



**PEMROGRAMAN PENJADUALAN SEMESTER PENDEK
DI JURUSAN MATEMATIKA DENGAN ALGORITMA
PEWARNAAN TITIK PADA GRAF**

SKRIPSI

Diajukan untuk Memenuhi Persyaratan Penyelesaian Program Sarjana Sains
Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Jember



Oleh :

Jati Aninka

NIM. 971810101033

**JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER**

2003

Motto

“Sesungguhnya sesudah kesulitan itu ada kemudahan, maka apabila kamu telah selesai dari suatu urusan, kerjakanlah dengan sungguh-sungguh urusan yang lain. Dan hanya kepada Allahlah hendaknya kamu berharap”

{ QS Al-Insyirah: 6-7 }

“Kesuksesan dan keberhasilan tidak akan dapat tercapai tanpa adanya niat, usaha dan do'a”

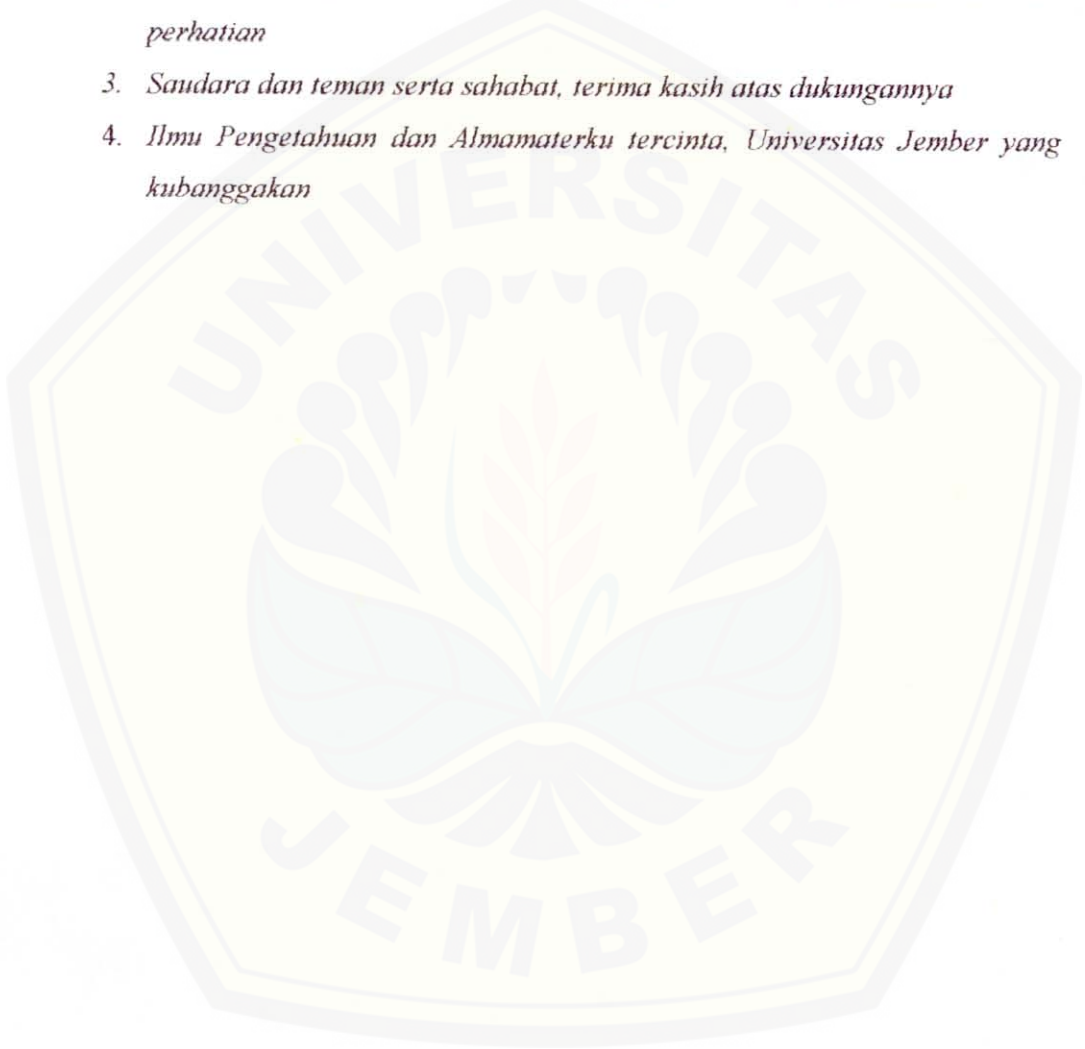
{N.N.}



Persembahkan

Aku persembahkan karya ini pada :

- 1. Yang kuhormati, Ayahanda Bambang Yudono dan Ibunda Rahayu Prihani yang memberi dorongan dan Do'a restu*
- 2. Yang kusayangi Kakak-kakakku dan juga Adikku yang selalu memberi perhatian*
- 3. Saudara dan teman serta sahabat, terima kasih atas dukungannya*
- 4. Ilmu Pengetahuan dan Almamaterku tercinta, Universitas Jember yang kubanggakan*



DEKLARASI

Skripsi ini berisi hasil kerja/penelitian mulai bulan Oktober 2002 sampai dengan bulan Juni 2003. Bersama ini saya menyatakan bahwa isi skripsi ini adalah hasil pekerjaan saya sendiri kecuali jika disebutkan sumbernya dan skripsi ini belum pernah diajukan pada institusi lain.

Jember, Juni 2003

Jati Aninka



ABSTRAK

“Pemrograman Penjadualan Semester Pendek di Jurusan Matematika dengan Algoritma Pewarnaan Titik pada Graf”, Jati Aninka, NIM. 971810101033, Skripsi, Juni 2003, Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Jember.

Dalam penyusunan jadwal Semester Pendek di Jurusan Matematika perlu diperhatikan elemen-elemen penjadualan yang ada antara lain jumlah mata kuliah dan SKS-nya, jumlah dosen dan mata kuliah yang diajarkan, serta jumlah mahasiswa yang mengikuti Mata kuliah yang ditawarkan. Dengan menganggap bahwa mata kuliah sebagai titik dalam graf, maka jika ada mahasiswa yang menempuh lebih dari dua mata kuliah maka mata kuliah tersebut tidak boleh dilaksanakan secara bersamaan. Cara menyelesaikan adalah dengan pewarnaan titik pada mata kuliah. Algoritma pewarnaan yang digunakan mempertimbangkan derajat titik dan keterhubungan titik. Jika titik mata kuliah terhubung dengan titik mata kuliah yang lain maka mata kuliah tersebut diwarnai berbeda yang berarti kedua mata kuliah tersebut dilaksanakan dalam periode waktu yang berbeda. Analisa pemrograman dengan bahasa komputer, adalah dengan mengubah masukan data menjadi matriks. Dengan menggunakan matriks, perhitungan derajat titik adalah menjumlahkan elemen baris dari matriks adjacent mata kuliah, dilanjutkan dengan pengurutan elemen baris berdasarkan derajat. Pewarnaan titik mata kuliah dalam matriks menggunakan algoritma pewarnaan penjadualan, lalu dibuat matriks jadwal 1 hari yang elemen barisnya berisi warna yang berbeda. Proses selanjutnya modifikasi matriks jadwal 1 hari menjadi jadwal keseluruhan dengan mempertimbangkan jumlah SKS mata kuliah. Penyusunan jadwal dengan komputerisasi dapat lebih cepat dan mudah dilakukan hanya dengan memasukkan data-data yang akan dijadualkan.

Kata kunci : Graf, Pewarnaan Titik, Algoritma Pewarnaan Penjadualan, Pemrograman Turbo Pascal for Windows.

PENGESAHAN

Skripsi ini diterima oleh Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember pada :

Hari : KAMIS

Tanggal : 26 JUN 2003

Tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Jember

Tim Penguji,

Ketua (DPU)



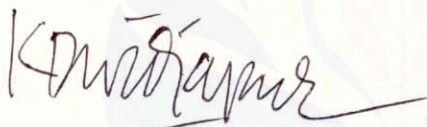
(Kristiana Wijaya, M.Si.)
NIP. 132 258 180

Sekretaris (DPA)



(Kiswara Agung Santoso, S.Si.)
NIP. 132 207 813

Anggota 1,



(Kosala Dwidja Purnomo, S.Si.)
NIP. 132 206 019

Anggota 2,



(M. Fatekurohman, M.Si.)
NIP. 132 210 538

Mengesahkan,

Dekan Fakultas MIPA

Universitas Jember



(Dr. Sumadi, MS)
NIP. 130 368 784

KATA PENGANTAR

Bismillahirrohmanirrohim.

Alhamdulillahirobil'alamin.

Puji syukur penulis panjatkan kehadiran Allah SWT atas rahmat, barokah, hidayah, dan inayah-Nya, sehingga skripsi ini dapat penulis selesaikan. Penulisan skripsi ini dimaksudkan untuk memenuhi salah satu syarat guna memperoleh gelar kesarjanaan Jurusan Matematika pada Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Dalam penulisan skripsi ini, penulis telah banyak mendapatkan bantuan dan dorongan baik secara langsung maupun tidak langsung dari berbagai pihak. Untuk itu penulis menyampaikan terima kasih yang sedalam-dalamnya kepada:

1. Kristiana Wijaya, M.Si., selaku Dosen Pembimbing Utama dan Kiswara Agung Santoso, S.Si, selaku Dosen Pembimbing Anggota yang telah memberikan bimbingannya kepada penulis sehingga karya tulis ilmiah ini dapat terselesaikan dengan baik.
2. Kosala Dwidja Purnomo, S.Si dan M. Fatekurohman, M.Si, selaku dosen penguji yang telah memberikan kritik, saran dan masukan sehingga karya tulis ilmiah ini dapat terselesaikan dengan baik.
3. Drs. Kusno, DEA, PhD, selaku Ketua Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.
4. Ir. Sumadi MS, selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.
5. Seluruh Dosen dan Civitas Akademika di lingkungan Fakultas Matematika dan Ilmu Pengetahuan Alam, khususnya Jurusan Matematika.
6. Kedua Orangtuaku, Kakak-kakak dan Adikku serta seluruh keluargaku yang telah banyak memberikan dorongan moril dan materiil kepada penulis.
7. Rian's, terima kasih atas dorongan semangat, do'a dan kesabarannya.

8. Teman-temanku : Athfal, Hadi, Taufik, Rina, Adi S, Wirid, Isnaini, Bahtiar, Vita, Agung dan lainnya, terima kasih dukungan semangatnya.
9. Teman-teman angkatan '97 dan adik angkatan di Jurusan Matematika, serta semua pihak yang telah membantu terselesaikannya skripsi ini.

Semoga Allah SWT memberikan anugerah yang setimpal.

Mengingat keterbatasan waktu, biaya, serta pengetahuan yang ada, maka penulis menyadari adanya kekurangan dan kelemahan pada penulisan karya ilmiah ini. Karena itu kritik dan saran yang konstruktif sangat penulis harapkan demi kesempurnaan skripsi ini.

Selanjutnya penulis berharap semoga skripsi ini dapat memberi kontribusi terhadap kemajuan ilmu pengetahuan khususnya bidang ilmu Matematika Komputasi dan memberikan manfaat.

Jember, Juni 2003

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN MOTTO	ii
HALAMAN PERSEMBAHAN	iii
HALAMAN DEKLARASI	iv
ABSTRAK	v
HALAMAN PENGESAHAN	vi
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN	xiv
BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Perumusan Masalah	1
1.3 Batasan Masalah	2
1.4 Tujuan	2
1.5 Manfaat	2
BAB II TINJAUAN PUSTAKA	
2.1 Pengertian Graf	3
2.2 Konsep Dasar	3
2.3 Penyajian Graf dalam Komputer	7
2.4 Pewarnaan Graf	8
2.4.1 Pewarnaan Titik	8
2.4.2 Pewarnaan Sisi	9
2.4.3 Algoritma Pewarnaan Titik	10
A. Algoritma Pewarnaan Berurutan	10
B. Algoritma Welsh-Powel	12

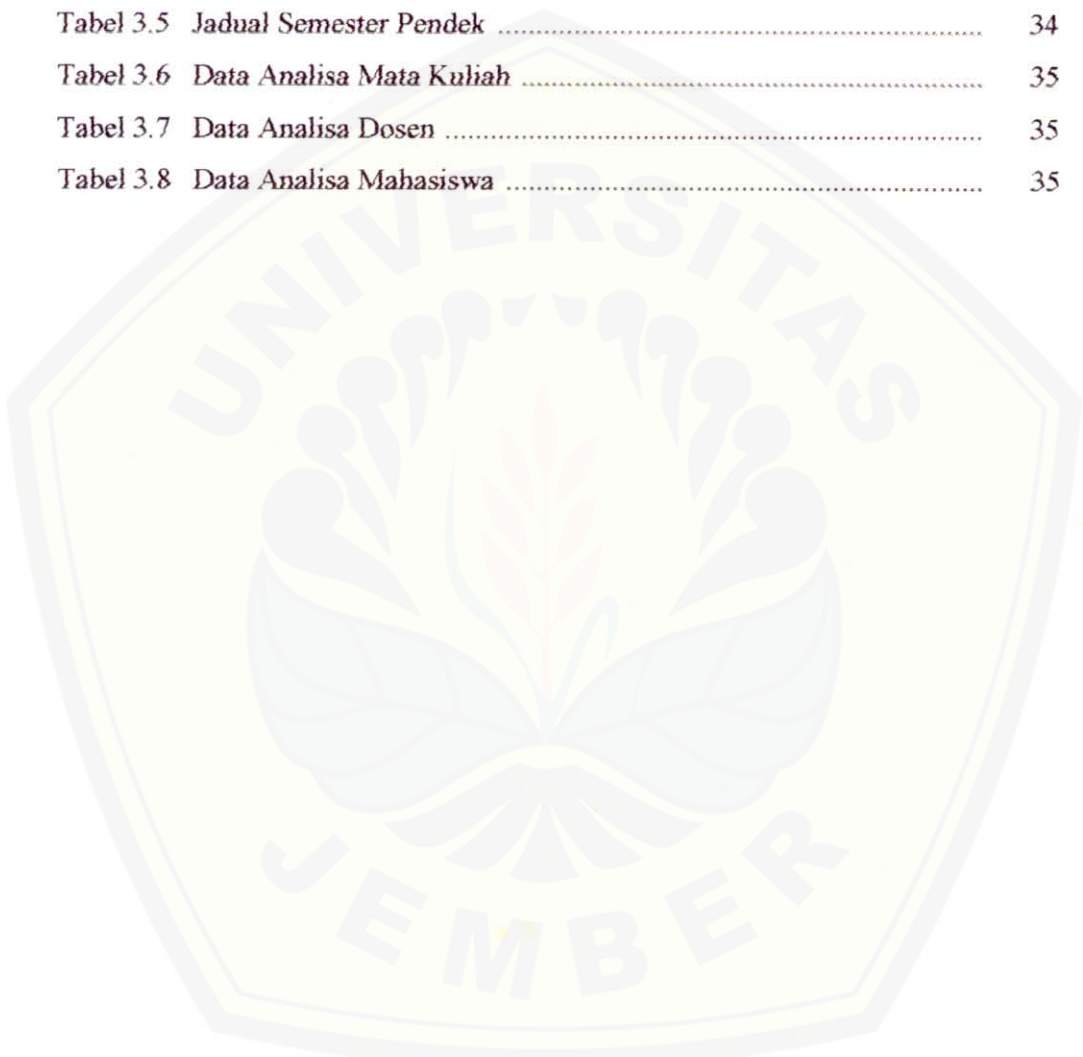
2.5	Penjadualan	13
2.6	Semester Pendek	13
BAB III HASIL DAN PEMBAHASAN		
3.1	Algoritma Pewarnaan Penjadualan	15
3.2	Elemen-Elemen Penjadualan	18
3.3	Prosedur Penjadualan	20
3.4	Pengolahan Data dan Penyusunan Jadwal	21
3.5	Algoritma Program Penjadualan	26
3.6	Ilustrasi Penjadualan	27
3.7	Analisa Pemrograman Penjadualan	35
BAB IV KESIMPULAN DAN SARAN		
4.1	Kesimpulan	43
4.2	Saran	43

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR TABEL

Tabel 3.1	Periode Penjadualan Semester Pendek di Jurusan Matematika	19
Tabel 3.2	Data Mata Kuliah Dosen dan Mahasiswa	21
Tabel 3.3	Jadual Perkuliahan	25
Tabel 3.4	Jadual Hasil Pewarnaan Penjadualan	34
Tabel 3.5	Jadual Semester Pendek	34
Tabel 3.6	Data Analisa Mata Kuliah	35
Tabel 3.7	Data Analisa Dosen	35
Tabel 3.8	Data Analisa Mahasiswa	35



DAFTAR GAMBAR

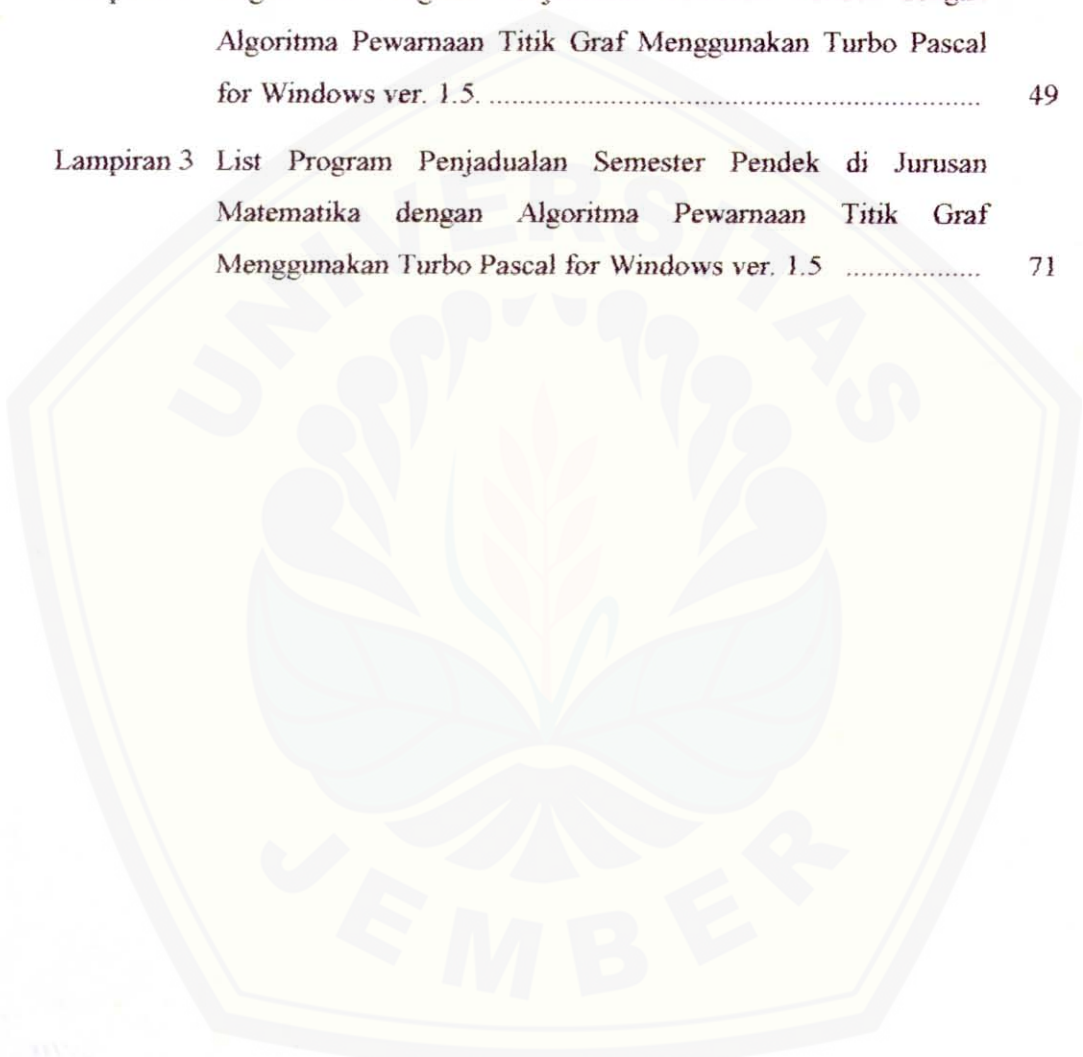
Gambar 2.1	Graf dengan 5 Titik dan 4 Sisi	3
Gambar 2.2	Graf dengan Gelung dan Sisi Rangkap	4
Gambar 2.3	Graf Ilustrasi <i>Adjacent</i> dan <i>Incident</i>	4
Gambar 2.4	Ilustrasi Graf Bagian dan Graf Perentang	5
Gambar 2.5	Graf Lengkap (K_5)	6
Gambar 2.6	Graf Bipartit	6
Gambar 2.7	Graf dan Komplemennya	6
Gambar 2.8	Graf G untuk Ilustrasi Matriks <i>Adjacent</i> dan Matriks <i>Incident</i>	7
Gambar 2.9	Pewarnaan Titik Graf	8
Gambar 2.10	Pewarnaan Sisi Graf	9
Gambar 2.11	Pewarnaan Titik Graf dengan Algoritma Pewarnaan Berurutan	12
Gambar 2.12	Pewarnaan Titik Graf dengan Algoritma Pewarnaan <i>Welsh</i> <i>Powel</i>	13
Gambar 3.1	Graf Ilustrasi Pewarnaan	16
Gambar 3.2	Pelabelan Titik Graf dengan Aturan Derajat Titik	16
Gambar 3.3	Graf Hasil Pewarnaan dengan Algoritma Pewarnaan	18
Gambar 3.4	Graf Bipartit Mata Kuliah-Dosen dan Mata Kuliah- Mahasiswa.....	22
Gambar 3.5	Graf Bipartit Mata Kuliah-Dosen-Mahasiswa	23
Gambar 3.6	Graf Mata Kuliah	24
Gambar 3.7	Pewarnaan Graf Mata Kuliah	25
Gambar 3.8	Graf Data Mata Kuliah – Dosen	28
Gambar 3.9	Graf Data Mata Kuliah – Mahasiswa	32
Gambar 3.10	Graf Pewarnaan	33
Gambar 3.11	Graf Hasil Pewarnaan	33
Gambar 3.12	Graf Analisa Mata Kuliah dan Dosen	36
Gambar 3.13	Graf Analisa Mata Kuliah dan Mahasiswa	37

Gambar 3.14 Graf Adjacent Mata Kuliah	39
Gambar 3.15 Graf Pelabelan Titik	40
Gambar 3.16 Graf Hasil Pewarnaan	41



