn sequence**.

a. Masalah pada Desain Awal

Pada desain awal rotating drum, pembacaan oleh 4 reader menghasilkan string biner yang tidak unik. Kombinasi tertentu dapat muncul di lebih dari satu posisi, yang menyebabkan ambiguitas dalam mengenali posisi drum.

b. Tujuan Desain Efisien

Rotating drum yang efisien harus:

- Menghasilkan kombinasi biner unik untuk setiap posisi yang dapat dikenali oleh 4 reader.
- Menghindari pengulangan kombinasi dalam satu siklus penuh.
- Menggunakan jumlah bit minimal untuk mencakup semua kombinasi unik.

c. Solusi dengan de Bruijn Sequence

De Bruijn sequence adalah urutan biner siklik yang mencakup semua kombinasi unik sepanjang r -bit dalam satu siklus penuh.

Karakteristik de Bruijn Sequence :

- Untuk $r = 4$, terdapat $2^r = 16$ kombinasi unik biner 4-bit (0000 hingga 1111).
- Panjang de Bruijn sequence adalah 16, cukup untuk mencakup semua kombinasi unik.
- Contoh sequence : 0000111101100101.

d. Langkah-Langkah Desain

Jalur Baca :

- Jalur baca drum disusun mengikuti pola de Bruijn sequence.
- Kombinasi output dari 4 reader akan unik untuk setiap posisi.

Reader dan Output:

- Gunakan 4 reader yang membaca secara simultan.
- Contoh pembacaan:
 - Posisi awal: 0000
 - Posisi 1: 0001
 - Posisi 2: 0011
 - Posisi 3: 0111
 - ... hingga posisi terakhir: 1000.

Dengan desain ini, posisi drum selalu dapat dikenali secara unik tanpa redundansi.

e. Representasi Graf de Bruijn

Desain ini dapat direpresentasikan sebagai graf siklik terarah (directed cyclic graph), dengan langkah-langkah berikut:

Node dan Edge :

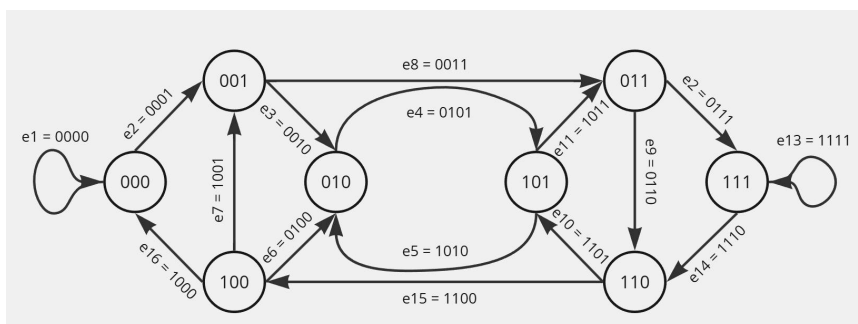
1. *Node*: Representasi string biner sepanjang $r - 1 = 3$ bit. Total node: $2^{r-1} = 8$.
 - Node: 000, 001, 010, 011, 100, 101, 110, 111.
2. *Edge*: Setiap node memiliki 2 edge keluar, yang dihasilkan dengan menambahkan 0 atau 1 ke akhir string.
 - Contoh:
 - Dari 000:
 - Tambah 0 → Edge ke 000.
 - Tambah 1 → Edge ke 001.

De Bruijn Sequence dari Graf :

- Dengan melintasi graf secara siklik, sequence 00001111011001010000111101100101 akan dihasilkan, mencakup semua kombinasi unik 4-bit.

