

BEBERAPA APLIKASI GRAF

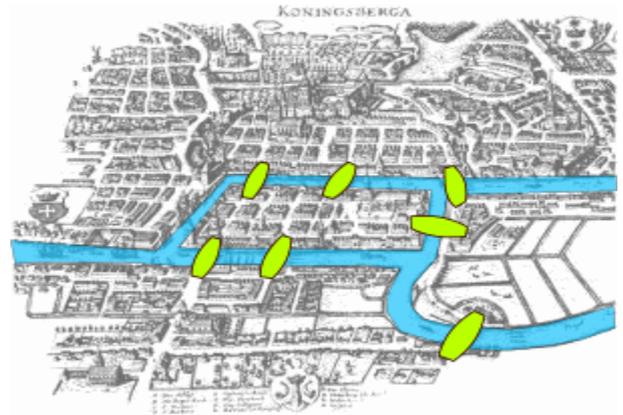
Bobby H. Suryanaga - 13508022

Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Sumber Sugih No. 53, Bandung
bobbyhsuryanaga@hotmail.com

ABSTRAK

Makalah ini membahas tentang graf dan pengaplikasiannya dalam kehidupan nyata. Dalam makalah ini akan dibahas beberapa aplikasi dari graf, yaitu untuk pengaturan jalan raya, pemodelan system basis data tersebar, penggambaran hubungan dalam ekosistem, pengaturan jadwal dengan pewarnaan graf

Kata kunci: graf, aplikasi, graf berbobot, graf berarah, pewarnaan graf.



Gambar 1. Permasalahan Jembatan Königsberg

1. PENDAHULUAN

Graf memiliki banyak aplikasi dalam kehidupan kita sehari-hari. Beberapa aplikasi tersebut adalah pencarian lintasan terpendek, persoalan pedagang keliling, persoalan tukang pos Cina, pedesainan chip, penggambaran hubungan dalam suatu ekosistem, pemodelan basis data tersebar, perancangan jadwal, dan penggambaran jalan raya untuk pengaturan lalu lintas.

Pada makalah ini akan dibahas beberapa aplikasi dari graf, yaitu pencarian lintasan terpendek, pengaturan jalan raya, pemodelan basis data tersebar, penggambaran hubungan dalam ekosistem, serta penyusunan jadwal menggunakan pewarnaan graf.

2. GRAF

Graf adalah pasangan antara himpunan simpul dan sisi. Sebuah graf tidak mungkin memiliki 0 buah simpul (*vertice/node*), tetapi mungkin memiliki 0 buah sisi (*edge/arc*). Graf yang memiliki sebuah simpul tanpa sisi disebut graf trivial.[1] Teori graf pertama kali muncul pada pemecahan permasalahan jembatan Königsberg.

Jembatan tersebut menghubungkan sebuah daratan yang dipisahkan oleh sungai sehingga membentuk seperti pulau. Jembatan Königsberg menghubungkan daratan tersebut dengan daratan di luarnya.

Permasalahannya adalah apakah mungkin melalui ketujuh jembatan itu masing-masing sekali dan kembali ke tempat asal mula keberangkatan.

Permasalahan tersebut dapat dipecahkan oleh seorang matematikawan Swiss, L. Euler dengan menggunakan graf untuk pemodelan masalah tersebut.

Jawaban yang dikemukakan Euler adalah tidak mungkin melalui ketujuh jembatan itu masing-masing satu kali dan kembali ke tempat asal keberangkatan jika derajat tiap simpul tidak seluruhnya genap.