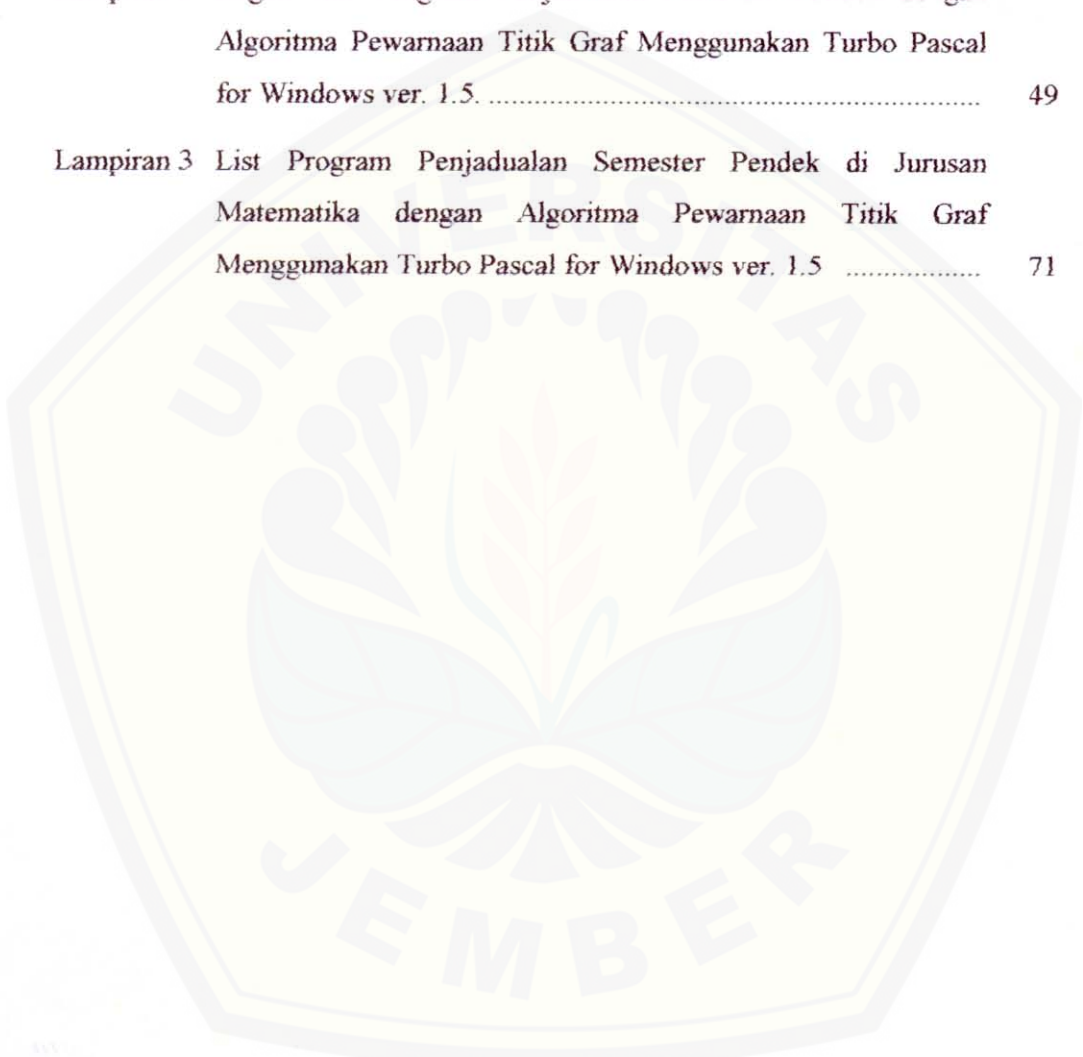
DAFTAR LAMPIRAN

Lampiran 1	Data Ilustrasi Mata Kuliah, Dosen dan Mahasiswa dari Pelaksanaan Semester Pendek di Jurusan Matematika Tahun 2002	45
Lampiran 2	Bagan Alir Program Penjadualan Semester Pendek dengan Algoritma Pewarnaan Titik Graf Menggunakan Turbo Pascal for Windows ver. 1.5.	49
Lampiran 3	List Program Penjadualan Semester Pendek di Jurusan Matematika dengan Algoritma Pewarnaan Titik Graf Menggunakan Turbo Pascal for Windows ver. 1.5	71



DAFTAR LAMPIRAN

Lampiran 1	Data Ilustrasi Mata Kuliah, Dosen dan Mahasiswa dari Pelaksanaan Semester Pendek di Jurusan Matematika Tahun 2002	45
Lampiran 2	Bagan Alir Program Penjadualan Semester Pendek dengan Algoritma Pewarnaan Titik Graf Menggunakan Turbo Pascal for Windows ver. 1.5.	49
Lampiran 3	List Program Penjadualan Semester Pendek di Jurusan Matematika dengan Algoritma Pewarnaan Titik Graf Menggunakan Turbo Pascal for Windows ver. 1.5	71



BAB I
PENDAHULUAN**1.1 Latar Belakang**

Setiap individu memiliki aktifitas yang berlainan dan tidak semua individu tersebut harus melakukan aktifitas dengan berada di dalam ruangan. Seiring dengan aktifitas manusia yang serba kompleks dan memerlukan waktu yang tepat, maka banyak kegiatan yang harus diatur sehingga seluruh aktivitas tersebut dapat diselesaikan dengan efisien. Pengaturan antara aktivitas kegiatan dengan waktu disebut juga sebagai penjadualan.

Penjadualan didefinisikan sebagai proses, pembuatan, cara menjadualkan atau memasukkan dalam jadwal, sedangkan jadwal itu sendiri dapat didefinisikan sebagai pembagian waktu berdasarkan rencana pengaturan urutan kerja atau daftar atau tabel kegiatan atau rencana kegiatan dengan pembagian waktu pelaksanaan yang terinci.

Penjadualan yang dimaksud dalam skripsi ini dapat didefinisikan sebagai suatu pencatatan atau penskemaan dari waktu secara sistematis sebagai daftar/rencana waktu pelajaran atau pekerjaan yang harus dijalankan. Dengan dibuatnya jadwal, maka setiap elemen yang dijadualkan mendapat kesesuaian dengan elemen yang lain sehingga tidak ada elemen-elemen dari jadwal yang beririsan. Elemen dari penjadualan antara lain adalah jumlah peserta, kegiatan peserta, jumlah ruangan, penyaji, periode dan alokasi waktu.

Salah satu cara untuk menyusun sebuah jadwal adalah digunakannya teori graf, yaitu pewarnaan graf. *Pewarnaan graf* merupakan suatu pemberian warna pada salah satu elemen-elemen (titik, sisi atau bidang) dari graf dengan menggunakan aturan tertentu. Dalam skripsi ini pewarnaan yang digunakan adalah pewarnaan titik.

Penjadualan dengan pewarnaan titik dapat dikaitkan, misalnya elemen penjadualan seperti peserta dan kegiatan diatur sedemikian hingga jika ada peserta yang melakukan dua atau lebih kegiatan maka elemen tersebut tidak boleh diwarnai dengan warna yang sama.

Dalam era teknologi saat ini pantas apabila dalam penyusunan jadwal digunakan metode komputerisasi, sehingga penyusunan jadwal dapat lebih cepat, efisien dan mudah dilakukan dengan memperhitungkan batasan beberapa elemen-elemen yang ada.

1.2 Perumusan Masalah

Masalah yang dibahas adalah menentukan suatu algoritma untuk merencanakan/menyusun sebuah jadwal dengan memperhatikan elemen-elemen yang ada seperti lain jumlah peserta, jumlah ruangan, jumlah penyaji, jumlah periode dan alokasi waktu, berdasarkan pewarnaan titik.

1.3 Batasan Masalah

Dalam skripsi ini pewarnaan graf yang digunakan adalah pewarnaan titik. Batasan permasalahan yang diajukan adalah perencanaan penjadualan pada Semester Pendek di Jurusan Matematika dengan semua persyaratannya yang disajikan dengan menggunakan bahasa pemrograman menggunakan Turbo Pascal for Windows ver 1.5.

1.4 Tujuan

Penulisan skripsi ini bertujuan untuk:

1. mendapatkan algoritma pewarnaan titik untuk perencanaan/penyusunan penjadualan Semester Pendek di Jurusan Matematika,
2. mengaplikasikan penyajian penjadualan menggunakan bahasa pemrograman.

1.5 Manfaat

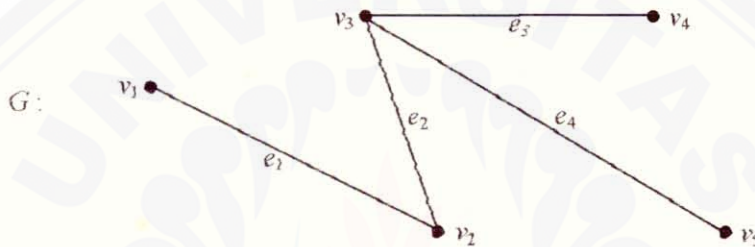
Pewarnaan graf dalam hal ini adalah pewarnaan titik dapat bermanfaat untuk merencanakan jadwal yang sesuai dengan keadaan yang ada, sehingga dalam penyusunan jadwal beberapa kesalahan yang sering terjadi seperti jadwal yang bersamaan, pengajar yang berbenturan dengan kegiatan lain, atau ruangan yang tidak ada, dapat dihindarkan atau paling tidak diminimalisir. Aplikasi program yang dibuat digunakan untuk mempermudah penyaji, dosen, lembaga pendidikan, atau lain-lain untuk mendapatkan jadwal yang sesuai.

BAB II

TINJAUAN PUSTAKA

2.1 Pengertian Graf

Sebuah graf G didefinisikan sebagai pasangan himpunan (V, E) , dimana $V(G) = \{v_1, v_2, v_3, \dots, v_n\}$ adalah sebuah himpunan hingga tak kosong yang elemen-elemennya disebut titik dan $E(G) = \{e_1, e_2, e_3, \dots, e_m\}$ adalah himpunan hingga boleh kosong yang elemen-elemennya menunjukkan pasangan tak terurut dari dua elemen di V yang disebut sisi, yaitu $e = (u, v)$ dengan $u, v \in V$ [Fletcher:418]. Sebagai contoh, Gambar 2.1 merupakan graf dengan 5 titik dan 4 sisi, $V = \{v_1, v_2, v_3, v_4, v_5\}$ dan $E = \{e_1, e_2, e_3, e_4\}$.



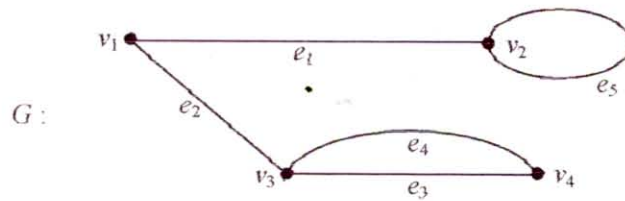
Gambar 2.1 Graf dengan 5 Titik dan 4 Sisi

2.2 Konsep Dasar

Berikut diberikan definisi dan notasi dalam teori graf yang berkaitan dengan skripsi ini.

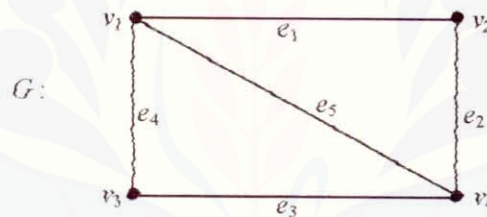
Order dari graf G adalah banyaknya titik yang ada di G yaitu $n = |V|$. Graf yang mempunyai order hingga dinamakan *graf hingga*. Sebagai contoh, graf G pada Gambar 2.1 adalah graf berorder 5.

Sebuah sisi dalam graf G yang menghubungkan sebuah titik dengan dirinya sendiri disebut *gelung (loop)*. Jika terdapat 2 atau lebih sisi yang menghubungkan dua titik, maka sisi-sisi tersebut disebut *sisi rangkap (multiple edges)*. Contoh graf dengan gelung dan sisi rangkap ditunjukkan pada Gambar 2.2. Sebuah graf yang tidak memiliki gelung dan sisi rangkap disebut *graf sederhana*. Dalam skripsi ini graf yang digunakan adalah graf hingga dan sederhana.



Gambar 2.2 Graf dengan Gelung dan Sisi Rangkap

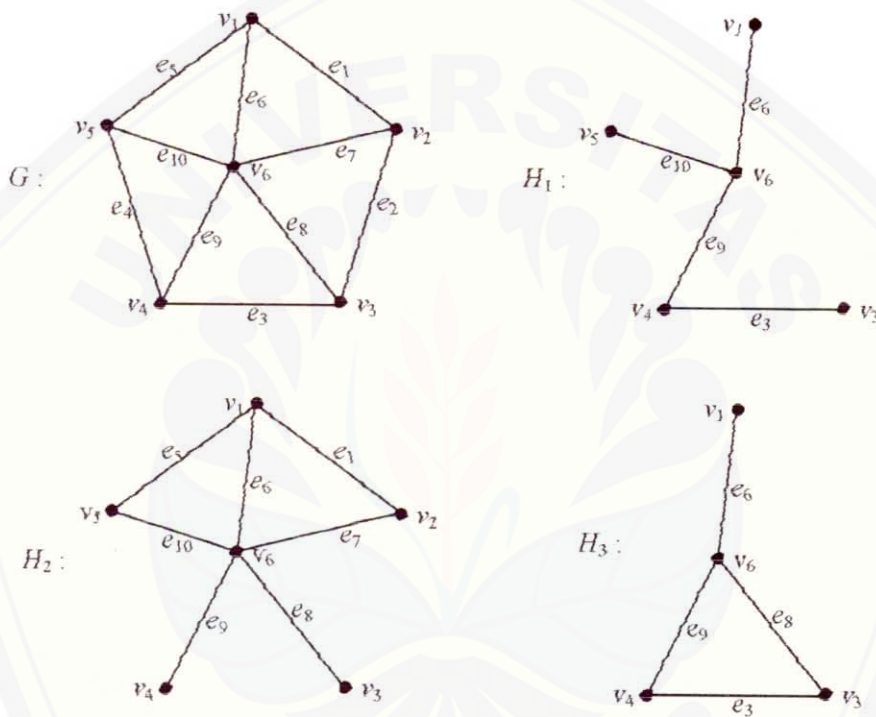
Misal u dan v adalah titik-titik di G dan $e = (u,v)$ adalah sisi di G , maka titik u dan v tersebut dikatakan bertetangga (*adjacent*), sedangkan sisi e dikatakan terkait (*incident*) dengan titik u atau v . Banyaknya sisi yang terkait dengan titik v di G disebut *derajat (degree) titik v* dan dinotasikan dengan $deg(v)$. Titik v di G yang mempunyai derajat terbesar/maksimum dinotasikan dengan $\Delta(G)$. Jika setiap titik dalam G mempunyai derajat titik yang sama, maka graf tersebut dinamakan *graf regular*. Pada Gambar 2.3, titik v_1 bertetangga dengan titik v_2, v_3 dan v_4 , sedangkan sisi e_5 terkait dengan titik v_1 dan v_4 ; $deg(v_2) = deg(v_3) = 2$ dan $deg(v_1) = deg(v_4) = 3$ dan juga $\Delta(G)=3$ pada v_1 dan v_4 .



Gambar 2.3 Graf Ilustrasi Adjacent dan Incident

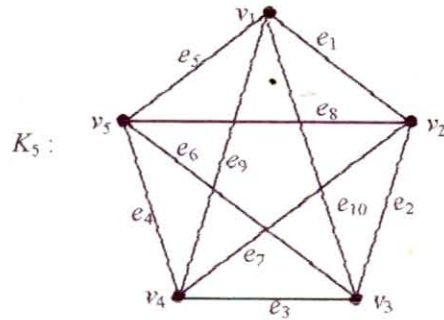
Sebuah *jalan (walk) W* dari v_0 ke v_n pada graf G adalah barisan berhingga (tak kosong) $W = v_0, e_1, v_1, e_2, v_2, e_3, \dots, v_n, e_n$ yang suku-sukunya bergantian antara titik dan sisi, sedemikian hingga v_{i-1} dan v_i adalah titik-titik akhir sisi e_i untuk $1 \leq i \leq n$ [1 Ketut: 75]. Jalan dikatakan tertutup jika $v_0 = v_n$, dan terbuka jika $v_0 \neq v_n$. Suatu jalan yang barisan titik-titiknya tidak ada pengulangan dinamakan *lintasan (path)*, sedangkan jika yang berbeda adalah sisi-sisinya, maka jalan tersebut disebut *jejak*. *Sikel (cycle)* didefinisikan sebagai jalan tertutup dengan barisan titik yang berbeda, dengan kata lain sikel adalah suatu lintasan yang tertutup.

Sebuah graf H disebut *graf bagian* (*sub graph*) dari graf G , ditulis $H \subseteq G$, jika $V(H) \subseteq V(G)$ dan $E(H) \subseteq E(G)$. Jika $H \subseteq G$ dan $V(H) = V(G)$, maka H disebut *graf bagian perentang* (*spanning sub graph*) dari G . Misalkan $V_1 \subseteq V(G)$, graf bagian dari G yang dibangun oleh V_1 , dilambangkan dengan $G[V_1]$, adalah sebuah graf bagian dari G yang himpunan titiknya adalah V_1 dan himpunan sisinya beranggotakan sisi G . Pada Gambar 2.4 graf H_1 adalah graf bagian (bukan perentang) dari graf G , graf H_2 adalah graf bagian perentang dari graf G , dan H_3 adalah graf bagian dari G yang dibangun oleh $V_1 = \{ v_1, v_3, v_4, v_6 \}$.

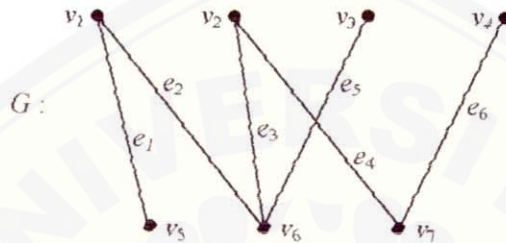


Gambar 2.4 Ilustrasi Graf Bagian dan Graf Perentang

Sebuah *graf lengkap* dengan n titik, dilambangkan dengan K_n , adalah graf sederhana dimana untuk setiap dua titik dalam graf tersebut bertetangga. Sebuah graf G disebut *graf bipartit* jika himpunan titik $V(G)$ dapat dipartisi menjadi dua himpunan bagian A dan B sedemikian hingga setiap sisi dari G menghubungkan titik di A dan titik di B [I Ketut: 72]. Graf lengkap dan graf bipartit ditunjukkan pada Gambar 2.6 sebagai berikut:

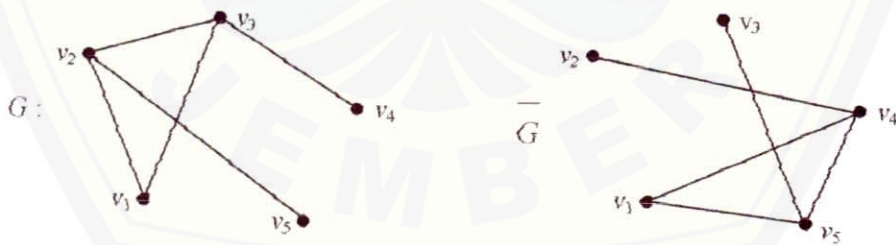


Gambar 2.5 Graf Lengkap (K_5)



Gambar 2.6 Graf Bipartit

Sebuah graf G dikatakan *terhubung* jika setiap dua titik u, v di G , terdapat lintasan di G yang menghubungkan ke dua titik tersebut. *Komplemen* G , dilambangkan dengan \bar{G} , adalah graf yang himpunan titiknya sama dengan himpunan titik G ; dan dua titik u, v di \bar{G} bertetangga jika dan hanya jika u, v di G tidak bertetangga. Contoh graf dan komplemennya diberikan pada Gambar 2.7 [Ketut: 76-77].



Gambar 2.7 Graf dan Komplemennya

2.3 Penyajian Graf dalam Komputer

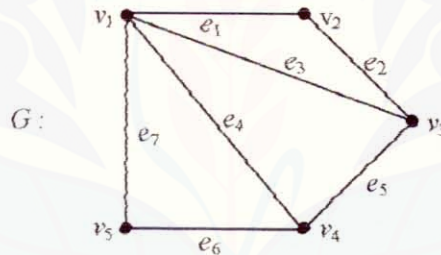
Ada beberapa cara untuk menyajikan sebuah graf di dalam komputer. Penyajian yang sering digunakan adalah matriks. Ada dua matriks yang akan digunakan yaitu *matriks adjacent* dan *matriks incident* [I Ketut: 85].

Misal $G(n,m)$ graf dengan n titik dan m sisi. Matriks dengan ukuran $n \times n$ dinotasikan dengan $A(G) = [a_{ij}]$, dinamakan *matriks adjacent* dari G , dimana

$$a_{ij} = \begin{cases} 0, & \text{jika titik } v_i \text{ tidak bertetangga dengan } v_j, \\ 1, & \text{jika titik } v_i \text{ bertetangga dengan } v_j. \end{cases}$$

Matriks dengan ukuran $n \times m$ dinotasikan dengan $M(G) = [m_{ij}]$, dinamakan *matriks incident*, dimana

$$m_{ij} = \begin{cases} 0, & \text{jika sisi } e_j \text{ tidak terkait dengan titik } v_i, \\ 1, & \text{jika } e_j \text{ terkait dengan } v_i \text{ dan } e_j \text{ bukan gelang,} \\ 2, & \text{jika } e_j \text{ terkait dengan } v_i \text{ dan } e_j \text{ adalah gelang.} \end{cases}$$



Gambar 2.8 Graf G untuk Ilustarsi Matriks Adjacent dan Matriks Incident

Matriks *adjacent* dan matriks *incident* dari graf pada Gambar 2.8 adalah sebagai berikut:

$$A(G) = \begin{matrix} & \begin{matrix} v_1 & v_2 & v_3 & v_4 & v_5 \end{matrix} \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix} \end{matrix}$$

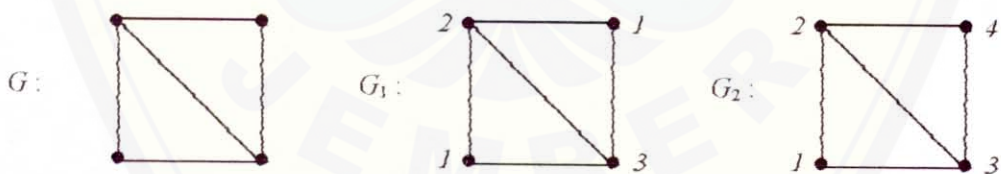
$$M(G) = \begin{matrix} & \begin{matrix} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \end{matrix} \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{matrix} & \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{matrix}$$

2.4 Pewarnaan Graf

Pewarnaan graf merupakan suatu pemberian warna pada salah satu elemen-elemennya (titik, sisi atau bidang), sehingga elemen-elemennya dapat diwarnai dengan warna yang berbeda menggunakan aturan tertentu. Dalam subbab ini akan dibahas pewarnaan titik dan pewarnaan sisi, dengan fokus pembahasan pada pewarnaan titik. Karena itu algoritma yang disajikan hanya algoritma dari pewarnaan titik.

2.4.1 Pewarnaan Titik

Pewarnaan titik pada graf G adalah suatu fungsi f dari himpunan titik $V(G)$ ke himpunan warna $\{1, 2, 3, \dots, w\}$ sehingga untuk setiap $e = (x, y)$ di G , $f(x) \neq f(y)$. Pewarnaan titik dapat pula didefinisikan sebagai pemberian warna untuk setiap titik dari G , sedemikian hingga dua titik yang bertetangga mempunyai warna yang berbeda. Bilangan kromatik $\chi(G)$ adalah jumlah warna minimal yang diperlukan untuk mewarnai titik pada graf G [Fletcher: 491]. Dengan kata lain jika $\chi(G) = n$ berarti titik-titik pada graf G dapat diwarnai dengan n warna tetapi tidak dapat diwarnai dengan $n-1$ warna. Maksimum warna yang dapat dipakai untuk mewarnai graf G adalah sejumlah titik pada graf G . Pewarnaan titik suatu graf adalah tidak tunggal. Sebagai contoh, pada Gambar 2.9, graf G dapat diwarnai dengan 2 cara seperti yang ditunjukkan pada graf G_1 dengan 3 pewarnaan dan graf G_2 dengan 4 pewarnaan dimana bilangan kromatik $\chi(G) = 3$.