f. Visualisasi Graf dan Desain

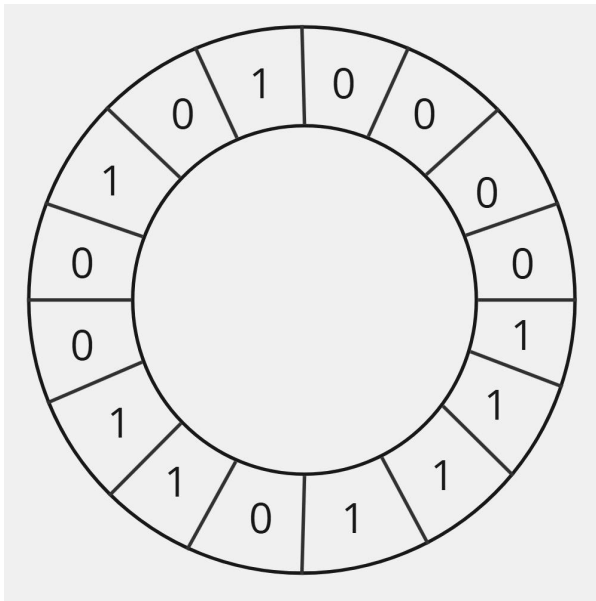
Berikut adalah graf De Bruijn untuk $r = 4$:



Penjelasan Visual:

- *Node*: Representasi semua kombinasi 3-bit (000 hingga 111).
- *Edge*: Menghubungkan node berdasarkan transisi dengan menambahkan 0 atau 1.
- *Arah Panah*: Menunjukkan hubungan antar-node secara siklik.

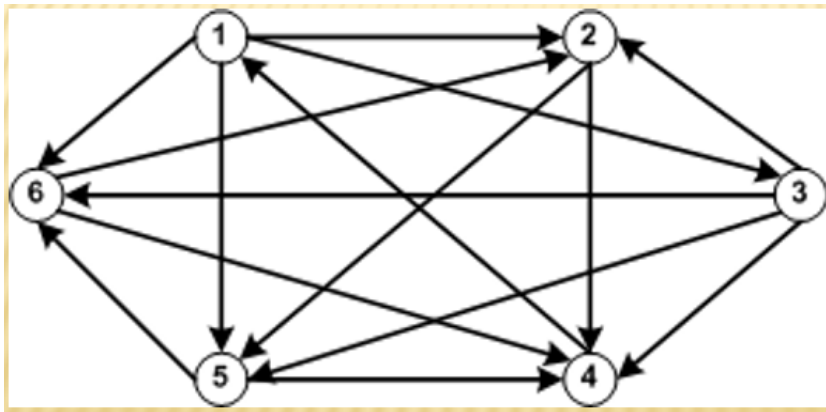
Berikut adalah desain Teleprinters akhir untuk $r = 4$:



g. Keuntungan Desain Baru

1. *Efisiensi* : Tidak ada kombinasi biner yang berulang, memastikan setiap posisi drum unik.
2. *Panjang Minimum* : Panjang jalur baca hanya $2^r = 16$, sesuai kebutuhan minimum.
3. *Skalabilitas* : Pendekatan ini dapat diperluas untuk $r > 4$.

Graph :



2. Sebuah turnamen bola volley di Jurusan Teknik Informatika ITS mempertandingkan 6 tim, dimana setiap tim akan bertanding dg semua tim lainnya. Jika hasil pertandingan antara masing2 tim diberikan dalam bentuk digraph di bawah, maka selanjutnya bagaimana cara anda menyusun ranking peserta? Graph di samping menunjukkan bahwa tim-1 mengalahkan tim-2. tim-2 dikalahkan tim tim-3. tim-3 mengalahkan tim-4. dst.

Untuk menyusun ranking peserta berdasarkan hasil pertandingan yang diberikan dalam bentuk digraf, kita dapat menggunakan langkah-langkah berikut :

a. Pendekatan

1. Interpretasi Digraf:
 - Pada graf tersebut, setiap simpul (node) mewakili tim, dan setiap arah panah menunjukkan kemenangan. Misalnya, jika ada panah dari node i ke node j, berarti tim i mengalahkan tim j.
2. Hitung Indegree atau Outdegree:
 - Outdegree untuk setiap simpul menunjukkan jumlah kemenangan yang diraih tim tersebut.
 - Indegree menunjukkan jumlah kekalahan tim tersebut.
3. Urutkan Berdasarkan Outdegree:
 - Tim dengan outdegree paling tinggi memiliki jumlah kemenangan terbanyak dan berada di peringkat atas. Jika ada tim dengan outdegree sama, Anda dapat mempertimbangkan menggunakan tiebreak (misalnya berdasarkan jumlah kekalahan/indegree lebih kecil lebih baik).
4. Proses Perhitungan:
 - Hitung jumlah panah keluar dari setiap simpul (outdegree).
 - Urutkan tim berdasarkan nilai outdegree dari besar ke kecil.

b. Identifikasi Panah pada Digraf

Dari graf, kita menghitung jumlah panah keluar (outdegree) untuk setiap tim. Panah keluar menunjukkan berapa kali tim tersebut menang.