2.1 Graf Berarah dan Tidak Berarah

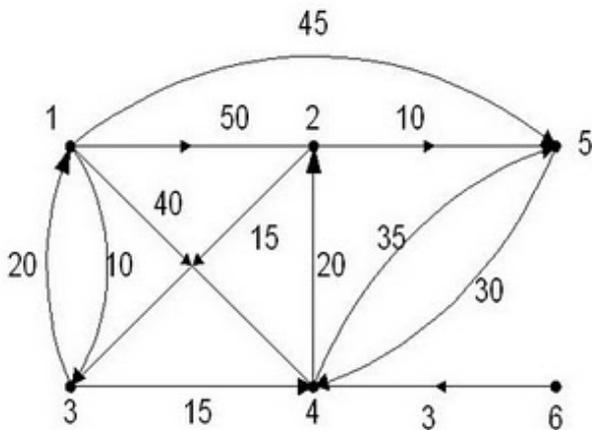
Graf disebut graf berarah jika setiap sisi dari graf memiliki orientasi arah. Pada graf berarah, busur $(v_j, v_k) \neq (v_k, v_j)$. Pada busur (v_j, v_k) , simpul v_j disebut simpul asal (*initial vertex*), simpul v_k disebut simpul terminal (*terminal vertex*).[1]

Graf disebut graf tidak berarah jika setiap sisi dari graf tidak memiliki orientasi arah. Pada graf tak berarah, urutan pasangan simpul yang dihubungkan oleh sisi tidak diperhatikan sehingga $(v_j, v_k) = (v_k, v_j)$. [1]

2.2 Graf Berbobot

Graf berbobot adalah graf yang setiap sisinya diberi sebuah harga (bobot). Nilai dari setiap harga (bobot) tersebut dapat merepresentasikan berbagai macam hal tergantung penggunaan graf tersebut. Misalnya, pada

penggunaan graf untuk representasi jalan, bobot dapat merupakan jarak, lebar jalan, dan lain-lain tergantung kebutuhan pengguna.



Gambar 2. Graf berarah berbobot

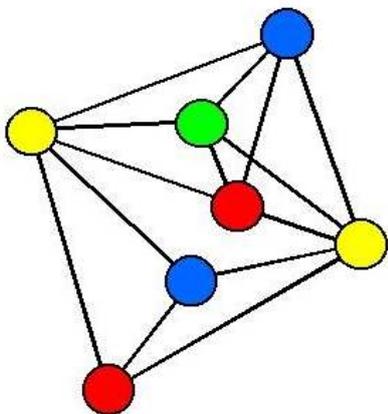
2.3 Pewarnaan Graf (Graph Colouring)

Ada 3 macam pewarnaan graf, yaitu pewarnaan simpul, pewarnaan sisi, dan pewarnaan wilayah (*region*). Pada makalah ini, hanya akan dibahas aplikasi dari pewarnaan simpul, yaitu pada pengaturan jadwal.

Pewarnaan simpul adalah memberi warna pada simpul-simpul suatu graf sedemikian sehingga tidak ada dua simpul bertetangga mempunyai warna yang sama.[1]

Dalam pewarnaan graf, dicari pewarnaan yang menggunakan sesedikit mungkin warna. Jumlah warna minimum yang dapat digunakan untuk mewarnai simpul pada suatu graf disebut bilangan kromatik graf G .

Pemecahan persoalan pewarnaan graf sangat berjasa dalam menentukan jumlah minimum warna yang digunakan untuk mewarnai sembarang peta. Persoalan ini berhasil dipecahkan oleh K. Apple dan W. Haken yang memperlihatkan bahwa empat buah warna sudah cukup untuk mewarnai sembarang graf planar.[1]



Gambar 3. Pewarnaan Graf

3. PENGATURAN JALAN RAYA

3.1 Kondisi Jalan Raya

Kondisi jalan raya di Indonesia masih sangat mengecewakan. Kemacetan kerap kali terjadi di berbagai ruas jalan di kota-kota besar. Selain itu, masih banyak terdapat jalan yang rusak sehingga tentunya menghambat kelancaran arus lalu lintas.

Kemacetan yang kerap kali terjadi tentu saja harus dicari solusinya. Solusi utama untuk memperlancar arus lalu lintas adalah dengan menertibkan para pengguna jalan, terutama angkutan kota yang banyak menjadi biang kemacetan, menambah ruas jalan, memperlebar jalan-jalan, dan memperbaiki jalan yang rusak.