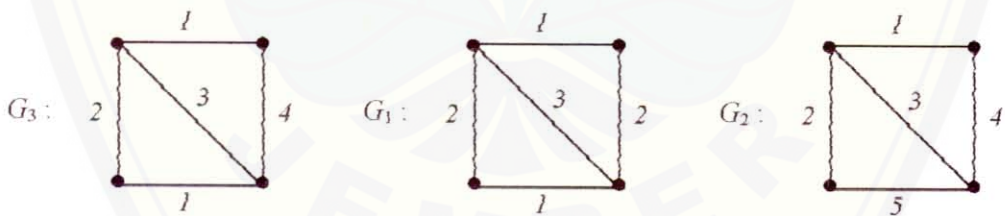
Gambar 2.9 Pewarnaan Titik Graf

Pewarnaan titik dari graf dapat diaplikasikan untuk permasalahan perencanaan jadwal (*problem of scheduling*). Misalkan kita mempunyai data kursus dan data dari calon peserta serta kelas kursus yang direncanakan. Kita ingin membuat jadwal mereka. Jika ada beberapa peserta merencanakan untuk

menempuh/mengambil dua kelas kursus, maka keduanya tidak dapat dilaksanakan dalam waktu yang bersamaan. Kondisi ini dapat di tunjukkan dengan sebuah graf G dimana titik-titik pada graf menunjukkan kelas kursus. Dua titik akan bertetangga jika dan hanya jika ada peserta merencanakan untuk menempuh/mengambil dua kelas yang berkorespondensi. Jika ada titik-titik yang terwarnai sama maka kelas kursus dapat dilaksanakan bersamaan, jika titik-titik terwarnai berbeda maka kelas kursus tidak dapat dilaksanakan secara bersamaan [Chartrand: 286].

2.4.2 Pewarnaan Sisi

Seperti pada pewarnaan titik, *pewarnaan sisi* pada graf G adalah suatu fungsi g dari himpunan sisi $E(G)$ ke himpunan warna $\{1,2,3,\dots,k\}$ sehingga untuk setiap $d = (x,y)$ dan $e = (x,z)$ di G , $g(d) \neq g(e)$, dengan kata lain sisi yang terkait mempunyai warna yang berbeda, pewarnaan sisi suatu graf juga tidak tunggal. Jumlah warna minimal yang diperlukan untuk mewarnai sisi pada graf G disebut *bilangan kromatik*, yang dinotasikan dengan $\chi'(G)$, yang dapat juga disebut sebagai *indeks kromatik*. Sebagai contoh, graf G pada Gambar 2.9, dapat diwarnai dengan 3 cara seperti yang ditunjukkan pada Gambar 2.10, yaitu G_1 dengan 3 pewarnaan, graf G_2 dengan 4 pewarnaan dan graf G_3 dengan 5 pewarnaan dimana bilangan indeks $\chi'(G) = 3$.



Gambar 2.10 Pewarnaan Sisi Graf

Pewarnaan sisi graf, digunakan pada permasalahan yang dikenal dengan *masalah penjadualan (timetabling problem)*. Misalkan dalam suatu sekolah, terdapat m guru $T_1, T_2, T_3, \dots, T_m$ dan n kelas $S_1, S_2, S_3, \dots, S_n$. Diketahui bahwa guru T_i harus mengajar kelas S_j untuk w_{ij} periode tiap minggu. Solusi masalah yang diinginkan adalah membuat jadwal untuk kelas-kelas tersebut dengan

meminimumkan periode. Permasalahan seperti di atas dapat ditunjukkan dengan menggunakan graf bipartit dengan partisi himpunan $T = \{T_1, T_2, T_3, \dots, T_m\}$ dan $S = \{S_1, S_2, S_3, \dots, S_n\}$, dimana sebuah titik T_i bertetangga dengan titik S_j yang dihubungkan dengan sisi w_{ij} jika seorang guru T_i harus mengajar di kelas S_j untuk w_{ij} periode. Dalam satu periode, seorang guru dapat mengajar satu kelas dan satu kelas dapat menerima pelajaran dari satu orang guru. Masalah penjadualan di sini adalah bagaimana mempartisi $E(G)$ dalam kemungkinan kecocokan atau sama dengan mewarnai sisi-sisi dari G dengan warna sejumlah minimum sisi. Permasalahan penjadualan yang dimodelkan menjadi graf bipartit tersebut mempunyai solusi dengan penjadualan kelas sebanyak $\Delta(G)$ [Chartrand: 300].

2.4.3 Algoritma Pewarnaan Titik

Algoritma (*algorithm*) adalah urutan langkah instruksi yang jelas dan rinci untuk menyelesaikan suatu masalah. Rancangan yang baik untuk suatu algoritma adalah menguraikan prosedur dalam beberapa subprosedur, dan subprosedur ini diuraikan lagi menjadi sub-subprosedur dan seterusnya. Metode ini disebut rancangan algoritma secara struktural yang memberikan kemudahan pemahaman logika algoritma. Instruksi dalam algoritma mempunyai beberapa karakteristik yaitu : Presisi, Unik, Terhingga, Masukan, Keluaran dan Umum [Richard: 135].

Terdapat beberapa algoritma pewarnaan titik yang telah ditemukan, antara lain *sequential coloring algorithm* (algoritma pewarnaan berurutan) dan *the Welsh and Powel algorithm* (algoritma Welsh Powel).

A. Algoritma Pewarnaan Berurutan

Pada algoritma pewarnaan berurutan, jumlah warna yang diperlukan untuk pewarnaan titik tergantung dari cara pelabelan titik. Algoritma ini dalam urutan rangkaian, kemungkinan terkecil warna untuk setiap titik tergantung dari graf yang diberikan [Chartrand: 292].

Berikut adalah algoritma pewarnaan berurutan yang digunakan untuk mewarnai graf G dengan $V(G) = \{v_1, v_2, v_3, \dots, v_n\}$.

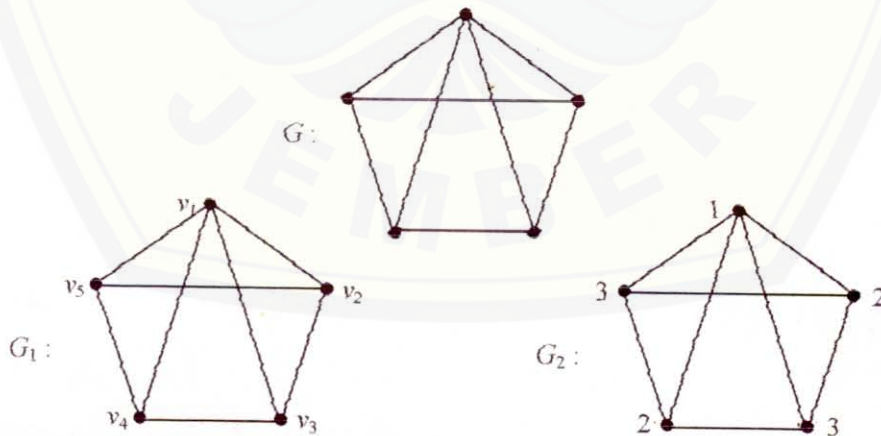
1. Menetapkan parameter i , yang digunakan untuk melabeli titik v_i

$$i \leftarrow 1.$$

2. Menetapkan warna k , untuk mewarnai v_i .
 $k \leftarrow 1$.
3. Warna terkecil yang mungkin, digunakan untuk mewarnai v_i .
 - 3.1 Urutkan warna yang bertetangga dengan v_i dengan aturan tanpa pengurangan dan tetapkan hasilnya dalam daftar D_i .
 - 3.2 Jika k tidak ada dalam daftar D_i , maka warnai k untuk v_i dan lanjutkan ke langkah 5; jika ada lanjutkan ke langkah 4.
4. Warna k berubah.
 $k \leftarrow k + 1$ dan kembali ke langkah 3.2
5. Parameter i berubah.
 Jika $i < n$, kemudian $i \leftarrow i + 1$, dan kembali ke langkah 2; jika ada kemungkinan lain, selesai.

Untuk sebarang graf G terhubung dengan order n , terdapat $n!$ kemungkinan cara untuk melabeli titik-titiknya. Beberapa dari pelabelan titik tersebut, dapat menetapkan $\chi(G)$ dengan tepat.

Sebagai contoh pewarnaan graf G pada Gambar 2.11 dengan menggunakan algoritma pewarnaan berurutan, ditunjukkan pada graf G_2 dan G_4 , dimana pewarnaan G_2 berdasarkan pelabelan titik pada G_1 , sedangkan pewarnaan G_4 berdasarkan pelabelan titik pada G_3 . Bilangan kromatik untuk graf G ini adalah $\chi(G) = 3$.





Gambar 2.11 Pewarnaan Titik Graf dengan Algoritma Pewarnaan Berurutan

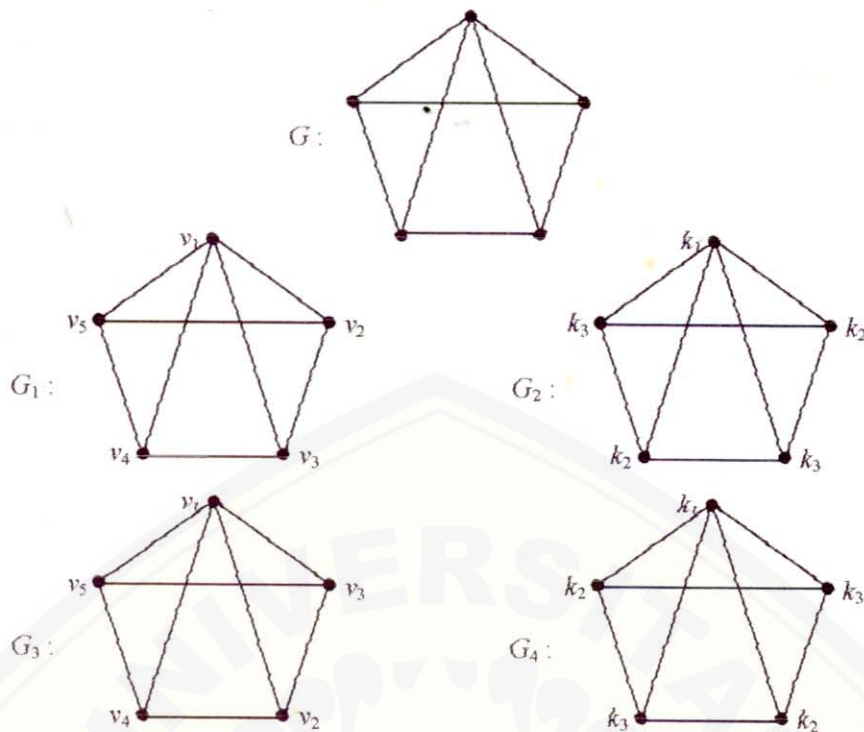
B. Algoritma Welsh Powel

Pewarnaan dengan menggunakan algoritma Welsh Powel prosedurnya hampir sama seperti pada algoritma pewarnaan berurutan, tetapi dalam algoritma Welsh Powel, pelabelan titik ditentukan/ditetapkan dalam algoritma. Beberapa dari pelabelan titik juga dapat menetapkan $\chi(G)$ dengan tepat.

Berikut adalah algoritma Welsh Powel yang digunakan untuk mewarnai graf G [Fletcher: 494].

1. Labeli titik-titik $V(G) = \{ v_1, v_2, v_3, \dots, v_n \}$ dengan aturan $\deg(v_1) \geq \deg(v_2) \geq \deg(v_3) \geq \dots \geq \deg(v_n)$.
2. Tetapkan warna k_1 untuk v_1 .
3. Buat daftar dari titik-titik dengan aturan pelabelan titik, tetapkan warna k_1 untuk setiap titik yang tidak bertetangga dengan v_1 .
4. Misalkan i sebagai bilangan asli terkecil dari v_i yang belum terwarnai. Tetapkan warna yang belum terpakai k untuk v_i dan pada daftar titik-titik yang belum terwarnai dengan aturan pelabelan titik, tetapkan warna k untuk setiap titik yang tidak bertetangga dengan sebarang titik dengan warna k .
5. Jika semua titik terwarnai, selesai, jika belum kembali ke langkah 4.

Sebagai contoh, pewarnaan graf G pada Gambar 2.12 dengan algoritma Welsh Powel di tunjukkan oleh G_2 dengan pelabelan titik graf G_1 , dan G_4 dengan pelabelan titik graf G_3 .



Gambar 2.12 Pewarnaan Titik Graf dengan Algoritma Welsh Powell

2.5 Penjadualan

Penjadualan merupakan salah satu studi matematika tentang proses pembuatan jadwal yang melibatkan waktu dan kegiatan. Penjadualan adalah suatu pencatatan atau penskemaan dari waktu secara sistematis sebagai daftar/rencana waktu pelajaran atau pekerjaan yang harus dijalankan, atau menyatakan urutan/rangkaian urutan dari peristiwa/kejadian. Penjadualan dapat juga diartikan sebagai tata cara atau prosedur penulisan keterangan untuk tujuan tertentu, biasanya menetapkan rangkaian urutan dan waktu yang tepat untuk setiap rincian di dalam prosedur.

2.6 Semester Pendek

Semester Pendek yang diselenggarakan oleh Fakultas MIPA Universitas Jember dapat dilaksanakan diantara dua semester reguler. Pelaksanaan Semester Pendek bertujuan:

1. mempercepat mahasiswa dalam menyelesaikan studinya,

2. meningkatkan motivasi belajar, efisiensi dan produktivitas,
3. memperbaiki prestasi belajar mahasiswa.

Pelaksanaan Semester Pendek diatur sebagai berikut:

1. dapat dilaksanakan pada masa libur panjang (bulan Juli-Agustus),
2. jumlah minimum peserta mata kuliah sebanyak 10 mahasiswa,
3. beban studi maksimum yang dapat ditempuh oleh mahasiswa sebanyak 10 SKS,
4. mata kuliah yang boleh diprogramkan adalah mata kuliah yang pernah diprogramkan dalam Kartu Rencana Studi (KRS) pada semester reguler sebelumnya dan tidak dibatalkan dalam semester bersangkutan,
5. mata kuliah yang diprogramkan terbatas pada Mata Kuliah Dasar Keahlian (MKDK), Mata Kuliah Keahlian (MKK) dan Mata Kuliah Pilihan (MKP),
6. jika mata kuliah ada praktikumnya, maka mahasiswa harus sudah lulus praktikum,
7. mata kuliah yang ditawarkan pada Semester Pendek ditetapkan oleh fakultas atas usulan jurusan,
8. mata kuliah yang diprogramkan oleh mahasiswa dalam Semester Pendek tidak berdasarkan pada perolehan indeks prestasi,
9. mahasiswa yang menempuh Semester Pendek diwajibkan mengisi KRS semester pendek yang disediakan oleh Fakultas,
10. indeks prestasi yang diperoleh dari Semester Pendek tidak dapat digunakan untuk menentukan pengambilan mata kuliah yang diprogramkan semester reguler berikutnya,
11. Semester Pendek dapat juga diikuti oleh mahasiswa dari Fakultas lain pada mata kuliah *Resource Sharing* / layanan di fakultas MIPA dengan persetujuan Fakultas yang bersangkutan,
12. hal-hal lain yang tidak diatur dalam pelaksanaan Semester Pendek tetap mengikuti aturan umum yang berlaku dalam semester reguler.



BAB III

HASIL DAN PEMBAHASAN

Pada bab ini dijelaskan mengenai algoritma pewarnaan titik untuk penjadualan, elemen-elemen penjadualan, urutan prosedur penjadualan, juga proses dan pengolahan data hingga menjadi sebuah jadwal. Untuk memperjelas proses penjadualan diberikan ilustrasi penjadualan Semester Pendek di Jurusan Matematika tahun 2002. Pada bagian terakhir akan disusun algoritma program yang nantinya diaplikasikan dalam bahasa pemrograman.

3.1 Algoritma Pewarnaan Penjadualan

Dalam penyusunan jadwal digunakan metode pewarnaan titik. Algoritma pewarnaan titik yang digunakan merupakan perpaduan atau gabungan dua algoritma, yaitu algoritma berurutan dan algoritma Welsh Powel. Alasan penggabungan kedua algoritma ini adalah untuk mendapatkan algoritma yang sesuai untuk menyelesaikan permasalahan yang ada. Algoritma pewarnaan titik yang baru ini, disebut Algoritma Pewarnaan Penjadualan. Beberapa prosedur yang ada pada kedua algoritma, antara lain penetapan label titik dan pemberian warna secara berurutan pada titik graf. Berikut adalah Algoritma Pewarnaan Penjadualan untuk mewarnai graf G .

1. Labeli titik-titik $V(G) = \{v_1, v_2, v_3, \dots, v_n\}$ dengan aturan $\deg(v_1) \geq \deg(v_2) \geq \deg(v_3) \geq \dots \geq \deg(v_n)$. Tetapkan parameter i pada titik yang terlabeli.

$$i \leftarrow 1.$$

2. Menetapkan warna w_k , untuk mewarnai v_i .

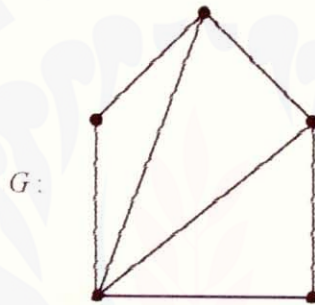
$$w_k \leftarrow 1.$$

3. Warna terkecil yang mungkin, digunakan untuk mewarnai v_i .

3.1 Jika v_i belum terwarnai maka urutkan warna titik dari titik yang bertetangga dengan v_i dengan aturan tanpa pengurangan dan tetapkan hasilnya dalam daftar D_i . Jika v_i sudah terwarnai maka lanjutkan ke langkah 5.

- 3.2 Jika tidak ada w_k dalam daftar D_i , maka warnai w_k untuk v_i dan untuk titik lainnya yang tidak bertetangga dengan v_i dan belum terwarnai, kemudian lanjutkan ke langkah 5; jika ada w_k dalam daftar D_i lanjutkan ke langkah 4.
4. Warna w_k berubah.
 $k \leftarrow k + 1$ dan kembali ke langkah 3.2
5. Parameter i berubah.
 Jika $i < n$, kemudian $i \leftarrow i + 1$, dan kembali ke langkah 2; jika $i = n$ maka semua titik terwarnai dan proses pewarnaan titik selesai.

Sebagai contoh pewarnaan penjadualan adalah sebagai berikut. Misalkan graf G pada Gambar 3.1, akan diwarnai menggunakan algoritma pewarnaan penjadualan.

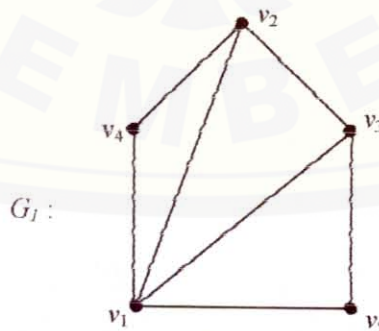


Gambar 3.1 Graf Ilustrasi Pewarnaan

- **Proses pewarnaan titik:**

A : 1. label titik pada graf G berdasarkan derajat titik, yaitu:

$$\deg(v_1) = 4 ; \deg(v_2) = \deg(v_3) = 3 ; \deg(v_4) = \deg(v_5) = 2$$



Gambar 3.2 Pelabelan Titik Graf dengan Aturan Derajat Titik

2. Tetapkan parameter $i = 1$, untuk titik v_i .

$$\{v_i = v_1\}$$

2. Tetapkan warna w_k , untuk mewarnai titik v_i .

$$\{k = 1\}$$

3.1 Titik yang bertetangga dengan v_i , adalah titik v_2, v_3 dan v_4 .

Daftar warna dari titik yang bertetangga dengan $v_i, D_1 = \{-, -, -\}$

3.2 Warna $k = 1$ tidak ada dalam daftar D_1 , maka warna untuk $v_i = w_1$.

Tidak ada titik yang tidak bertetangga dengan v_i .

5. Parameter i berubah.

$$\{i = 1 < 5\} \{i = 1 + 1 \rightarrow i = 2\}$$

B : 2. Tetapkan warna k , untuk mewarnai titik v_2 .

$$\{k = 1\}$$

3.1 Titik yang bertetangga dengan v_2 adalah titik v_1, v_3 dan v_4 .

Daftar warna dari titik yang bertetangga dengan $v_i, D_2 = \{w_1, -, -\}$

3.2 Warna $k = 1$ ada dalam daftar D_2 .

4. Warna k berubah.

$$\{k = 1 + 1 \rightarrow k = 2\}$$

3.2 Warna $k = 2$ tidak ada dalam daftar D_2 , maka warna untuk $v_2 = w_2$.

Titik yang tidak bertetangga dengan v_2 dan belum terwarnai adalah titik v_5 , maka warna untuk $v_5 = w_2$.

5. Parameter i berubah.

$$\{i = 2 < 5\} \{i = 2 + 1 \rightarrow i = 3\}$$

C : 2. Tetapkan warna k , untuk mewarnai titik v_3 .

$$\{k = 1\}$$

3.1 Titik yang bertetangga dengan v_3 adalah titik v_1, v_2 dan v_5 .

Daftar warna dari titik yang bertetangga dengan $v_3, D_3 = \{w_1, w_2, w_2\}$

3.2 Warna $k = 1$ ada dalam daftar D_3 .

4. Warna k berubah.

$$\{k = 1 + 1 \rightarrow k = 2\}$$

3.2 Warna $k = 2$ ada dalam daftar D_3 .

4. Warna k berubah.

$$\{ k = 2 + 1 \rightarrow k = 3 \}$$

3.2 Warna $k = 3$ tidak ada dalam daftar D_3 , maka warna untuk $v_3 = w_3$.

Titik yang tidak bertetangga dengan v_3 dan belum terwarnai adalah titik v_4 , maka warna untuk $v_4 = w_3$.

5. Parameter i berubah.

$$\{ i = 3 < 5 \} \{ i = 3 + 1 \rightarrow i = 4 \}$$

3.1 Titik v_4 sudah terwarnai, warna untuk $v_4 = w_3$.

5. Parameter i berubah.

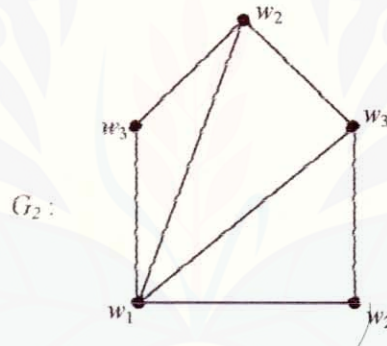
$$\{ i = 4 < 5 \} \{ i = 4 + 1 \rightarrow i = 5 \}$$

3.1 Titik v_5 sudah terwarnai, warna untuk $v_5 = w_2$.

5. Parameter $i = 5$. Pewarnaan selesai.

$$\{ i = 5 \}$$

Hasil pewarnaan titik pada graf G dapat dilihat pada graf G_2 :



Gambar 3.3 Graf Hasil Pewarnaan dengan Algoritma Pewarnaan Penjadualan

3.2 Elemen-Elemen Penjadualan

Dalam penyusunan jadwal perlu diperhatikan beberapa elemen penjadualan. Berikut akan dijelaskan mengenai elemen-elemen penjadualan Semester Pendek di Jurusan Matematika, elemen-elemen itu antara lain Mahasiswa, Dosen, Mata Kuliah, Periode dan Ruang Kuliah.

Elemen pertama adalah elemen Mahasiswa. Ada beberapa persyaratan mengenai elemen mahasiswa yang harus terpenuhi yaitu:

1. beban maksimum yang boleh ditempuh oleh mahasiswa sebanyak 12 SKS,

2. jika mata kuliah ada praktikumnya, maka mahasiswa harus lulus praktikum,
3. mahasiswa yang menempuh Semester Pendek wajib mengisi KRS semester pendek yang disediakan oleh Fakultas, untuk mempermudah atau memperlancar bagian akademik jurusan,

Elemen kedua adalah elemen Dosen. Dosen dapat berasal dari Fakultas MIPA atau dari Fakultas Ilmu Pendidikan dan Keguruan (FKIP), dimana seorang dosen dapat mengajar lebih dari 1 (satu) mata kuliah.