Panah Keluar (Outdegree)

- **Tim 1:** Panah keluar menuju Tim 2, 3, 4, dan 5. Jadi, outdegree Tim 1 = 4.
- **Tim 2:** Panah keluar menuju Tim 3, 4, dan 5. Jadi, outdegree Tim 2 = 3.
- **Tim 3:** Panah keluar menuju Tim 4, 5, dan 6. Jadi, outdegree Tim 3 = 3.
- **Tim 4:** Panah keluar menuju Tim 5 dan 6. Jadi, outdegree Tim 4 = 2.
- **Tim 5:** Panah keluar menuju Tim 6. Jadi, outdegree Tim 5 = 1.
- **Tim 6:** Tidak memiliki panah keluar (tidak mengalahkan tim mana pun). Jadi, outdegree Tim 6 = 0.

c. Susun Berdasarkan Outdegree

Urutkan tim berdasarkan nilai outdegree dari besar ke kecil:

- **Tim 1:** Outdegree = 4
- **Tim 2:** Outdegree = 3
- **Tim 3:** Outdegree = 3
- **Tim 4:** Outdegree = 2
- **Tim 5:** Outdegree = 1
- **Tim 6:** Outdegree = 0

d. Tangani Kasus Seri (Tiebreaker)

Karena **Tim 2** dan **Tim 3** memiliki outdegree yang sama (3), kita periksa jumlah kekalahan (indegree) sebagai tiebreaker.

Panah Masuk (Indegree)

- **Tim 1:** Panah masuk dari Tim 6. Jadi, indegree Tim 1 = 1.
- **Tim 2:** Panah masuk dari Tim 1. Jadi, indegree Tim 2 = 1.
- **Tim 3:** Panah masuk dari Tim 1 dan 2. Jadi, indegree Tim 3 = 2.
- **Tim 4:** Panah masuk dari Tim 1, 2, dan 3. Jadi, indegree Tim 4 = 3.
- **Tim 5:** Panah masuk dari Tim 1, 2, 3, dan 4. Jadi, indegree Tim 5 = 4.
- **Tim 6:** Panah masuk dari Tim 3, 4, dan 5. Jadi, indegree Tim 6 = 3.

Tiebreaker:

- Tim 2 memiliki indegree = 1 (lebih kecil).
- Tim 3 memiliki indegree = 2.

Jadi, **Tim 2** berada di peringkat lebih tinggi dibandingkan Tim 3.

e. Hasil Akhir

Urutan ranking peserta berdasarkan hasil pertandingan adalah:

1. **Tim 1** (Outdegree = 4, Indegree = 1)
2. **Tim 2** (Outdegree = 3, Indegree = 1)
3. **Tim 3** (Outdegree = 3, Indegree = 2)
4. **Tim 4** (Outdegree = 2, Indegree = 3)
5. **Tim 5** (Outdegree = 1, Indegree = 4)
6. **Tim 6** (Outdegree = 0, Indegree = 3)

adjustment time :

	J_1	J_2	J_3	J_4	J_5	J_6
J_1	0	5	3	4	2	1
J_2	1	0	1	2	3	2
J_3	2	5	0	1	2	3
J_4	1	4	4	0	1	2
J_5	1	3	4	5	0	5
J_6	4	4	2	3	1	0

3. Terdapat 6 buah pekerjaan ($J_1, J_2, J_3, J_4, J_5, J_6$) yg harus diselesaikan oleh sebuah mesin. Karakteristik yg digunakan adalah mesin tsb memerlukan waktu (adjustment time) untuk penyesuaian dg bentuk pekerjaan yg akan dikerjakan selanjutnya. Misalkan setelah mengerjakan J_i , maka mesin tsb memerlukan waktu selama t_{ij} , sebelum mesin tsb mengerjakan pekerjaan J_j . Carilah urutan pekerjaan yg dapat meminimalkan total adjustment time. Masalahnya adalah sampai saat ini belum tersedia metode untuk menyelesaikan permasalahan di atas secara optimal. Namun anda dapat menggunakan Teorema Redei sebagai guidance.