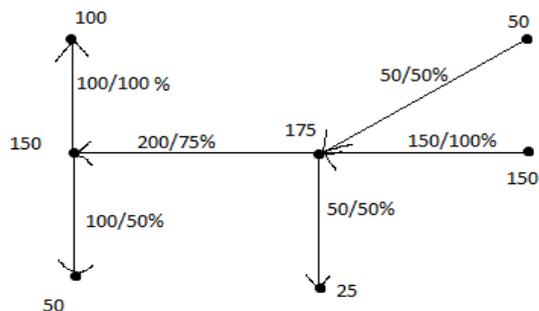
Sementara solusi tersebut belum direalisasikan, untuk menanggulangi kemacetan adalah dengan mengerahkan polisi lalu lintas untuk mengatur arus lalu lintas. Pengaturan arus lalu lintas tersebut dapat diterapkan melalui graf.

3.2 Aplikasi Graf

Pengaturan jalan raya dapat dimodelkan menggunakan graf berbobot yang sedikit dimodifikasi. Modifikasi dilakukan dengan menambah bobot pada tiap simpul dan menambah persentase pada bobot di tiap sisi. Hal ini dilakukan untuk memudahkan polisi lalu-lintas untuk mengatur arus lalu lintas di saat jalan raya sedang padat.

Graf yang digunakan berupa graf berarah yang menunjukkan arah arus kendaraan. Bila jalan raya memiliki 2 arah arus kendaraan, graf digambarkan berdampingan dengan arah yang berbeda.

Bobot yang dituliskan merupakan kapasitas efektif jalan untuk dilalui sejumlah kendaraan setiap satuan waktu. Selain itu, pada pengaplikasiannya di keadaan sebenarnya, perlu ditulis juga persentase dari kondisi arus sebenarnya dibandingkan dengan kapasitas jalan. Bila arus telah melebihi kapasitas jalan efektif, maka persentase menjadi bernilai lebih dari 100%. Sementara itu pada tiap simpul juga dituliskan bobot tertentu, yaitu jumlah kendaraan yang melewati simpul tersebut setiap satuan waktu.



Gambar 4. Penggambaran jalan raya dengan graf berarah

Bila terjadi kepadatan di suatu simpul, polisi diharapkan dapat mengatur arus lalu lintas dengan mengalihkan sebagian kendaraan ke jalur yang belum padat. Tentu saja hal ini harus dilakukan dengan koordinasi dengan anggota polisi lalu-lintas yang berada di simpul lainnya agar tidak terjadi kemacetan di simpul lainnya.

Sistem ini terutama dapat diterapkan secara efektif pada kondisi tertentu, seperti mudik tahunan yang tujuan dari para pengendaranya dapat dicapai dengan melalui jalur utama atau melalui jalur alternatif.

4. PENCARIAN LINTASAN TERPENDEK

4.1 Penggunaan

Pencarian lintasan terpendek banyak digunakan pada sistem navigasi GPS (*Global Positioning System*) dan juga pada jaringan komunikasi.

Pada GPS, lintasan terpendek dicari dan juga dikombinasikan dengan kondisi lalu-lintas pada saat tersebut untuk mencari waktu tercepat untuk sampai pada tujuan yang diinginkan. Keadaan lalu lintas didapatkan melalui hubungan GPS dengan satelit.

Alat GPS kemudian menghitung berapa lama waktu yang dibutuhkan untuk melalui jalur tertentu dan memilih jalur yang memiliki waktu tempuh minimal. Selanjutnya GPS akan memandu pengendara kendaraan untuk melalui lintasan yang dimaksud.