Elemen ketiga adalah elemen Mata Kuliah. Beberapa syarat yang harus dipenuhi antara lain:

1. jumlah minimum peserta mata kuliah Semester Pendek sebanyak 10 mahasiswa,
2. mata kuliah yang boleh diprogramkan adalah mata kuliah yang pernah diprogramkan dalam KRS pada semester reguler sebelumnya dan tidak dalam semester bersangkutan,
3. mata kuliah yang diprogramkan oleh mahasiswa dalam semester pendek tidak berdasarkan pada perolehan indeks prestasi.

Elemen keempat adalah elemen Periode. Semester Pendek dibagi dalam 5 (lima) periode, tiap periode adalah 100 menit dan antar periode terdapat waktu istirahat selama 10 menit, yang diperlihatkan pada Tabel 3.1 sebagai berikut:

**Tabel 3.1 Periode Penjadualan Semester Pendek
di Jurusan Matematika**

Periode	Waktu pelaksanaan
I	07.00 – 08.40
II	08.50 – 10.30
III	10.40 – 12.20
IV	12.30 – 14.10
V	14.20 – 16.00

Elemen kelima adalah elemen Ruang Kuliah. Pelaksanaan Semester Pendek di Jurusan Matematika menempati 3 (tiga) ruangan yang dapat digunakan untuk

ruang kuliah, dan memiliki kapasitas tempat duduk tertentu untuk mahasiswa yang menjadi prioritas persyaratan untuk penempatan kuliah. Ruang kuliah untuk melaksanakan proses belajar mengajar, yaitu 1 (satu) ruang kuliah di lantai I dan 2 (dua) ruang kuliah di lantai II.

3.3 Prosedur Penjadualan

Penjadualan yang akan didapat, terlebih dahulu harus memenuhi beberapa prosedur. Prosedur perencanaan jadwal dari awal hingga keluarnya jadwal untuk Semester Pendek di Jurusan Matematika adalah sebagai berikut:

1. Perencanaan Semester Pendek

Dimulai dengan mendaftar semua mata kuliah yang ada di Fakultas MIPA menjadi daftar mata kuliah di masing-masing Jurusan. Pihak Fakultas memberi keleluasaan kepada Jurusan tentang mata kuliah yang akan diselenggarakan, peraturan Semester Pendek ditetapkan oleh Fakultas.

2. Mata kuliah yang diusulkan oleh Jurusan.

Mata kuliah yang diusulkan Jurusan kepada pihak Fakultas sudah melalui seleksi di Jurusan, dan dosen untuk tiap mata kuliah sudah dikonfirmasi oleh masing-masing Jurusan. Fakultas menetapkan mata kuliah yang diusulkan oleh Jurusan untuk diselenggarakan dalam Semester Pendek.

Daftar mata kuliah tersebut oleh Jurusan ditawarkan ke mahasiswa, untuk selanjutnya mahasiswa memilih mata kuliah yang diinginkan.

3. Pendaftaran Semester Pendek.

Pendaftaran mata kuliah Semester Pendek di Jurusan Matematika melalui bagian Akademik dengan pengisian KRS oleh mahasiswa.

4. Data Mahasiswa Semester Pendek.

Data pendaftaran meliputi nama mahasiswa, NIM, mata kuliah, dan jumlah SKS.

5. Pengolahan data mahasiswa dan data mata kuliah,

Pengolahan data memperhitungkan persyaratan, perincian pengolahan data dijelaskan lebih lanjut pada sub bab 3.4.

6. Jadwal Semester Pendek,

3.4 Pengolahan Data dan Penyusunan Jadwal

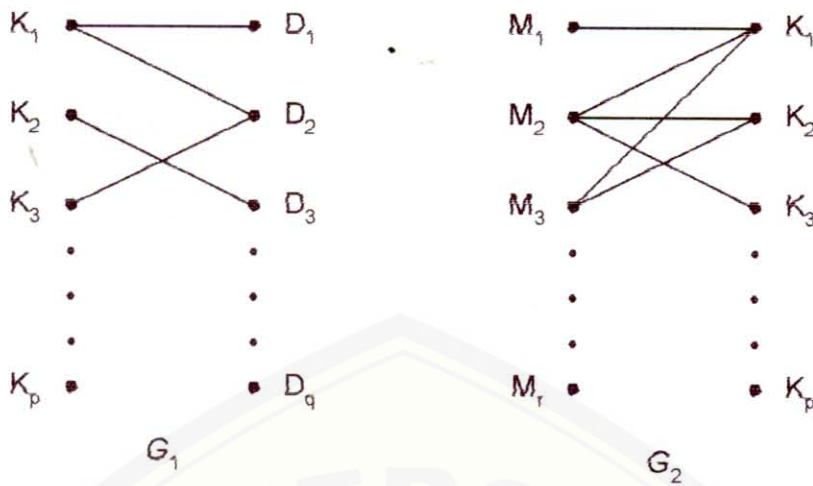
Proses penjadualan meliputi pengolahan data-data yang telah didapat, yaitu data mata kuliah dan data mahasiswa. Data mata kuliah terdiri dari mata kuliah, kode mata kuliah, SKS tiap mata kuliah, dan dosen mata kuliah (bisa lebih dari 1 orang dosen), sedangkan data mahasiswa terdiri dari Nama mahasiswa, Nomor Induk Mahasiswa (NIM), dan mata kuliah yang diminati.

Dari data-data tersebut, data kode mata kuliah dapat diartikan sebagai penciri yang dapat disamakan dengan mata kuliah, begitu juga dengan NIM disamakan dengan nama mahasiswa. Data SKS yang diambil oleh tiap mahasiswa diasumsikan telah memenuhi persyaratan maksimal yaitu 12 SKS, sedangkan SKS tiap mata kuliah digunakan untuk menetapkan jumlah pertemuan tiap minggu dalam Semester Pendek di Jurusan Matematika dimana untuk 1 SKS berarti 1 pertemuan. Data yang ada dapat dibagi menjadi dua tabel, yaitu tabel antara untuk mempermudah penganalisaan yang ditunjukkan pada Tabel 3.2.

Tabel 3.2 Data Mata Kuliah Dosen dan Mahasiswa

Mata Kuliah	SKS	Dosen			Mahasiswa	Mata Kuliah			
		1	2	3		1	2	3	4

Ketiga data di atas, yaitu mata kuliah, dosen, dan mahasiswa, dapat dibuat sebagai graf, dimana titik dimisalkan sebagai mahasiswa, mata kuliah dan dosen, sedangkan sisi sebagai pemisalan jika seorang mahasiswa menempuh satu mata kuliah, dan juga jika dosen mengajar mata kuliah. Graf yang terbentuk merupakan graf bipartit, karena dapat dikelompokkan menjadi 2 (dua) himpunan, yaitu himpunan 1 adalah himpunan titik mahasiswa dan dosen dan himpunan 2 adalah himpunan titik mata kuliah. Sebagai gambaran graf dari data ditunjukkan pada Gambar 3.4 sebagai berikut:



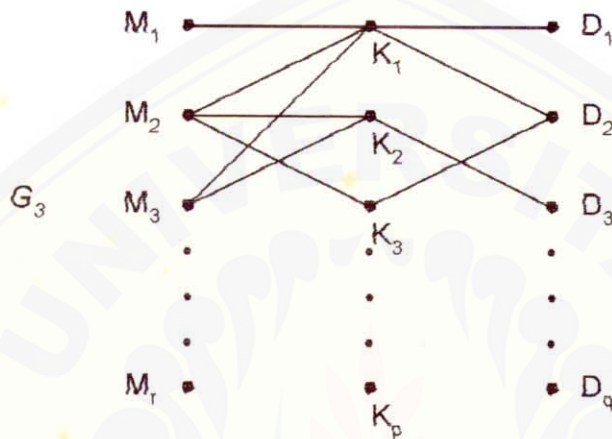
Gambar 3.4 Graf Bipartit Mata Kuliah-Dosen dan Mata Kuliah-Mahasiswa

Graf G_1 pada Gambar 3.4 merupakan graf bipartit dengan himpunan mata kuliah dan dosen, dimana himpunan titik $\{K_1 \dots K_p\}$ menyatakan mata kuliah, dan himpunan titik $\{D_1 \dots D_q\}$ menyatakan dosen. Sisi yang menghubungkan K_i dan D_j menyatakan bahwa dosen D_j mengajar mata kuliah K_i untuk suatu $i = 1, 2, \dots, p$ dan $j = 1, 2, \dots, q$. Derajat titik K_i menyatakan jumlah dosen yang mengajar mata kuliah K_i , sedangkan derajat pada titik D_j menyatakan jumlah mata kuliah yang diajar dosen.

Graf G_2 pada Gambar 3.4 merupakan graf bipartit dengan himpunan mata kuliah dan mahasiswa, dimana himpunan titik $\{K_1 \dots K_p\}$ menyatakan mata kuliah, dan himpunan titik $\{M_1 \dots M_r\}$ menyatakan mahasiswa. Sisi yang menghubungkan M_k dan K_i menyatakan bahwa mahasiswa M_k mengikuti mata kuliah K_i . Derajat pada titik K_i menyatakan jumlah mahasiswa yang mengikuti mata kuliah K_i , sedangkan derajat pada titik M_k menyatakan jumlah mata kuliah yang ditempuh oleh mahasiswa M_k .

Salah satu persyaratan dalam Semester Pendek di Jurusan Matematika adalah ketentuan jumlah minimum mahasiswa yang memilih suatu mata kuliah. Persyaratan ini dapat diketahui dari derajat titik mata kuliah K_i , yaitu 10. Jika derajat K_i untuk suatu $i = 1, 2, \dots, p$ kurang dari 10 maka titik K_i dihapus artinya mata kuliah K_i tidak diadakan dalam Semester Pendek di Jurusan Matematika.

Graf G_1 dan G_2 pada Gambar 3.4 dari data dapat diringkas menjadi satu graf bipartit yaitu graf G_3 , dengan menggabungkan antara himpunan dosen dengan mahasiswa. Penggabungan ini berarti mengasumsikan dosen mengikuti mata kuliah, dan jika dua dosen mengajar satu mata kuliah yang sama dan salah satu dosen tersebut juga mengajar mata kuliah yang lain maka modifikasi pergantian dosen untuk tiap-tiap mata kuliah harus sama yaitu pada setengah masa kuliah. Graf gabungan G_3 dibuat untuk meringkas proses pewarnaan mejadi satu kali pewarnaan. Graf dari gabungan ditunjukkan pada Gambar 3.2 sebagai berikut:



Gambar 3.5 Graf Bipartit Mata Kuliah-Dosen-Mahasiswa

Setelah didapat graf bipartit G_3 maka proses selanjutnya adalah membuat graf baru G_4 yang ditunjukkan pada Gambar 3.6, yaitu graf mata kuliah berdasarkan graf G_3 . Graf G_4 terdiri dari himpunan titik mata kuliah dimana himpunan sisinya memenuhi ketentuan bahwa jika ada Mahasiswa yang mengikuti lebih dari satu mata kuliah maka mata kuliah itu saling terhubung.

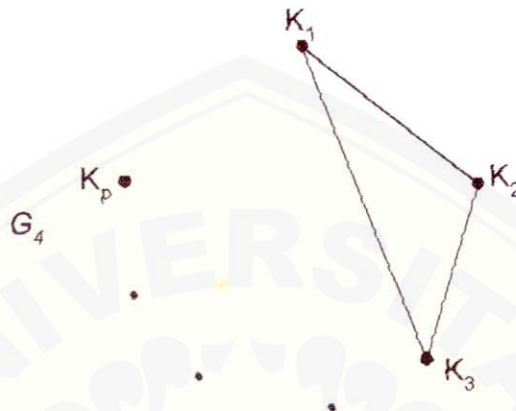
Sisi Graf G_4 dimaksudkan bahwa jika mahasiswa menempuh dua mata kuliah sekaligus, maka mata kuliah tersebut tidak boleh dilaksanakan secara bersamaan. Sebagai gambaran, graf mata kuliah dari Gambar 3.5 dapat dijelaskan sebagai berikut:

- titik M_1 , D_1 , dan D_3 mempunyai derajat 1, ini berarti mata kuliah yang bertetangga dengan masing-masing titik tersebut tidak terhubung,
- titik M_3 , dan D_2 mempunyai derajat 2, titik M_3 bertetangga dengan titik K_1 dan K_2 , berarti dalam graf mata kuliah titik K_1 dan K_2 terhubung,

begitu juga dengan titik D_2 , titik D_2 bertetangga dengan titik K_1 dan K_3 , berarti dalam graf mata kuliah titik K_1 dan K_3 terhubung,

- titik M_2 mempunyai derajat 3 dan bertetangga dengan titik K_1 , K_2 , dan K_3 , berarti dalam graf mata kuliah titik K_1 dan K_2 terhubung, titik K_1 dan K_3 terhubung dan titik K_2 dan K_3 terhubung

Graf mata kuliah G_4 dari Gambar 3.5 ditunjukkan pada Gambar 3.6 sebagai berikut:



Gambar 3.6 Graf Mata Kuliah

Setelah didapat graf G_4 maka proses selanjutnya adalah pewarnaan titik dengan menggunakan algoritma pewarnaan penjadualan yang telah ditetapkan. Pewarnaan ini akan menunjukkan bisa tidaknya mata kuliah dilaksanakan bersamaan. Jika setelah pewarnaan terdapat titik yang mempunyai warna yang sama maka mata kuliah yang diwakili oleh titik tersebut dapat dilaksanakan dalam periode yang sama, begitu juga sebaliknya jika mempunyai warna yang berbeda maka mata kuliah dilaksanakan tidak bersamaan.

Jumlah titik yang berwarna sama berpengaruh terhadap penempatan ruangan. Jika terdapat 3 (tiga) titik dengan warna yang sama maka perkuliahan masih dapat dilaksanakan secara bersamaan, tetapi jika terdapat lebih dari 3 (tiga) maka titik ke-4 dan seterusnya dapat dimisalkan mempunyai warna yang berbeda serta menempati periode berikutnya. Hasil pewarnaan titik graf G_4 dengan menggunakan algoritma pewarnaan penjadualan ditunjukkan pada Gambar 3.7, yang berarti mata kuliah M_1 , M_2 , dan M_3 dilaksanakan dalam periode yang berbeda.

3.5 Algoritma Program Penjadualan

Secara garis besar dapat dikatakan bahwa program yang dibuat adalah dengan memperhatikan masukan (*input*) data yang berupa data mata kuliah, data dosen dan data mahasiswa. Dari data-data masukan diubah menjadi data matriks yang kemudian diproses menggunakan algoritma pewarnaan penjadwalan, hasil akhir merupakan susunan jadwal. Algoritma programnya sebagai berikut:

Inisialisasi :

Jumlah ruangan = 3.

Jumlah periode 1 hari = 5.

Syarat minimum mahasiswa = 10.

Kapasitas ruangan 1 = 50.

Kapasitas ruangan 2 = 50.

Kapasitas ruangan 3 = 35.

Baca :

Masukan data berupa :

data mata kuliah = mata kuliah, jumlah SKS

data dosen = dosen, mata kuliah yang diajarkan (maksimal 3 mata kuliah)

data mahasiswa = nama, mata kuliah yang diambil (maksimal 4 mata kuliah)

Konversi Masukan Data :

Mengubah masukan data menjadi matriks data untuk mempermudah proses perhitungan. Matriks data dioperasikan menjadi matriks data antara matriks data mata kuliah - dosen dan mata kuliah - mahasiswa.

Hitung :

Jumlah peserta mata kuliah dan persyaratan jumlah minimum mahasiswa (10 mahasiswa tiap mata kuliah).

Periksa :

Apakah jumlah peserta memenuhi syarat minimum peserta mata kuliah. Jika memenuhi maka mata kuliah terus diproses.

Konversi Data :

Mengubah matriks data mata kuliah - mahasiswa menjadi matriks adjacent mata kuliah.

Hitung :

Jumlah elemen baris dari matriks adjacent, dan pengurutan elemen berdasarkan jumlah elemen baris.

Proses pewarnaan elemen baris, dan pengurutan, dan jumlah total warna.

Pengaturan periode dan ruang berdasar hasil hitungan pewarnaan.

Periksa :

Jumlah mahasiswa dan kapasitas tiap ruangan yang akan ditempati dari hasil pengaturan periode dan ruang.

Konversi Data :

Mengubah matriks hasil pengaturan periode dan kapasitas ruangan menjadi matriks mata kuliah data mata kuliah untuk jadual 1 hari.

Modifikasi mata kuliah 1 hari menjadi jadual perkuliahan 3 hari.

Periksa :

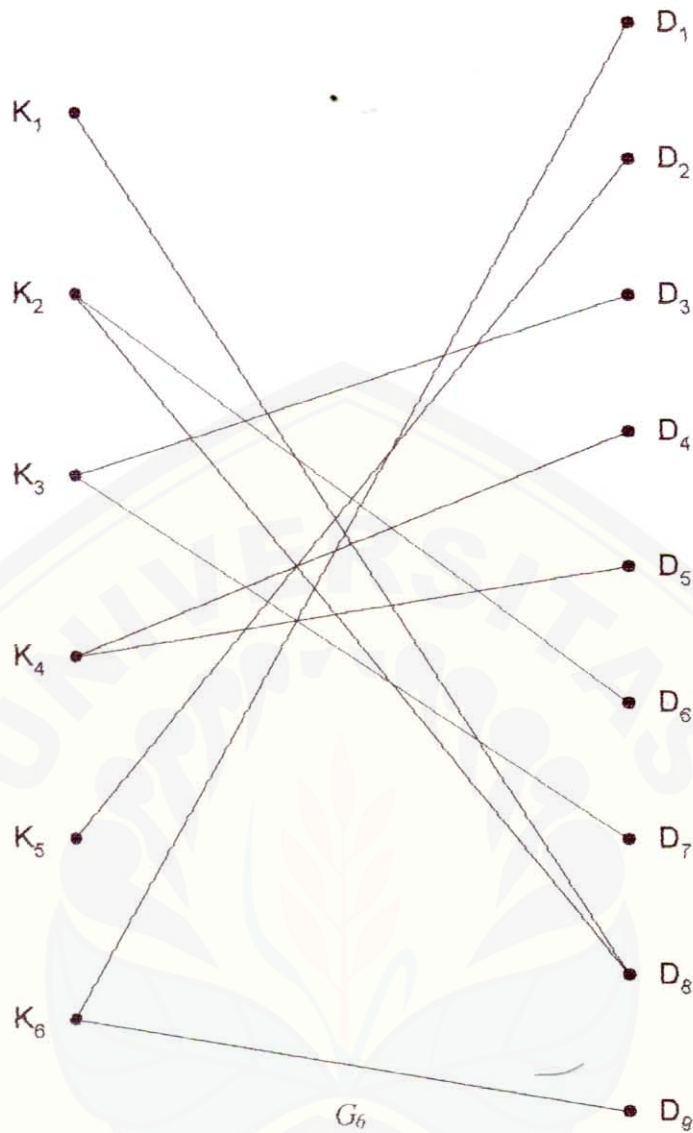
Jumlah SKS tiap mata kuliah. Jika SKS mata kuliah = 4 maka mata kuliah tersebut terjadual pada hari ke-4.

Tulis :

Jadual perkuliahan sesuai dengan masukan data.

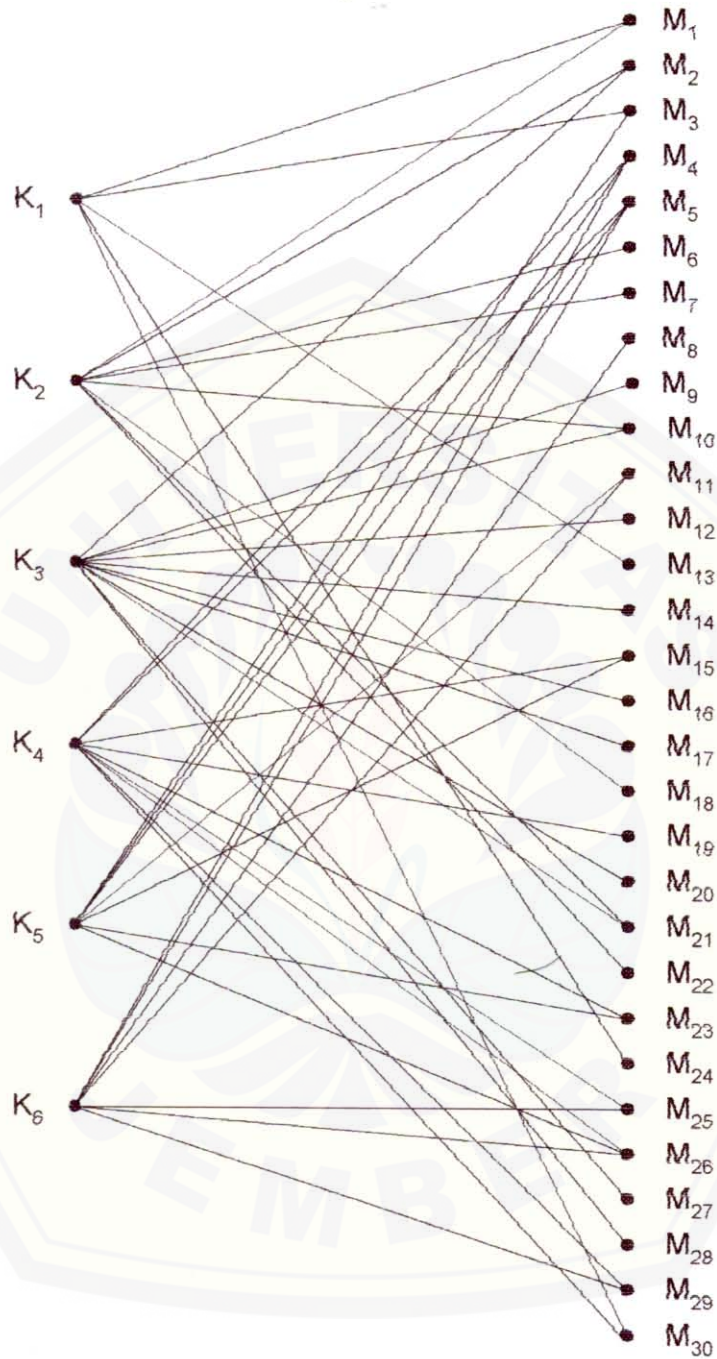
3.6 Ilustrasi Penjadualan

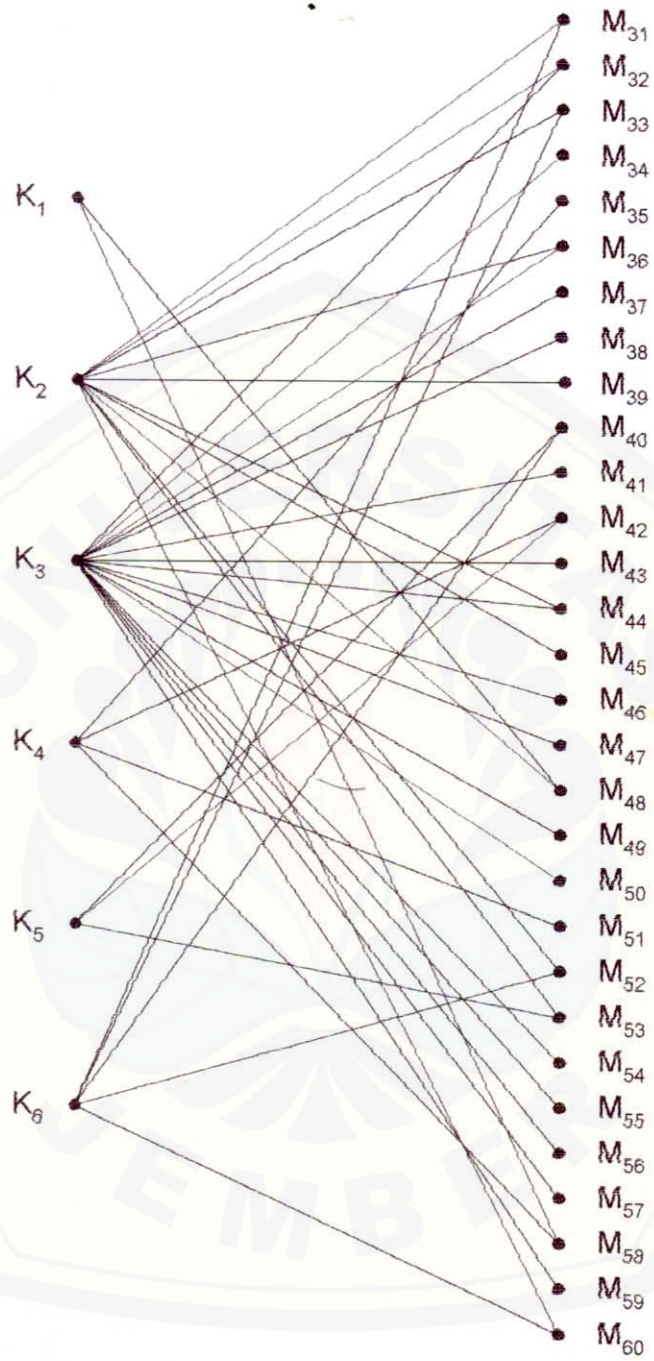
Proses dan pengolahan data untuk Penjadualan Semester Pendek di Jurusan Matematika akan diperjelas dengan mengilustrasikannya kedalam sebuah contoh kasus, dalam hal ini data-data yang akan digunakan adalah data-data Semester Pendek di Jurusan Matematika pada tahun 2002. Data-data elemen penjadualan ditunjukkan pada lampiran 1. Data mata kuliah dan data dosen pada lampiran 1 dapat digambarkan dengan graf G_6 pada Gambar 3.8 sebagai berikut:

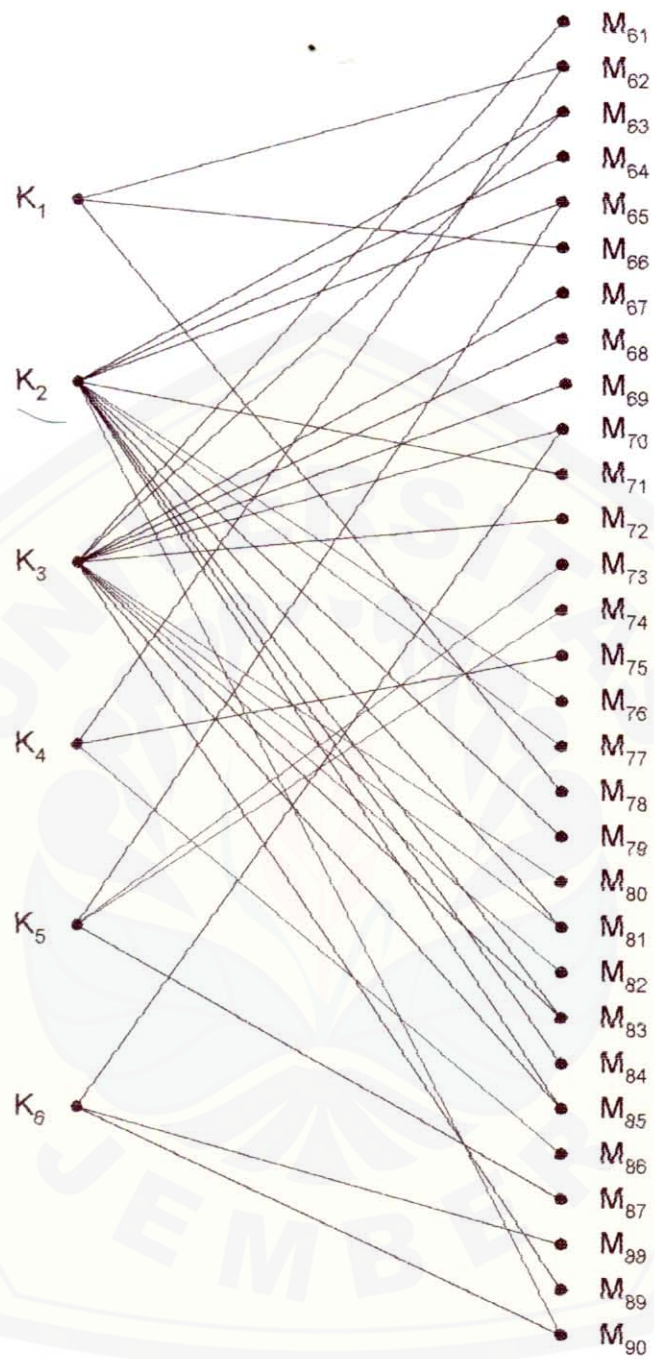


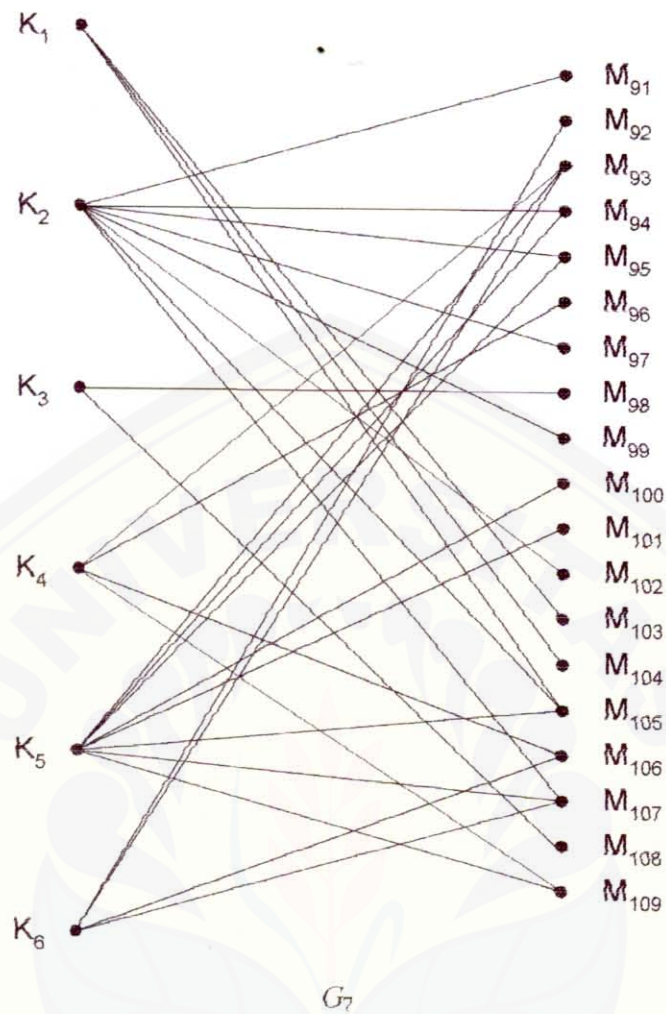
Gambar 3.8 Graf Data Mata Kuliah - Dosen

Data mata kuliah dan data mahasiswa pada lampiran 1 dapat digambarkan dengan graf G_7 pada Gambar 3.9 sebagai berikut:



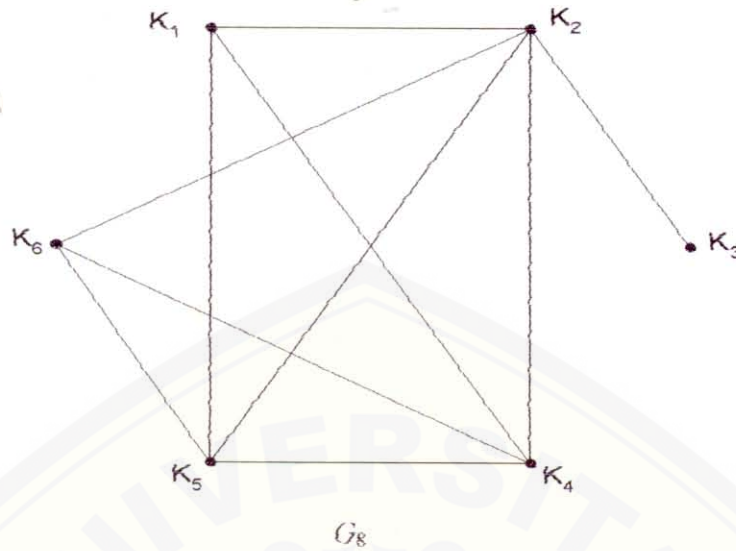






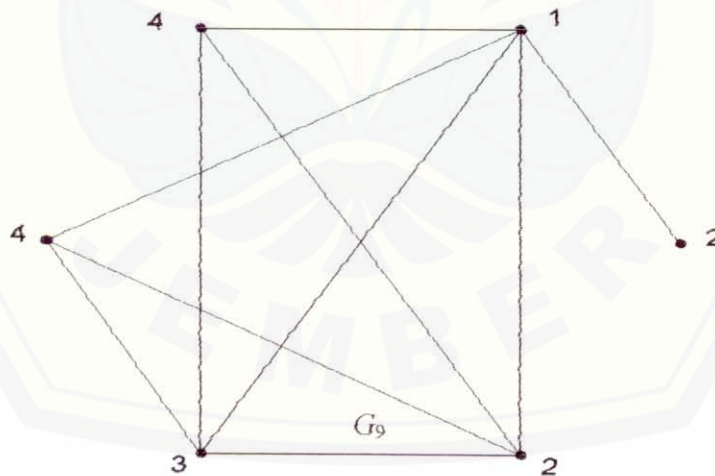
Gambar 3.9 Graf Data Mata Kuliah - Mahasiswa

Dari graf G_6 dan graf G_7 dibuat graf pewarnaan yang hasilnya adalah graf G_8 pada Gambar 3.10.



Gambar 3.10 Graf Pewarnaan

Setelah didapat graf G_8 maka dengan algoritma pewarnaan penjadwalan ditemukan bahwa titik K_1 dan K_6 berwarna sama, titik K_3 dan K_4 berwarna sama sedangkan titik K_2 dan titik K_5 berbeda, yang diperlihatkan oleh graf G_9 pada Gambar 3.11.



Gambar 3.11 Graf Hasil Pewarnaan

3.7 Analisa Program Penjadualan

Untuk mengetahui lebih jelas bagaimana proses pemrograman penjadualan, berikut akan diberikan contoh penjadualan beserta analisa pemrogramannya. Data-data elemen penjadualan ditunjukkan pada tabel berikut:

Tabel 3.6 Data Analisa Mata Kuliah

No.	Mata Kuliah	Jumlah SKS
1	Metode Numerik	3
2	Struktur Aljabar II	4
3	Persamaan Diferensial II	3
4	Matematika Deskrit	4
5	Statistika Dasar	3

Tabel 3.7 Data Analisa Dosen

No.	Dosen	Mata Kuliah	
		1	2
1	Rusli	Metode Numerik	Persamaan Diferensial II
2	Dafik	Metode Numerik	Persamaan Diferensial II
3	Kristiana	Matematika Deskrit	Struktur Aljabar II
4	Slamin	Matematika Deskrit	
5	Antonius	Struktur Aljabar II	
6	Alfian	Statistika Dasar	
7	Yuliani	Statistika Dasar	

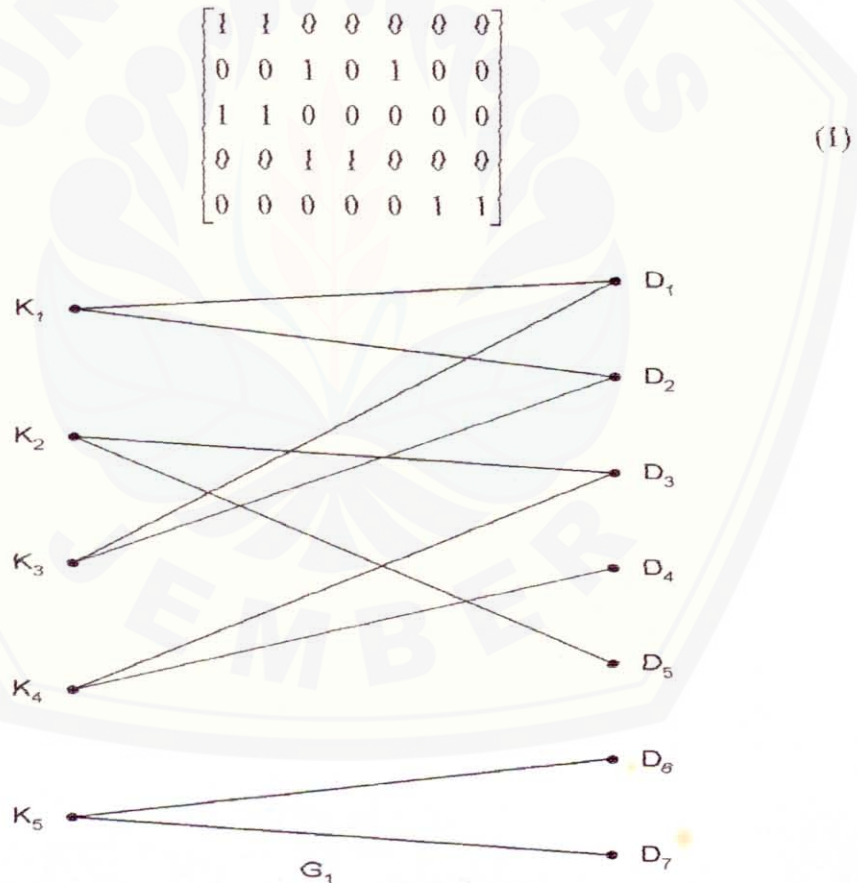
Tabel 3.8 Data Analisa Mahasiswa

No.	Mahasiswa	Mata Kuliah		
		1	2	3
1	Andamari	Metode Numerik	Persamaan Diferensial II	
2	Bahrul Muhith	Matematika Deskrit	Statistika Dasar	
3	Edi Santoso	Metode Numerik	Struktur Aljabar II	
4	Imron	Metode Numerik	Matematika Deskrit	
5	Jati Aninka	Struktur Aljabar II	Persamaan Diferensial II	
6	Lukman I.Y.P	Struktur Aljabar II	Matematika Deskrit	Statistika Dasar
7	Minartin	Struktur Aljabar II		
8	Tutut Triastuti	Matematika Deskrit	Persamaan Diferensial II	
9	Urip Damayanti	Struktur Aljabar II	Statistika Dasar	

Dalam analisa pemrograman penjadualan, beberapa konstanta yang dipakai yaitu ruangan = 3, periode = 5, minimal mahasiswa untuk 1 mata kuliah = 1, kapasitas ruangan 1 = 3, kapasitas ruangan 2 = 5, kapasitas ruangan 3 = 10.

Proses pemrograman yang pertama adalah memasukan (*input*) data, masukan data tersebut adalah mata kuliah dan jumlah SKS-nya, dosen dan mata kuliah yang diajarkan, dan masukan terakhir adalah mahasiswa dan mata kuliah yang diambil.

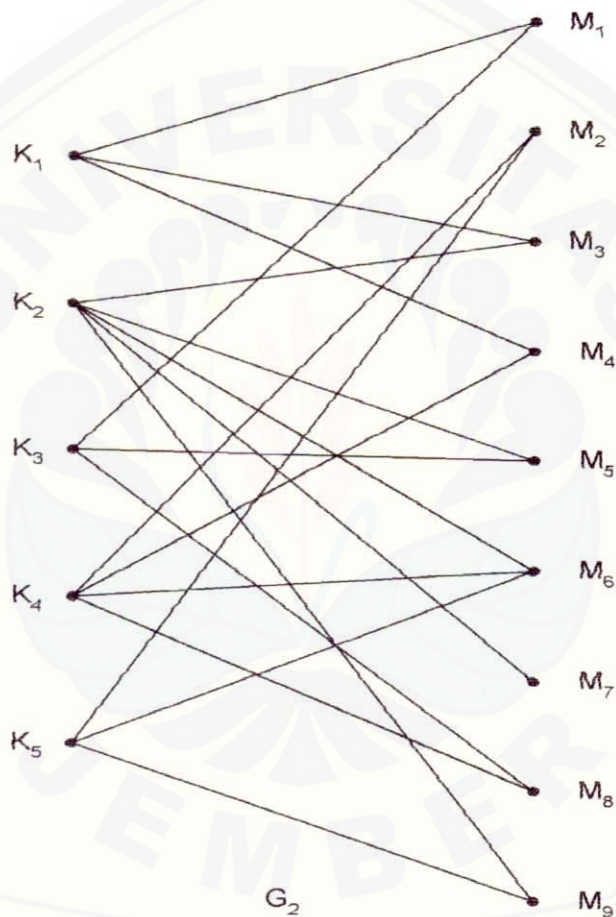
Masukan data diubah menjadi bentuk matriks, yang merupakan matriks adjacent mata kuliah dengan dosen dan mata kuliah dengan mahasiswa. Elemen matriks $[i,j]$ bernilai 0 jika dosen j tidak mengajar mata kuliah i , elemen matrik $[i,j]$ bernilai 1 jika dosen j mengajar mata kuliah i . Data analisa mata kuliah pada tabel 3.6 dan data analisa dosen tabel 3.7 dapat ditunjukkan dalam matriks (1) atau dalam graf seperti pada Gambar 3.12.



Gambar 3.12 Graf Analisa Mata Kuliah dan Dosen

sedangkan data analisa mata kuliah pada tabel 3.6 dan data analisa mahasiswa pada tabel 3.8 dapat ditunjukkan dalam matriks (2)) atau dalam graf seperti pada Gambar 3.13.

$$\begin{bmatrix}
 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\
 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1
 \end{bmatrix}
 \tag{2}$$



Gambar 3.13 Graf Analisa Mata Kuliah dan Mahasiswa

Dari matriks adjacent mata kuliah dengan mahasiswa dapat diketahui jumlah mahasiswa yang mengambil mata kuliah, yaitu dengan menjumlahkan elemen baris matriks. Jumlah mahasiswa digunakan untuk memeriksa persyaratan jumlah minimum mahasiswa yang mengambil mata kuliah, dan juga untuk persyaratan kapasitas ruangan. Matriks jumlah mahasiswa ditunjukkan pada matriks (3) sebagai berikut:

$$\begin{bmatrix} 3 \\ 5 \\ 3 \\ 4 \\ 3 \end{bmatrix} \quad (3)$$

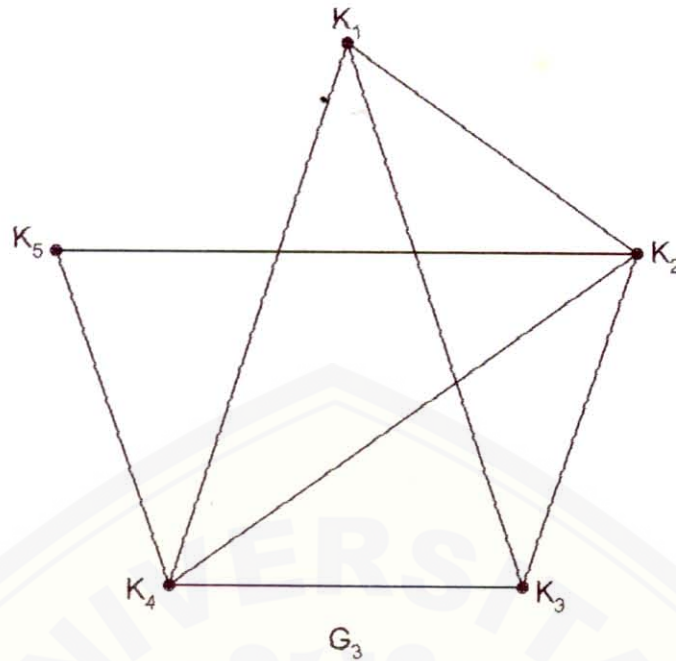
Jumlah mahasiswa yang mengikuti mata kuliah 1 adalah 3 orang, mata kuliah 2 adalah 5 orang, mata kuliah 4 adalah 4 orang, dan mata kuliah 5 adalah 3 orang. Jika jumlah mahasiswa kurang dari syarat minimum maka mata kuliah i dihapus, dengan cara merubah elemen baris i menjadi 0 pada matriks (2).

Proses selanjutnya adalah menggabungkan matriks (1) dengan matriks (2) yang telah melalui pemeriksaan persyaratan minimum mahasiswa. Matriks gabungan ditunjukkan pada matriks (4) sebagai berikut:

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

Dari matriks (4) dapat dibuat matriks adjacent mata kuliah, yang nantinya akan digunakan untuk proses pewarnaan. Matriks adjacent mata kuliah dari masukan adalah seperti matriks (5) atau dalam graf seperti ditunjukkan pada Gambar 3.14.

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix} \quad (5)$$

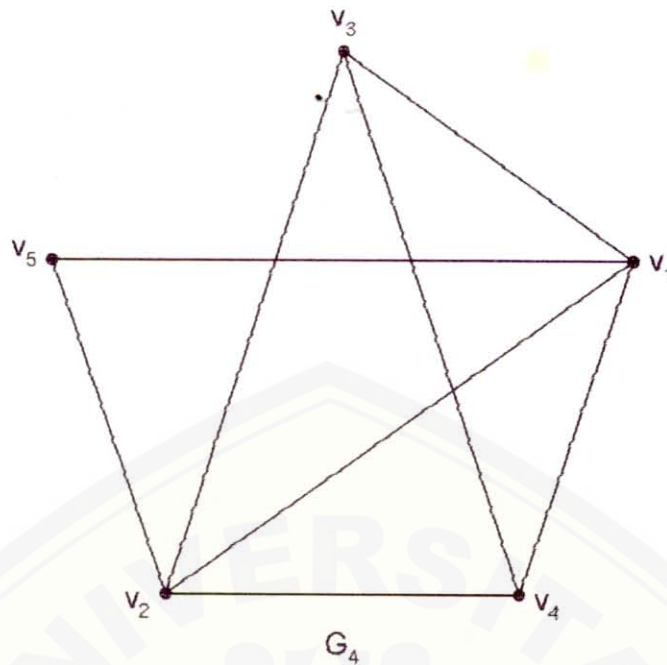


Gambar 3.14 Graf Adjacent Mata Kuliah

Proses pewarnaan pada matriks dimulai dengan menjumlahkan elemen baris yang merupakan derajat titik, ditunjukkan pada matriks (6), titik 1 berderajat 3, titik 2 berderajat 4, titik 3 berderajat 3, titik 4 berderajat 4 dan titik 5 berderajat 2. Kemudian mengurutkannya sesuai aturan dalam algoritma pewarnaan penjadwalan. Pengurutan matriks berpengaruh pada elemen yang lain, sehingga terjadi pertukaran antara baris matriks. Matriks (7) merupakan matriks terurut dan keterangan titiknya. Dalam graf, seperti gambar 3.15.