Berikut adalah langkah-langkah untuk menyelesaikan permasalahan menggunakan Teorema Redei sebagai panduan :

a. Teorema Redei dan Modifikasi Masalah

Teorema Redei berlaku untuk graf berarah tanpa siklus (DAG). Karena matriks waktu penyesuaian (t_{ij}) mencerminkan graf lengkap berarah (complete directed graph), kita perlu memodifikasinya agar menjadi graf DAG.

Modifikasi:

- Untuk setiap pasangan pekerjaan (J_i, J_j), kita hanya mempertahankan arah $J_i \rightarrow J_j$ jika $t_{ij} < t_{ji}$. Dengan demikian, kita menghilangkan arah balik yang lebih besar dan menciptakan graf tanpa siklus.

b. Bangun Graf DAG

Selanjutnya bangun graf DAG dari matriks waktu yang diberikan. Matriks waktu adalah :

	J ₁	J ₂	J ₃	J ₄	J ₅	J ₆
J ₁	0	5	3	4	2	1
J ₂	1	0	1	2	3	2
J ₃	2	5	0	1	2	3
J ₄	1	4	4	0	1	2
J ₅	1	3	4	5	0	5
J ₆	4	4	2	3	1	0

Kita hanya mempertahankan arah $J_i \rightarrow J_j$ jika $t_{ij} < t_{ji}$.

Contoh:

- Antara J₁ dan J₂ :
 - $t_{12} = 5, t_{21} = 1$. Karena $t_{21} < t_{12}$, maka kita pertahankan arah $J_2 \rightarrow J_1$.
- Antara J₁ dan J₃:
 - $t_{13} = 3, t_{31} = 2$. Karena $t_{31} < t_{13}$, maka kita pertahankan arah $J_3 \rightarrow J_1$.

Setelah melakukan hal ini untuk semua pasangan pekerjaan, kita mendapatkan graf DAG berikut:

- $J_2 \rightarrow J_1$
- $J_3 \rightarrow J_1$
- $J_3 \rightarrow J_2$
- $J_4 \rightarrow J_1$
- $J_4 \rightarrow J_2$
- $J_4 \rightarrow J_3$
- $J_5 \rightarrow J_1$
- $J_5 \rightarrow J_2$
- $J_5 \rightarrow J_3$
- $J_5 \rightarrow J_4$
- $J_6 \rightarrow J_1$
- $J_6 \rightarrow J_2$
- $J_6 \rightarrow J_3$
- $J_6 \rightarrow J_4$
- $J_6 \rightarrow J_5$

c. Temukan Urutan Topologis

Graf DAG memungkinkan kita menemukan urutan topologis menggunakan algoritma seperti *Kahn's Algorithm* atau DFS. Urutan topologis memastikan bahwa semua pekerjaan dilakukan dalam urutan yang meminimalkan konflik waktu.

Prosesnya :

1. Cari simpul dengan indegree 0 (simpul tanpa panah masuk).
2. Masukkan simpul tersebut ke urutan.
3. Hapus simpul dari graf (hapus semua panah keluar dari simpul tersebut).
4. Ulangi hingga semua simpul diurutkan.

Untuk graf ini, urutan topologisnya adalah:

$$J6 \rightarrow J5 \rightarrow J4 \rightarrow J3 \rightarrow J2 \rightarrow J1$$

d. Hitung Total Adjustment Time

Dari urutan topologis $J6 \rightarrow J5 \rightarrow J4 \rightarrow J3 \rightarrow J2 \rightarrow J1$, kita hitung total waktu penyesuaian berdasarkan matriks t_{ij} :

- $t_{65} = 1$
- $t_{54} = 5$
- $t_{43} = 4$
- $t_{32} = 5$
- $t_{21} = 1$

Total waktu penyesuaian = $t_{65} + t_{54} + t_{43} + t_{32} + t_{21} = 1 + 5 + 4 + 5 + 1 = 16$.

e. Hasil Akhir

Urutan pekerjaan berdasarkan Teorema Redei untuk meminimalkan total waktu penyesuaian adalah:

$$J6 \rightarrow J5 \rightarrow J4 \rightarrow J3 \rightarrow J2 \rightarrow J1$$

Dengan total waktu penyesuaian = 16.