Pada jaringan komunikasi, lintasan terpendek dicari untuk menghemat waktu dan biaya pengiriman. Dengan mencari lintasan terpendek, waktu untuk mengirim pesan agar sampai ke tujuan menjadi lebih singkat dibanding tidak melalui lintasan terpendek. Biaya pengiriman pesan pun menjadi lebih efisien menggunakan lintasan terpendek. Selain itu, dengan penggunaan lintasan terpendek, akan mengurangi beban pada sistem yang bekerja.

4.2 Aplikasi Graf

Pencarian lintasan terpendek telah dilakukan oleh banyak orang. Algoritma untuk mencari lintasan terpendek yang paling terkenal adalah algoritma yang ditemukan oleh Edsger Wybe Dijkstra yang disebut algoritma Dijkstra.

Algoritma Dijkstra mencari lintasan terpendek menggunakan prinsip *greedy*. Prinsip *greedy* pada algoritma Dijkstra menyatakan bahwa pada setiap langkah kita memilih sisi yang berbobot minimum dan memasukkannya ke dalam himpunan solusi.

Berikut ini adalah algoritma Dijkstra dalam bahasa pseudocode [7]:

```
1 function Dijkstra(Graph, source):
2   for each vertex v in Graph: //
    Initializations
```

```
3     dist[v] := infinity //
    Unknown distance function from source to v
4     previous[v] := undefined //
    Previous node in optimal path from source
5     dist[source] := 0 //
    Distance from source to source
6     Q := the set of all nodes in Graph
    // All nodes in the graph are unoptimized
- thus are in Q
7     while Q is not empty: //
    The main loop
8         u := vertex in Q with smallest dist[]
9         if dist[u] = infinity:
10            break //
    all remaining vertices are inaccessible from
    source
11        remove u from Q
12        for each neighbor v of u: //
    where v has not yet been removed from Q.
13            alt := dist[u] + dist_between(u,
    v)
14            if alt < dist[v]: //
    Relax (u,v,a)
15                dist[v] := alt
16                previous[v] := u
17        return dist[]
```

Jika kita hanya ingin mendapatkan jarak terdekat antara simpul asal dan tujuan, kita dapat menghilangkan pencarian pada baris 11 jika $u = \text{target}$ sehingga kita dapat mencari lintasan terdekat menggunakan iterasi:

```
1 S := empty sequence
2 u := target
3 while previous[u] is defined:
4     insert u at the beginning of S
5     u := previous[u]
```

5. PEMODELAN BASIS DATA TERSEBAR

5.1 Definisi Basis Data Tersebar

Sebuah basis data tersebar (*distributed database*) adalah sebuah kumpulan koleksi data (*database/basisdata*) di mana piranti-piranti penyimpanan data (*data storage devices*) tidak dikendalikan oleh CPU (*Central Processing Unit*) yang sama.

Piranti penyimpanan ini bisa saja terletak pada komputer yang berbeda pada lokasi fisik yang sama atau bisa juga terletak di lokasi-lokasi fisik yang berbeda yang saling dihubungkan dengan jaringan komputer (*computer network*). Walaupun secara fisik disimpan di tempat-tempat yang berbeda, secara logik basisdata tersebut adalah satu buah basisdata.

Data-data yang tersebar dalam lokasi-lokasi tersebut dapat merupakan data yang direplikasi (data yang sama diduplikasi di lebih dari 1 lokasi yang berbeda) dan difragmentasi (kelompok data yang sama dipotong-potong dan tiap potongan disimpan di lokasi yang berbeda).

Jika seorang pengguna melakukan *query* ke basisdata tersebut dari lokasi mana pun, maka ia harus mendapatkan semua data secara lengkap tanpa perlu tahu di lokasi mana data tersebut disimpan.

Pada setiap lokasi pada sistem basis data tersebar, terdapat sebuah *client* saja, sebuah *server* saja, atau keduanya.

Pada lokasi yang memiliki *client* saja, operasi yang dapat dilakukan adalah menampilkan data dari *server* komputer yang lainnya dan mengeditnya. *Client* tidak dapat menambah data baru karena tidak memiliki *server*.