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 1 & 3 \\ 1 & 0 & 1 & 1 & 1 & 2 & 4 \\ 1 & 1 & 0 & 1 & 0 & 3 & 3 \\ 1 & 1 & 1 & 0 & 1 & 4 & 4 \\ 0 & 1 & 0 & 1 & 0 & 5 & 2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 2 & 4 \\ 1 & 1 & 1 & 0 & 1 & 4 & 4 \\ 1 & 1 & 0 & 1 & 0 & 3 & 3 \\ 0 & 1 & 1 & 1 & 0 & 1 & 3 \\ 0 & 1 & 0 & 1 & 0 & 5 & 2 \end{bmatrix}$$



Gambar 3.15 Graf Pelabelan Titik

Setelah matriks terurut sesuai aturan derajat, maka proses selanjutnya adalah pewarnaan, Langkah-langkahnya adalah jika pada baris i terdapat elemen j yang bernilai 0, maka elemen $[i, j]$ tersebut diubah menjadi i , untuk elemen j yang bernilai selain 0 pada baris ke i diubah menjadi 0. Matriks hasil pewarnaan ditunjukkan pada matriks (8). Pewarnaan untuk setiap titik ditunjukkan oleh elemen baris terkecil dari tiap-tiap kolom pada matriks (8) seperti ditunjukkan pada matriks (8a).

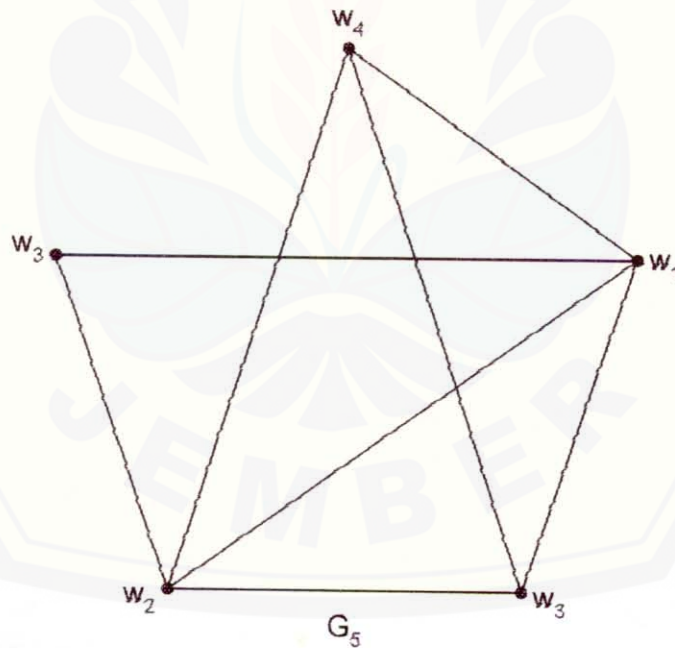
$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 3 & 0 & 3 \\ 4 & 0 & 0 & 0 & 4 \\ 5 & 0 & 5 & 0 & 5 \end{bmatrix} \tag{8}$$

$$\begin{bmatrix} 0 & \textcircled{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & \textcircled{2} & 0 \\ 0 & 0 & \textcircled{3} & 0 & \textcircled{3} \\ \textcircled{4} & 0 & 0 & 0 & 4 \\ 5 & 0 & 5 & 0 & 5 \end{bmatrix} \tag{8a}$$

Hasil pewarnaan ditunjukkan pada matriks (9) atau dalam graf seperti ditunjukkan pada Gambar 3.16. Titik 1 terwarnai dengan warna 4, titik 2 terwarnai dengan warna 1, titik 3 terwarnai dengan warna 3, titik 4 terwarnai dengan warna 2, dan titik 5 terwarnai dengan warna 3. Jumlah total warna yang digunakan adalah 4 warna, kemudian hasil pewarnaan diurutkan dari warna terkecil, seperti ditunjukkan pada matriks (10).

$$\begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 1 & 4 & 4 \\ 1 & 1 & 1 & 0 & 1 & 2 & 4 & 1 \\ 1 & 1 & 0 & 1 & 0 & 3 & 3 & 3 \\ 0 & 1 & 1 & 1 & 0 & 4 & 3 & 2 \\ 0 & 1 & 0 & 1 & 0 & 5 & 2 & 3 \end{bmatrix} \quad (9)$$

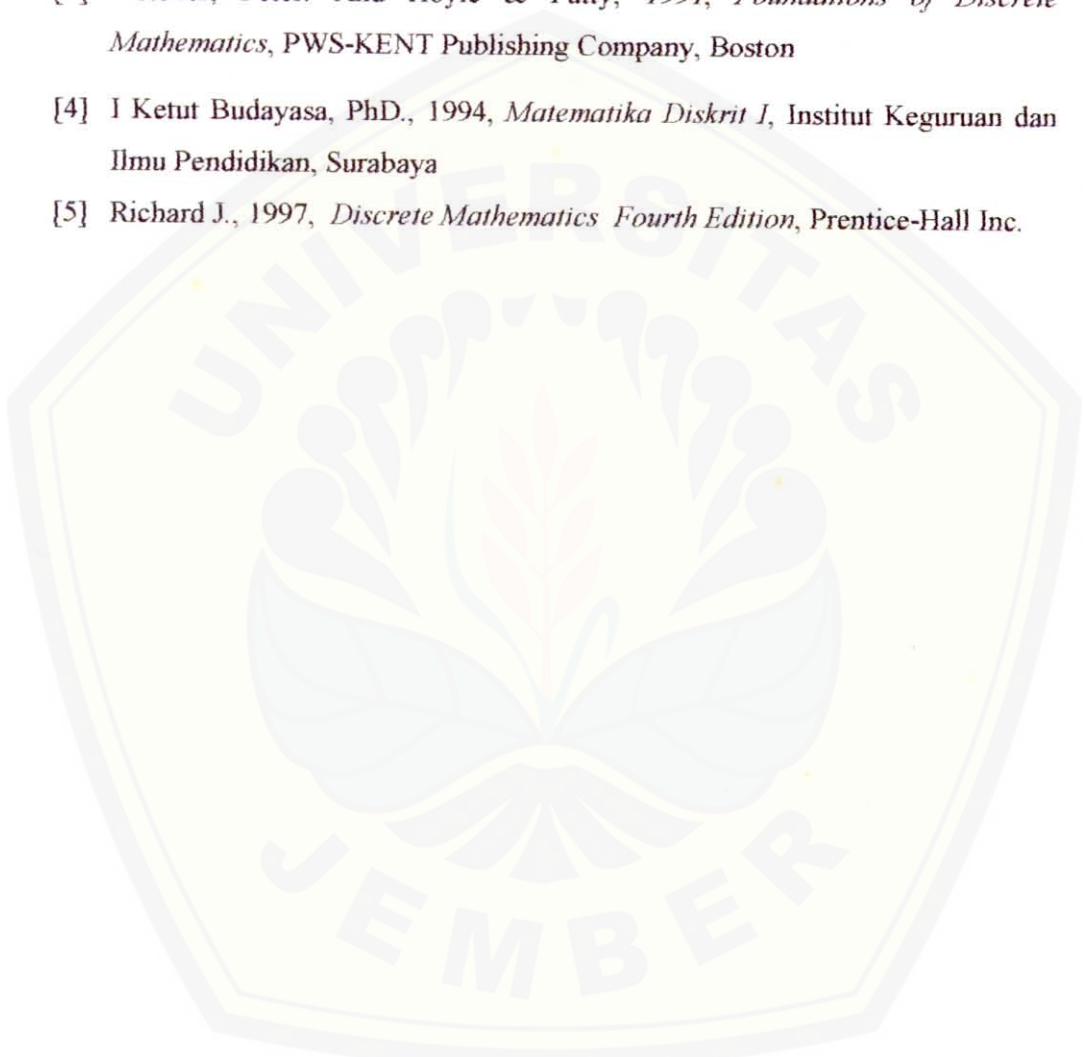
$$\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 2 & 4 & 1 \\ 0 & 1 & 1 & 1 & 0 & 4 & 3 & 2 \\ 1 & 1 & 0 & 1 & 0 & 3 & 3 & 3 \\ 0 & 1 & 0 & 1 & 0 & 5 & 2 & 3 \\ 1 & 0 & 1 & 1 & 1 & 1 & 2 & 4 \end{bmatrix} \quad (10)$$



Gambar 3.16 Graf Hasil Pewarnaan

DAFTAR PUSTAKA

- [1] ____, 2001, *Pedoman Penyelenggaraan Pendidikan FMIPA UNEJ*, Fakultas MIPA Universitas Jember
- [2] Chartrand, Gary and Ortrud R. Oellermann, 1993, *Applied and Algorithmic Graph Theory*, McGraw-Hill, Inc., USA.
- [3] Fletcher, Peter. And Hoyle & Patty, 1991, *Foundations of Discrete Mathematics*, PWS-KENT Publishing Company, Boston
- [4] I Ketut Budayasa, PhD., 1994, *Matematika Diskrit I*, Institut Keguruan dan Ilmu Pendidikan, Surabaya
- [5] Richard J., 1997, *Discrete Mathematics Fourth Edition*, Prentice-Hall Inc.



Lampiran 1 : Data Ilustrasi Mata Kuliah, Dosen dan Mahasiswa dari Pelaksanaan Semester Pendek di Jurusan Matematika tahun 2002.

Data Mata Kuliah

No.	Mata Kuliah	SKS
1	Geometri	4
2	Geometri Analitik	4
3	Kalkulus I	4
4	Kalkulus III	3
5	Persamaan Diferensial I	3
6	Statistika Dasar	3

Data Dosen

No.	Dosen	Mata Kuliah	
		1	2
1	Alfian	Statistika Dasar	
2	Dafik	Persamaan Diferensial I	
3	Didik	Kalkulus I	
4	Kosala	Kalkulus III	
5	Kristiana	Kalkulus III	
6	Kusno	Geometri	
7	Suharto	Kalkulus I	
8	Susanto	Geometri	Geometri Analitik
9	Yuliani	Statistika Dasar	

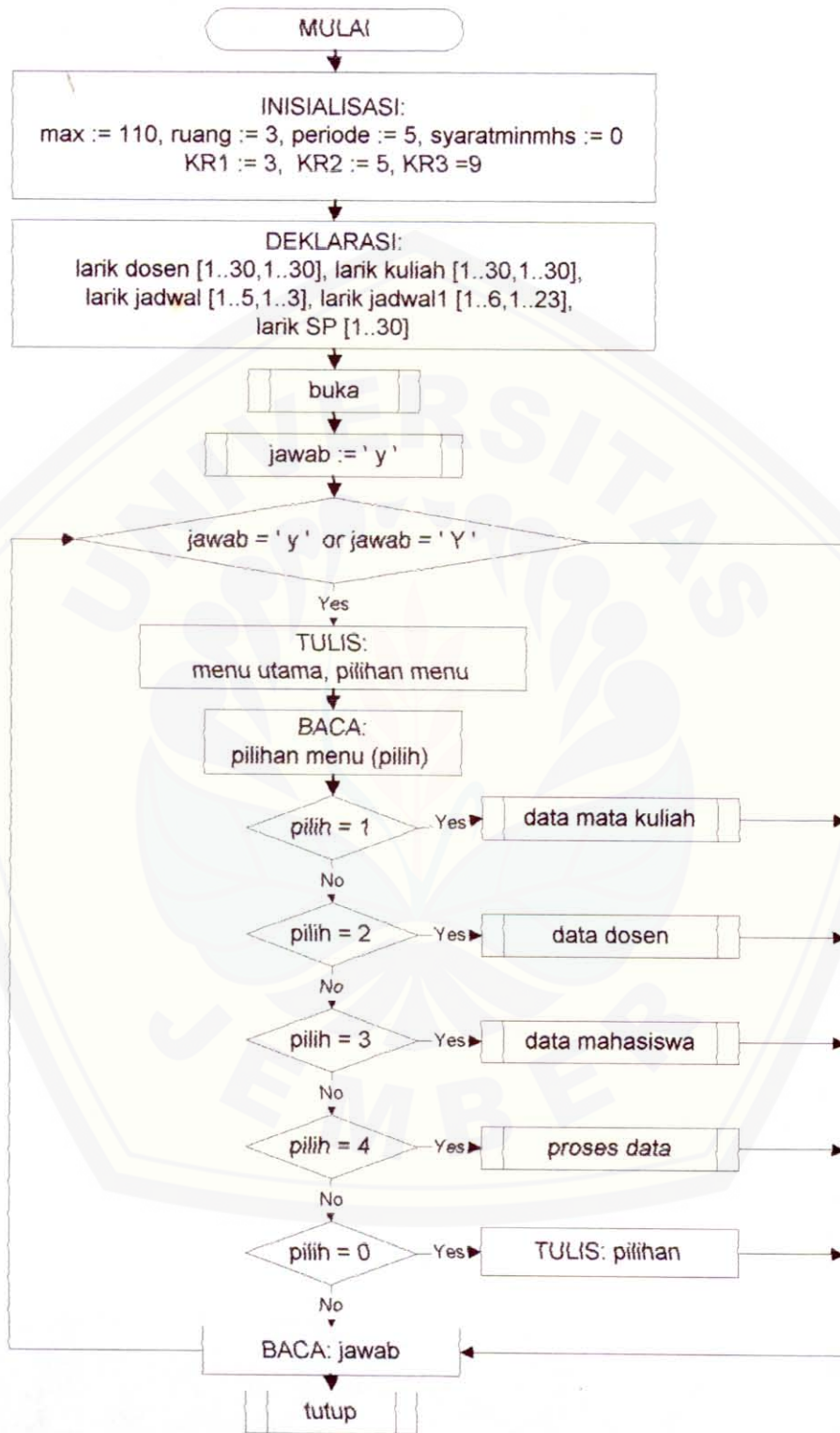
Data Ilustrasi Mahasiswa

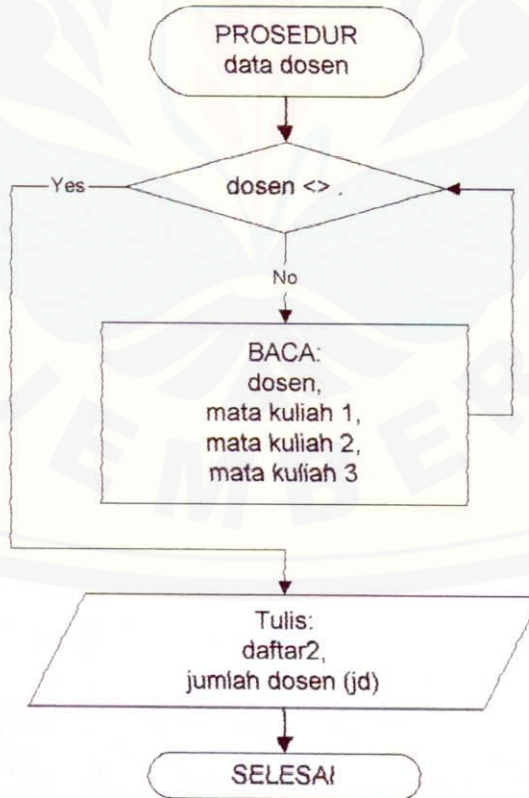
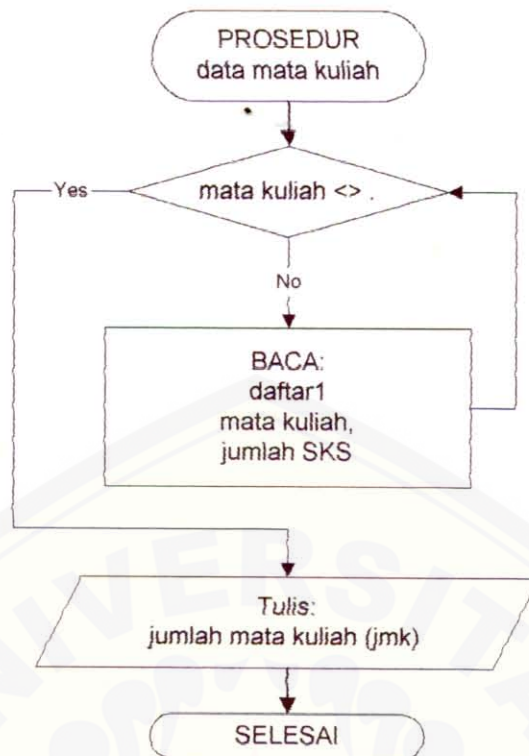
No.	Nama Mahasiswa	Mata Kuliah					
		GA	Geo	KI	K3	PDI	SD
1	A. Heru P.	1	1	0	0	0	0
2	A. Zaenal	0	1	1	0	0	0
3	Abdul Gofur	1	0	0	0	1	0
4	Abdul Kamil	0	0	0	1	1	1
5	Abdul Manan	0	0	0	1	1	1
6	Aka Widi	0	1	0	0	0	0
7	Alifah	0	1	0	0	0	0
8	Anang K.	0	0	0	0	0	1
9	Andamari	0	0	1	0	0	0
10	Anis Susiana	0	1	1	0	0	0
11	Arifa Rahmawati	0	0	0	0	1	1
12	Anjang R.	0	0	1	0	0	0
13	Anne Rufaidah	1	0	0	0	0	0
14	Ari Budi	0	0	1	0	0	0
15	Arif Elly	0	0	0	1	1	0
16	Aris Yuli	0	0	1	0	0	0
17	Atus Sasmono	0	0	1	0	0	0
18	Bahran Bahirian	0	1	0	0	0	0
19	Bahrud Muhiith	0	0	0	1	0	0
20	Camila Krisnawati	0	0	1	0	0	0
21	Dian Afrianti	0	1	1	0	0	0
22	Dian Retno	0	0	1	0	0	0
23	Dina Maria	0	0	0	1	1	0
24	Drita Trihapsari	1	0	0	0	0	0
25	Dwi Indrayanti	0	0	0	1	0	1
26	Dwi Sekarsari	0	0	0	1	1	1
27	Dyah AS.	0	0	1	0	0	0
28	Edi Santoso	0	0	1	0	0	0
29	Eli Aminingsih	0	0	0	1	0	1
30	Eli J.	1	0	0	1	0	0
31	Endang Arifah	0	1	0	0	0	1
32	Endang Sri	0	1	1	0	0	0
33	Endang Wahyuni	0	1	0	0	0	1
34	Eni Sunarmi	0	0	1	0	0	0

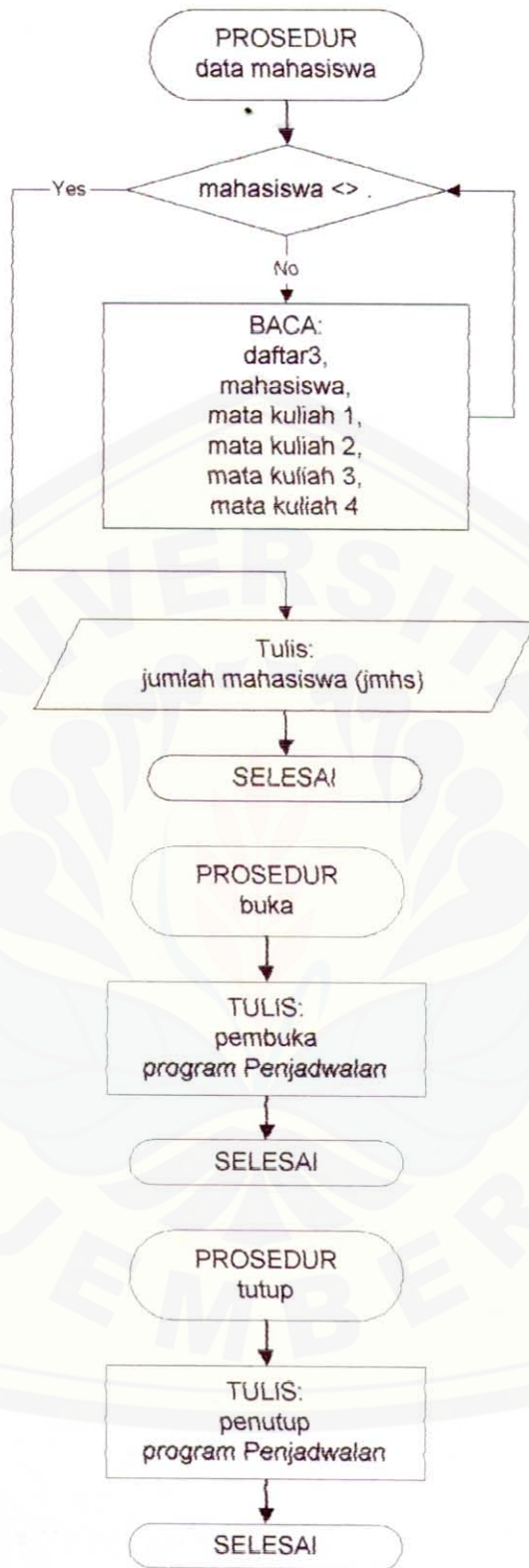
35	Erlina agustin	0	0	0	1	0	0
36	Erta Dwi	0	1	1	0	0	0
37	Fauziah Agustin	0	0	1	0	0	0
38	Febriana S.	0	0	1	0	0	0
39	Fitria Eka	0	1	0	0	0	0
40	Hendra Kurniawan	0	0	0	0	1	1
41	Hendra Kusuma	0	0	1	0	0	0
42	Herlina Yuliani	0	0	0	1	1	0
43	Iin Mardiaty	0	0	1	0	0	0
44	Ika Sunda	0	1	1	0	0	0
45	Ike	0	1	0	0	0	0
46	Ike Rahmania	0	0	1	0	0	0
47	Imam Wibowo	0	0	1	0	0	0
48	Imron Kurniawan	1	1	0	0	0	0
49	Inaf Batoran	0	0	1	0	0	0
50	Indah Yuliana	0	0	1	0	0	0
51	Indari setyowati	0	0	0	1	0	0
52	Indriana Santi	0	1	0	0	0	1
53	Irma Wijayanti	0	1	0	0	1	0
54	Isda Titis	0	0	1	0	0	0
55	Isnaini Hasan	0	0	1	0	0	0
56	Isnaini Maifaryanti	0	0	1	0	0	0
57	Jacobus Jangku	0	0	1	0	0	0
58	Jati Aninka	1	0	0	1	0	0
59	Jeni Parintak	0	0	1	0	0	0
60	Joko Muryanto	0	1	0	0	0	1
61	Lilis Widyawati	0	0	1	0	0	0
62	Lukman I.Y.P	1	0	0	1	0	0
63	M. Rizal	0	1	1	0	0	0
64	Mardiana	0	1	0	0	0	0
65	Maria Ulfa	0	1	0	0	1	0
66	Masita	1	0	0	0	0	0
67	Maya Wulandari	0	0	1	0	0	0
68	Mery Wahyuni	0	0	1	0	0	0
69	MH. Hidayatulah	0	0	1	0	0	0
70	Miftahul R.	0	1	0	0	0	1
71	Minartin	0	1	0	0	0	0
72	Minda Lestari	0	0	1	0	0	0
73	Nanik Sri	0	0	0	0	1	0