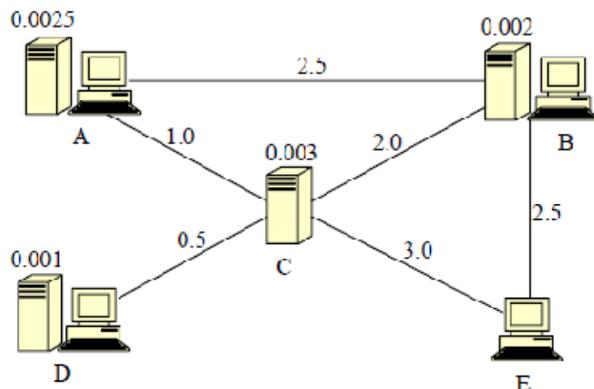
Pada lokasi yang memiliki *server* saja, penambahan data dapat dilakukan, tetapi tidak dapat mengubah data-data yang terdapat pada *server*.

5.2 Aplikasi Graf

Pemodelan basis data tersebar juga dilakukan dengan menggunakan graf berbobot yang dimodifikasi dengan menambahkan bobot pada tiap simpulnya seperti pada aplikasi pengaturan jalan raya.

Bobot yang dituliskan pada tiap sisi merupakan laju data transmisi (*bandwidth*) yang menyatakan jumlah maksimum data yang bisa dilewatkan dalam jaringan (baik untuk proses pengambilan/*download* data maupun pengiriman/*upload* data, dalam satuan tertentu (satuan ukuran data/satuan waktu)).

Bobot yang dituliskan pada tiap node merupakan waktu proses (*CPU time*) yang menyatakan berapa waktu rata-rata yang dibutuhkan untuk menyelesaikan satu buah operasi yang melibatkan suatu *record* data. Pada node yang tidak memiliki server tidak ada bobot yang ditulis.



Gambar 5. Graf yang menggambarkan jaringan basis data tersebar.

Pada gambar terlihat bahwa pada lokasi C hanya terdapat *server*, sedangkan pada lokasi E, hanya terdapat *client*. Pada lokasi lainnya terdapat *client* dan *server*.

Nilai yang terdapat pada setiap sisi dan titik digunakan untuk menghitung berapa lama waktu yang dibutuhkan oleh sebuah *client* pada lokasi tertentu untuk mengambil data pada lokasi yang lainnya dan kemudian memrosesnya untuk digunakan pada lokasi tersebut.

Nilai tersebut didapat dengan mencari lintasan terpendeknya dan menghitung waktu yang dibutuhkan agar *record* sampai ke tujuan untuk diproses.

Pertama-tama, dihitung lama proses *download* data dengan membagi ukuran data yang ditransfer dengan kecepatan *download* ke sever tujuan pertama pada lintasan terpendeknya. Selanjutnya dihitung demikian sampai mencapai lokasi yang diinginkan. Setelah tiba di lokasi tersebut, data diolah. Lama pengolahan data bergantung pada nilai *CPU time*. Waktu total yang diperlukan merupakan jumlah dari seluruh proses tersebut, yaitu lama proses *download* dan pengolahan *record*.

6. EKOSISTEM

6.1 Definisi Ekosistem

Ekosistem adalah suatu sistem ekologi yang terbentuk oleh hubungan timbal balik antara makhluk hidup dengan lingkungannya. Ekosistem bisa dikatakan juga suatu tatanan kesatuan secara utuh dan menyeluruh antara segenap unsur lingkungan hidup yang saling mempengaruhi.

Para ahli ekologi mempelajari hal berikut:

1. Perpindahan energi dan materi dari makhluk hidup yang satu ke makhluk hidup yang lain ke dalam lingkungannya dan faktor-faktor yang menyebabkannya.
2. Perubahan populasi atau spesies pada waktu yang berbeda dalam faktor-faktor yang menyebabkannya.
3. Terjadi hubungan antarspesies (interaksi antarspesies) makhluk hidup dan hubungan antara makhluk hidup dengan lingkungannya.

Perpindahan energi melalui proses makan dan dimakan serta juga energi dari cahaya matahari dalam ekosistem dapat digambarkan dengan rantai makanan.

Rantai makanan adalah perpindahan energi makanan dari sumber daya tumbuhan melalui seri organisme atau melalui jenjang makan (tumbuhan-herbivora-carnivora-omnivora). Pada setiap tahap pemindahan energi, 80%–90% energi potensial hilang sebagai panas, karena itu langkah-langkah dalam rantai makanan terbatas 4-5 langkah saja. Dengan perkataan lain, semakin pendek rantai makanan semakin besar pula energi yang tersedia.

Berdasarkan[5], ada dua tipe dasar rantai makanan:

1. Rantai makanan rerumputan (*grazing food chain*). Misalnya: tumbuhan-herbivora-carnivora-omnivora.
2. Rantai makanan sisa (*detritus food chain*). Bahan mati mikroorganisme (detrivora = organisme pemakan sisa) predator dan bangkai.

Berdasarkan [6], para ilmuwan ekologi mengenal tiga macam rantai pokok, yaitu rantai pemangsa, rantai parasit, dan rantai saprofit.

1. Rantai Pemangsa

Rantai pemangsa landasan utamanya adalah tumbuhan hijau sebagai produsen. Rantai pemangsa dimulai dari hewan yang bersifat herbivora sebagai konsumen I, dilanjutkan dengan hewan karnivora yang memangsa herbivora sebagai konsumen ke-2 dan berakhir pada hewan pemangsa karnivora maupun herbivora sebagai konsumen ke-3.

2. Rantai Parasit

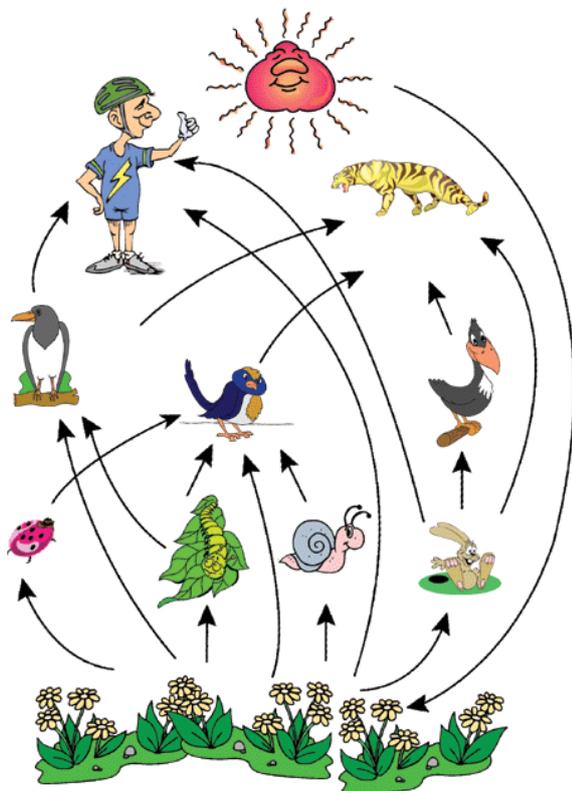
Rantai parasit dimulai dari organisme besar hingga organisme yang hidup sebagai parasit. Contoh organisme parasit antara lain cacing, bakteri, dan benalu.

3. Rantai Saprofit

Rantai saprofit dimulai dari organisme mati ke jasad pengurai. Misalnya jamur dan bakteri. Rantai-rantai di atas tidak berdiri sendiri tapi saling berkaitan satu dengan lainnya sehingga membentuk jaring-jaring makanan.

6.2 Aplikasi Graf

Aplikasi graf dalam ekosistem digunakan untuk penggambaran rantai makanan. Graf yang digunakan adalah graf berarah. Simpul awal merupakan makhluk hidup yang dimangsa, sedangkan simpul tujuan merupakan makhluk hidup pemangsa.