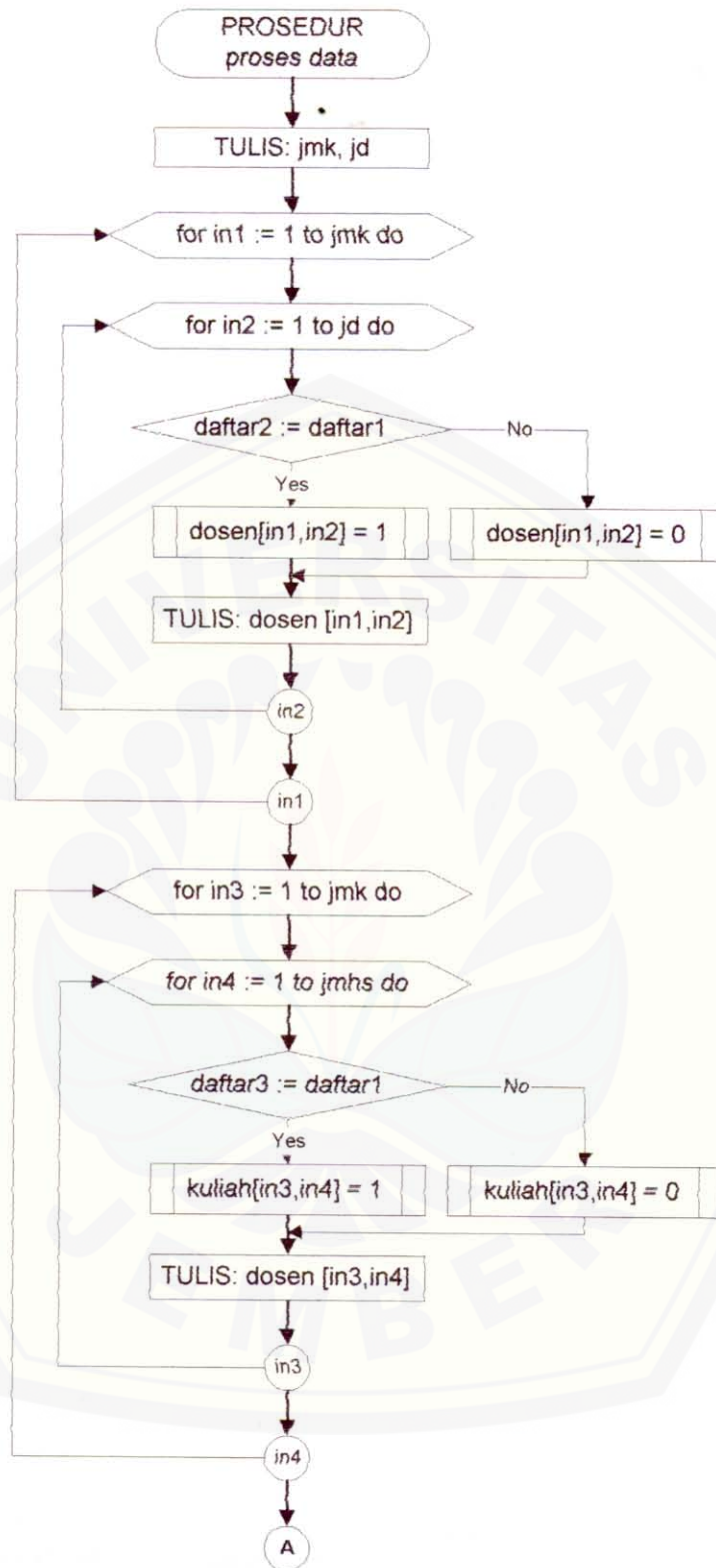
74	Nanung Nurhayati	0	0	0	0	1	0
75	Noviana Sapta	0	0	0	1	0	0
76	Nurul Hidayati	0	1	0	0	0	0
77	Palupi Meyda	0	1	0	0	0	0
78	Rahayu Indah	1	0	0	0	0	0
79	Ratu Silva	0	1	0	0	0	0
80	Riza Afifah	0	0	1	0	0	0
81	Riza MA.	0	1	1	0	0	0
82	Selfi Aprilia	0	0	1	0	0	0
83	Shofyan Imam	0	1	1	0	0	0
84	Silviana Widi	0	1	0	0	0	0
85	Siti Aminah	0	1	1	0	0	0
86	Siti Fauziyah	0	0	0	1	0	0
87	Siti Yulaihah	0	0	0	0	1	0
88	Solviana W.	0	0	0	0	0	1
89	Subhan Suryonoto	0	0	1	0	0	0
90	Sugiyatiningsih	0	1	0	0	0	1
91	Theresia Trias	0	1	0	0	0	0
92	Tri Bagus	0	0	0	0	0	1
93	Tri Prasetyowati	0	0	0	1	1	1
94	Tri Susilowati	0	1	0	0	1	0
95	Triana Arik	0	1	0	0	1	0
96	Tutut Triastuti	0	0	0	1	0	0
97	Urip Damayanti	0	1	0	0	0	0
98	Very Dwi	0	0	1	0	0	0
99	Vita Desy	0	1	0	0	0	0
100	Wahyu Eianingsih	0	0	0	0	1	0
101	Wahyudi H.	0	0	0	0	1	0
102	Wahyudi Hidayat	0	1	0	0	0	0
103	Wiwid	1	0	0	0	0	0
104	Yeni Amalia	1	0	0	0	0	0
105	Yenny Eka	1	1	0	0	1	0
106	Yetie Herawati	0	0	0	1	0	1
107	Yogie Dana	0	1	0	0	1	1
108	Yuli Fitria	0	0	1	0	0	0
109	Zuli Wulan	0	0	0	1	1	0
	Jumlah Mahasiswa	13	39	43	20	22	19

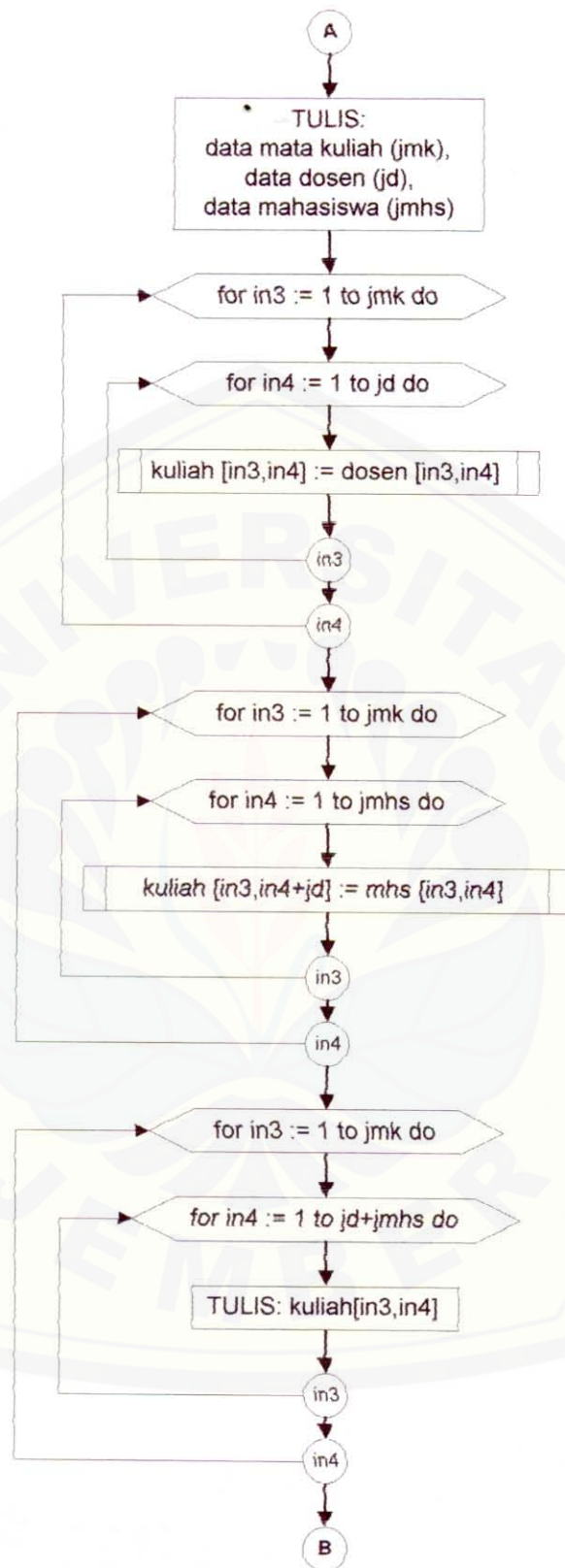
Lampiran 2 : Bagan Alir Program Penjadualan Semester Pendek dengan Algoritma Pewarnaan Titik Graf Menggunakan Turbo Pascal for Windows ver. 1.5.