Gambar 6. Penggambaran rantai makanan menggunakan graf

7. PEMBUATAN JADWAL

7.1 Penggunaan Jadwal dengan Graf

Pembuatan jadwal dapat digunakan dalam kehidupan sehari-hari orang banyak. Namun, untuk pembuatan jadwal yang melibatkan banyak orang diperlukan suatu metode agar tidak terjadi 'bentrok' jadwal, yaitu satu pihak harus melakukan dua kegiatan yang dilakukan pada waktu yang sama.

Hal ini dapat diterapkan dalam penyusunan jadwal ujian, jadwal pertandingan suatu perlombaan atau kejuaraan, dan lain-lain.

7.2 Aplikasi Graf

Pada pembuatan jadwal ujian menggunakan graf, tiap simpul merupakan hari ujian mata kuliah tertentu dilaksanakan, dengan sisi yang menghubungkannya menggambarkan mahasiswa yang mengambil mata kuliah tersebut.

Sebagai contoh, bila simpul A terhubung dengan simpul B dan C berarti ada mahasiswa yang mengambil mata kuliah A dan B dan ada mahasiswa lainnya yang mengambil mata kuliah A dan C, atau bahkan ada juga mahasiswa yang mengambil mata kuliah A, B, dan C bila B dan C juga terhubung.

Untuk itu dilakukan pewarnaan graf sehingga simpul yang bertetanggan tidak memiliki warna yang sama agar tidak ada mahasiswa yang harus mengikuti 2 ujian pada waktu yang sama.

Begitu juga pada pembuatan jadwal pertandingan suatu kejuaraan. Simpul diibaratkan sebagai pertandingan tertentu dengan jadwal tertentu, sedangkan sisi merepresentasikan orang atau tim yang harus bertanding pada pertandingan tersebut.

Pewarnaan dilakukan agar tidak ada tim atau orang yang harus bertanding pada dua pertandingan pada waktu yang sama.

Selain menggunakan pewarnaan graf, sekarang ini telah ada program untuk menyusun jadwal menggunakan algoritma genetic termodifikasi.



Gambar 7. Program penyusunan jadwal

8. KESIMPULAN

Kesimpulan yang didapat dari pembahasan mengenai aplikasi graf adalah graf memiliki berbagai macam penerapan yang dapat didesain sendiri sesuai dengan kebutuhan pengguna untuk memecahkan permasalahan yang ada atau juga untuk memodelkan suatu system yang komponennya memiliki hubungan dengan komponen lainnya.

Saya ingin mengucapkan banyak terima kasih kepada Bapak Rinaldi dan Ibu Harlili selaku dosen pengajar mata kuliah struktur diskrit. Selain itu, saya juga mengucapkan terima kasih kepada teman-teman saya yang ikut memberikan masukan dan dorongan atas berlangsungnya pembuatan makalah ini.

REFERENSI

- [1] Munir, Rinaldi, Diktat Kuliah IF 2091 Struktur Diskrit Edisi Keempat. Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung. Hal. VIII-1 - VIII-55.
- [2] Liem, Inggriani, Draft Diktat Struktur Data, Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung.
- [3] http://en.wikipedia.org/wiki/Graph_theory. Waktu akses 11:09 19/12/2009
- [4] <http://id.wikipedia.org/wiki/Ekosistem>. Waktu akses 11:11 19/12/2009
- [5] http://id.wikipedia.org/wiki/Rantai_makanan. Waktu akses 11:33 19/12/2009
- [6] <http://kasatindra.wordpress.com/2008/08/11/rantai-makanan>. Waktu akses 11:52 19/12/2009
- [7] http://en.wikipedia.org/wiki/Dijkstra%27s_algorithm. Waktu akses 13:00 19/12/2009
- [8] <http://sutanto.staff.uns.ac.id/2008/09/07/algorithm-genetic-termodifikasi/>. Waktu akses 13:13 19/12/2009