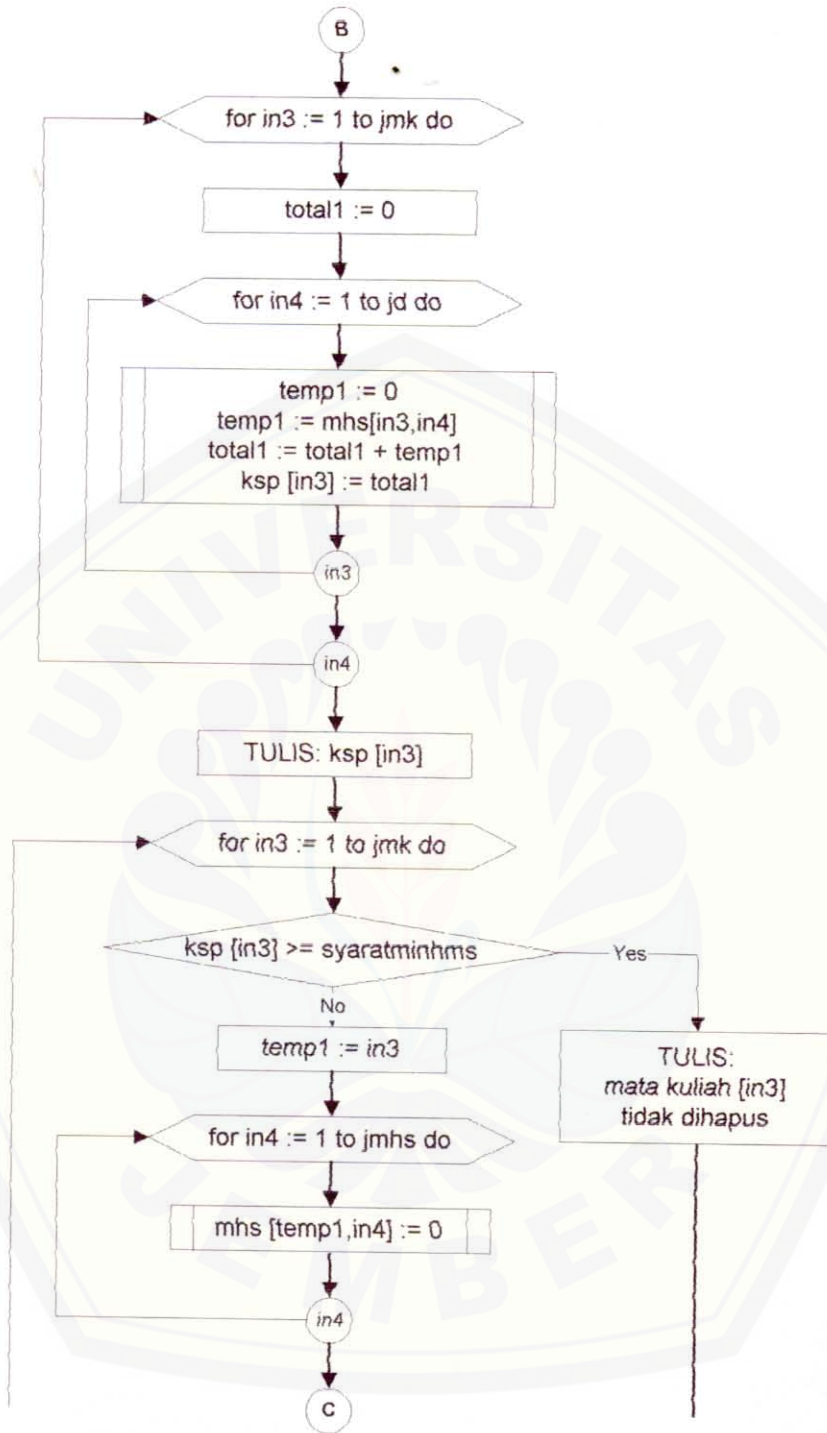


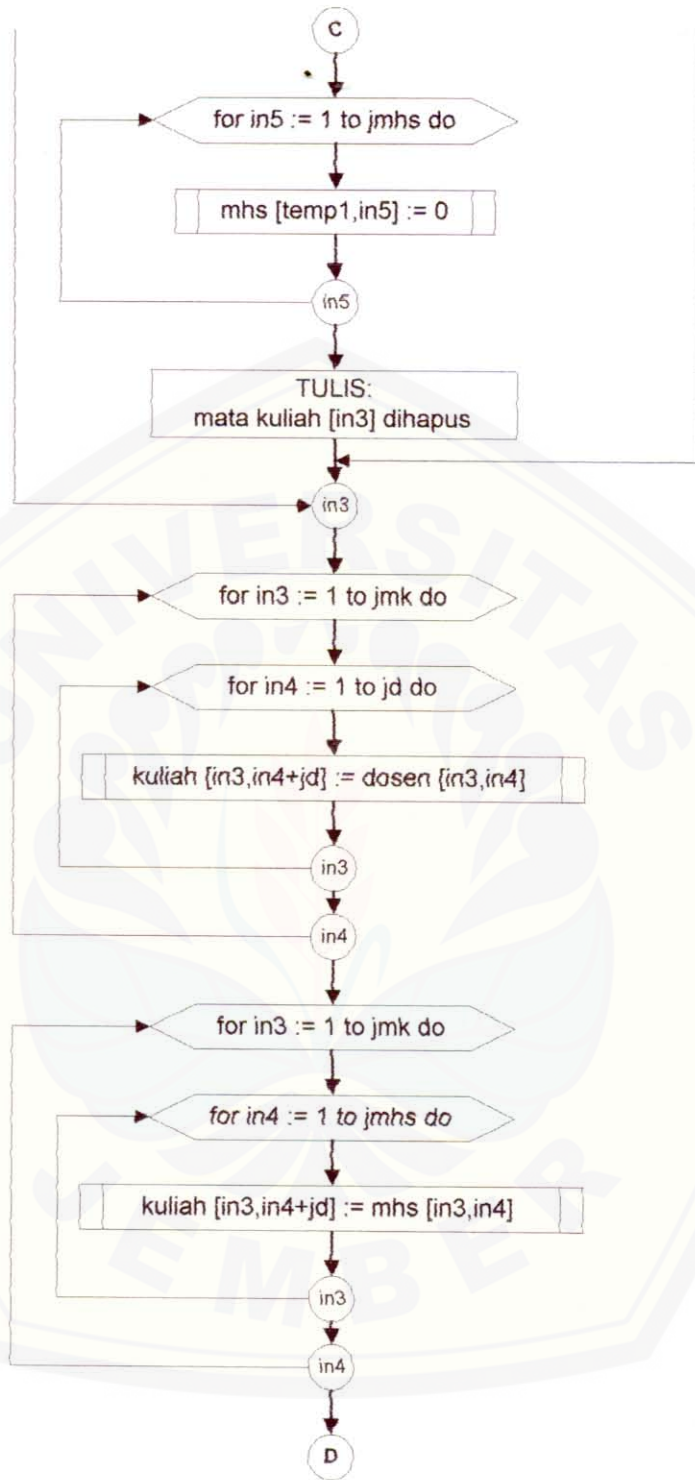


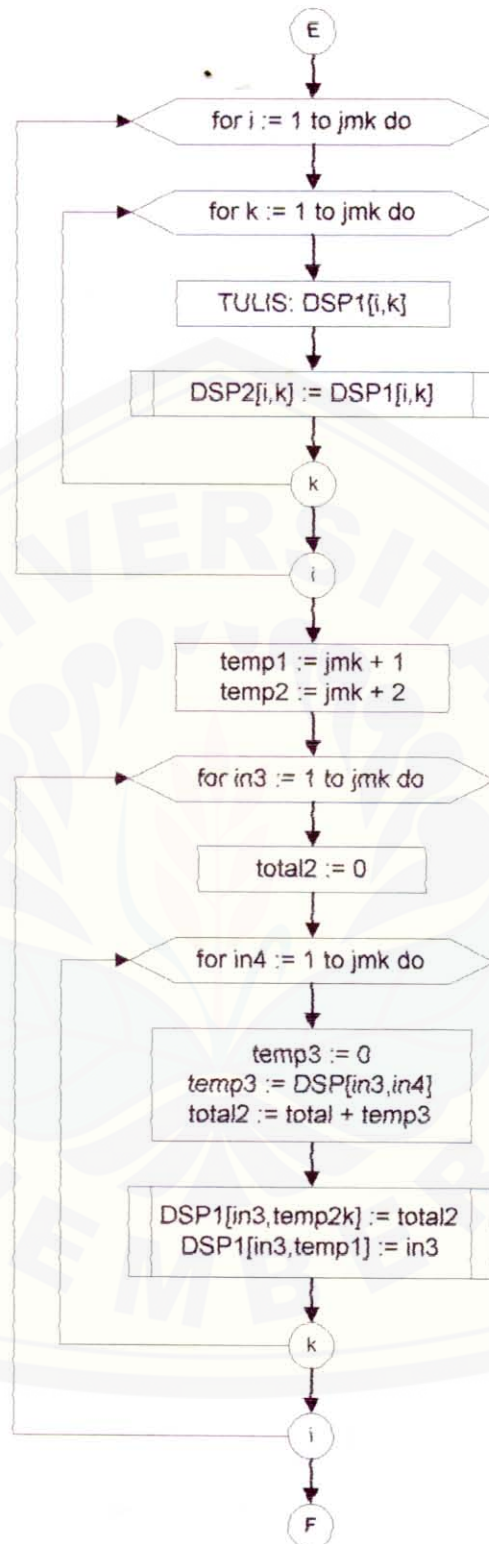


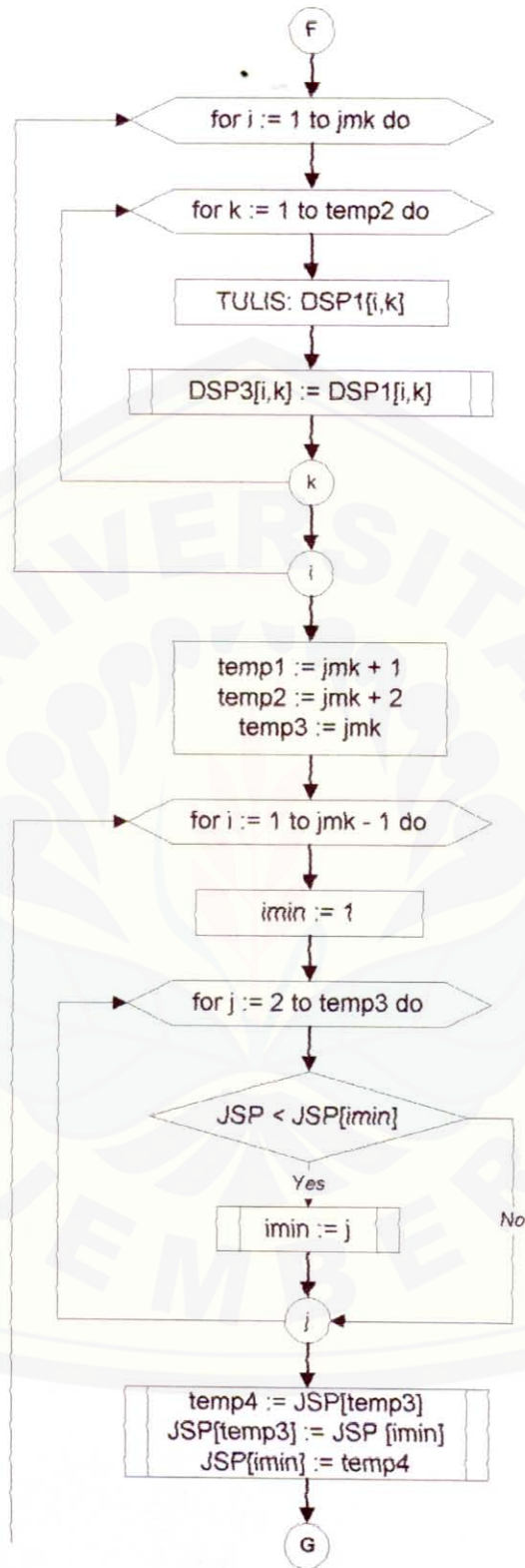


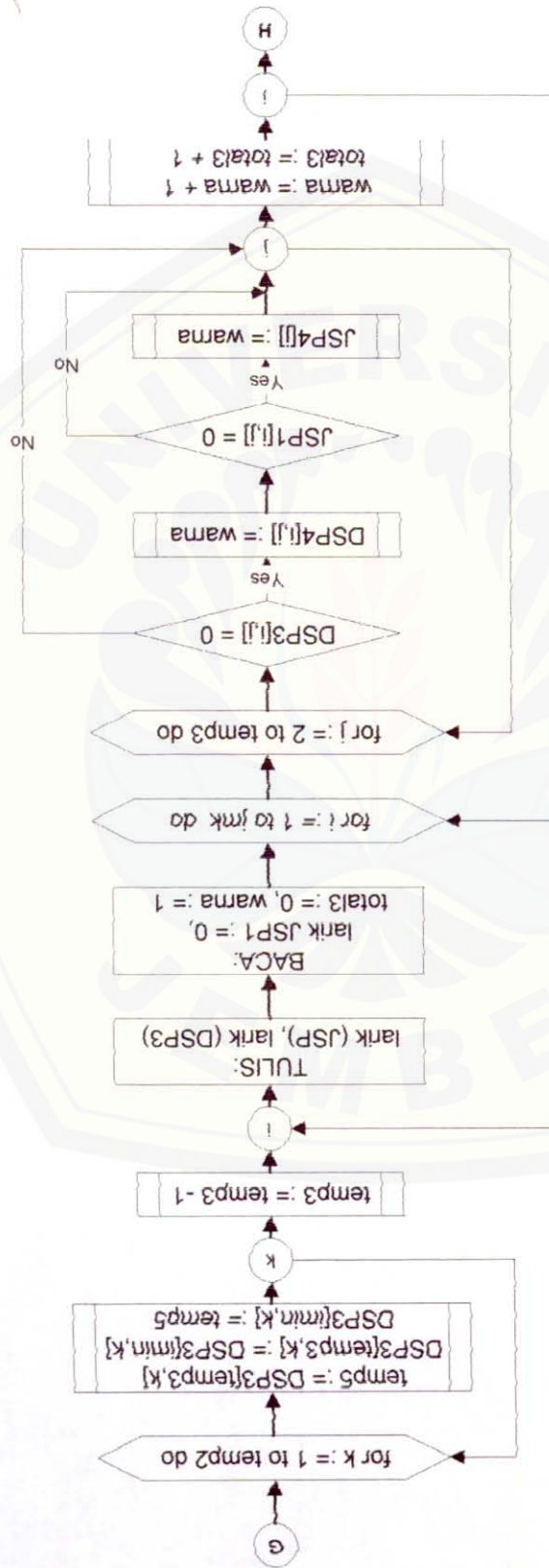


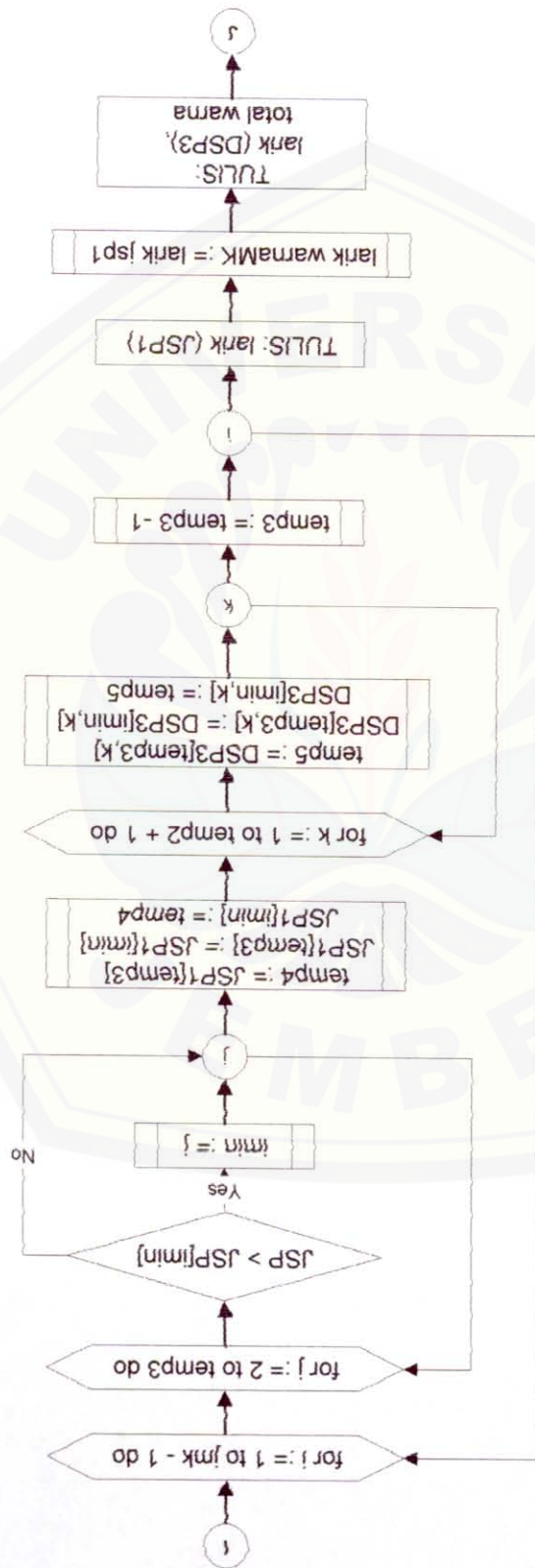


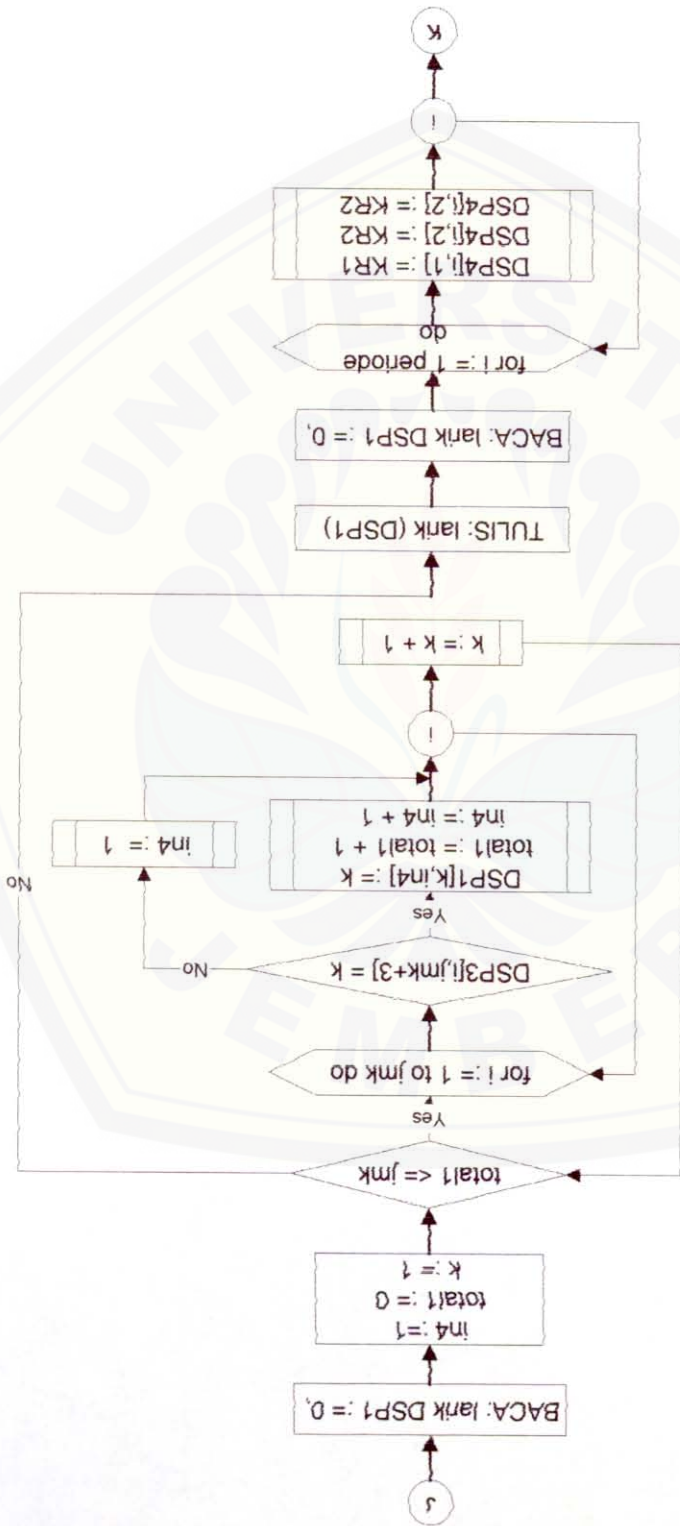


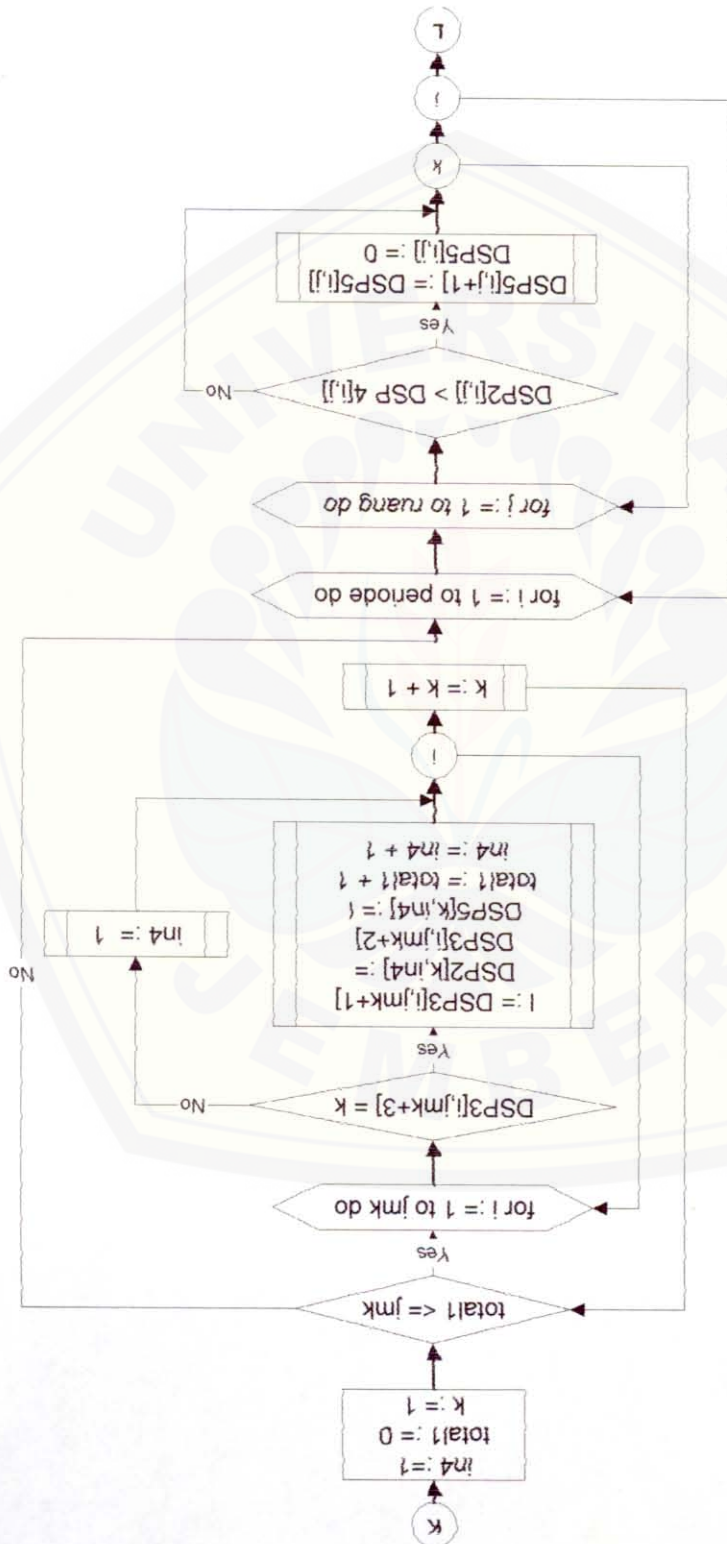


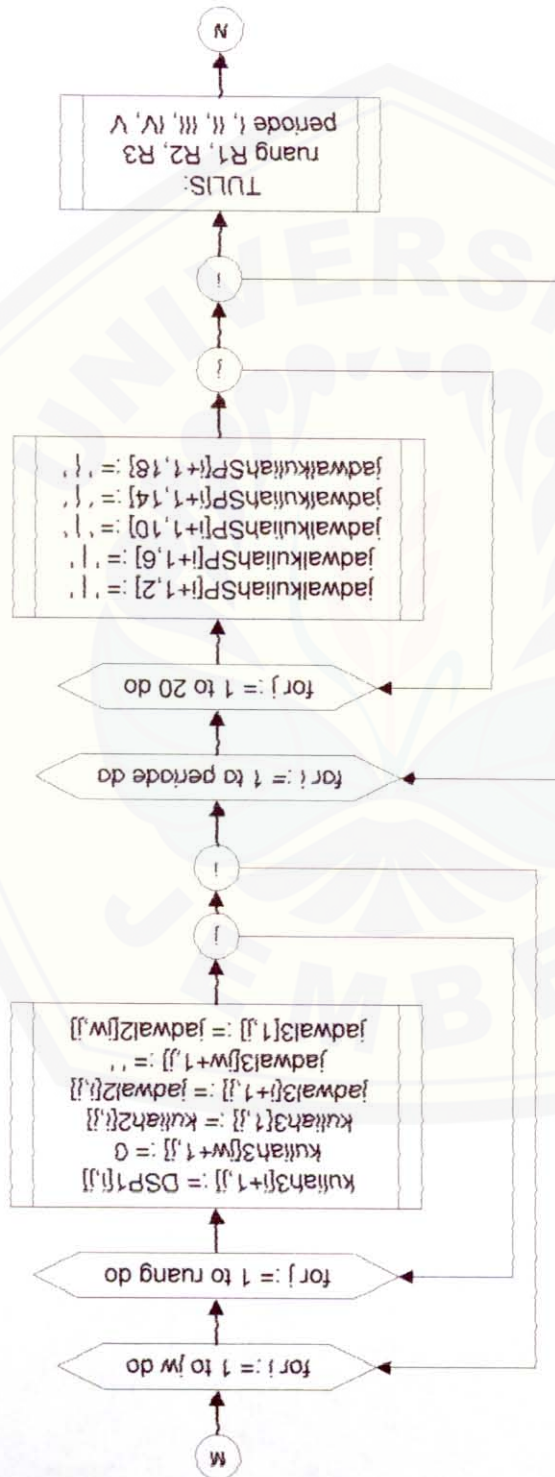


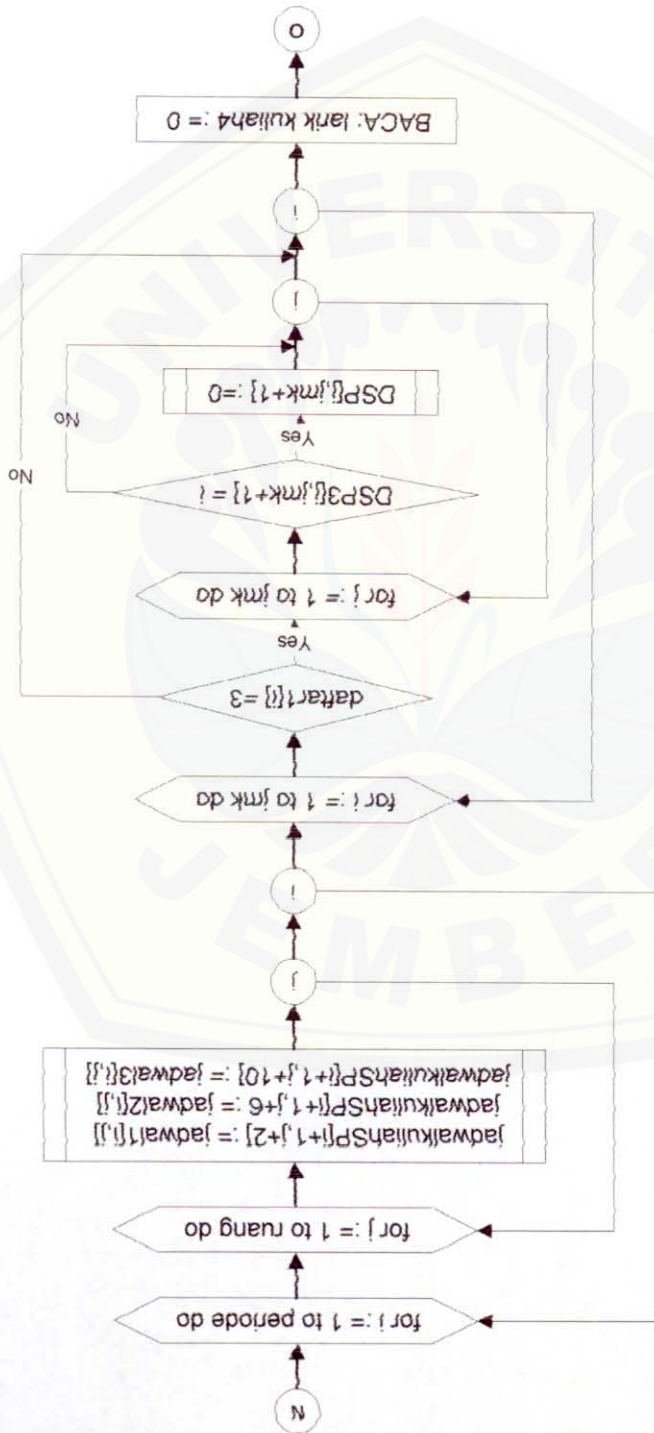


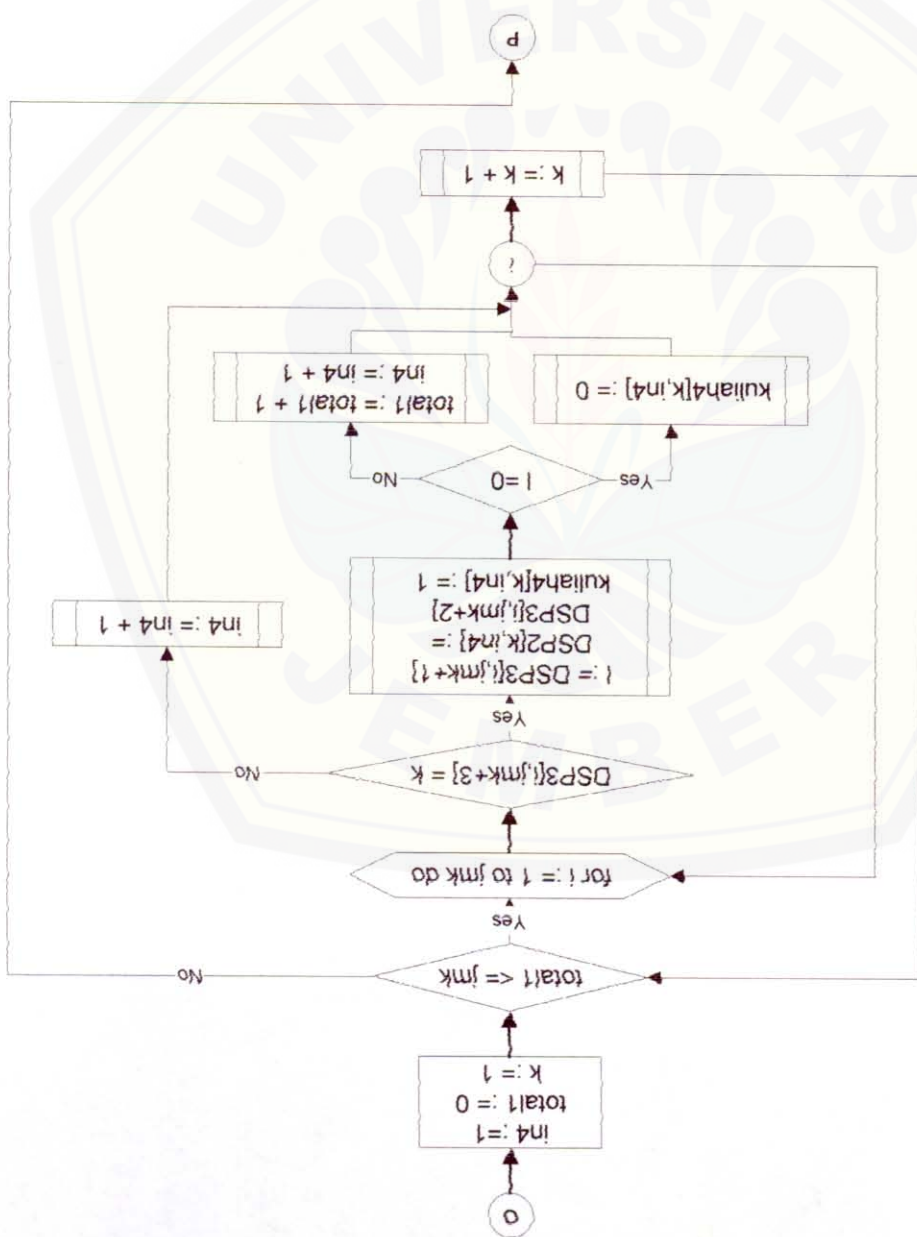


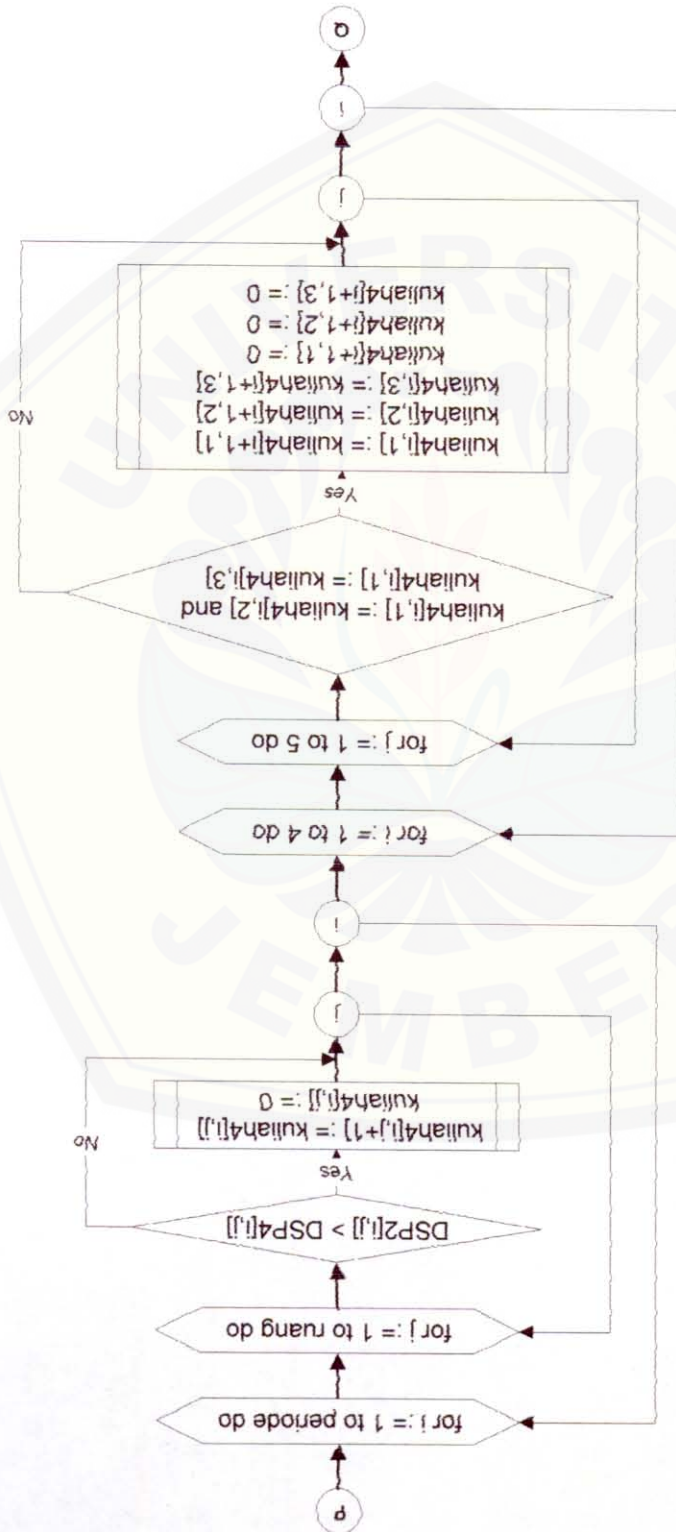


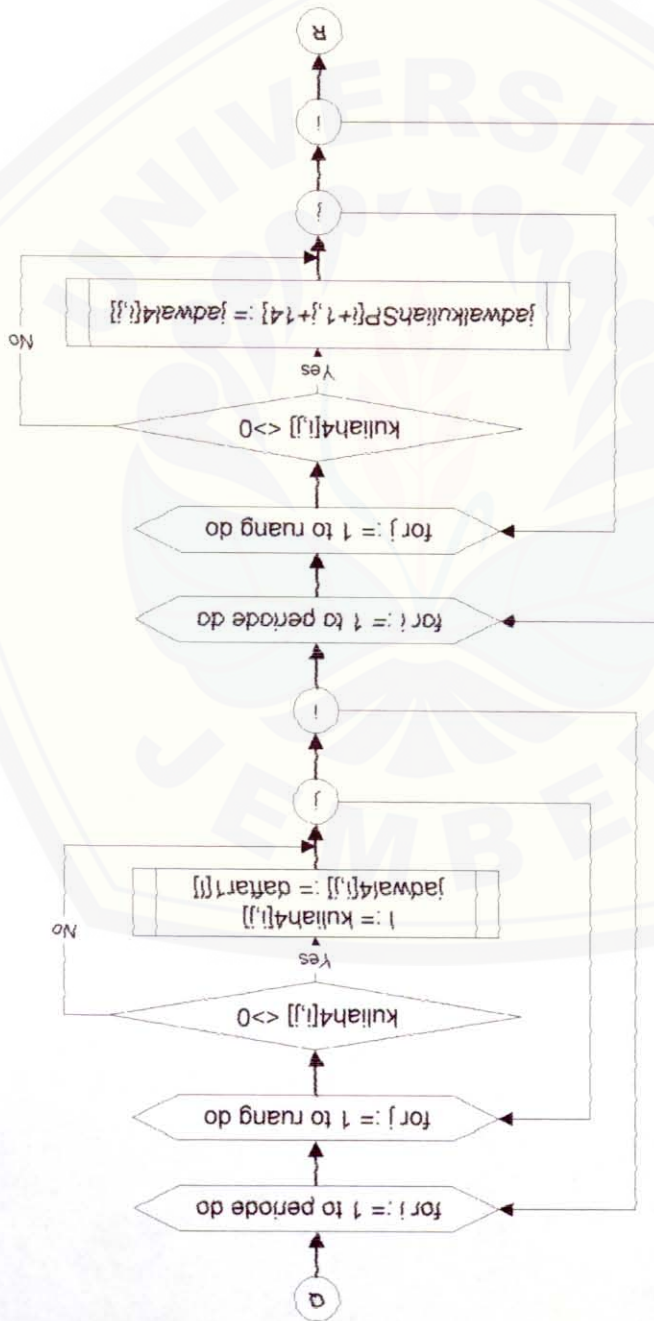


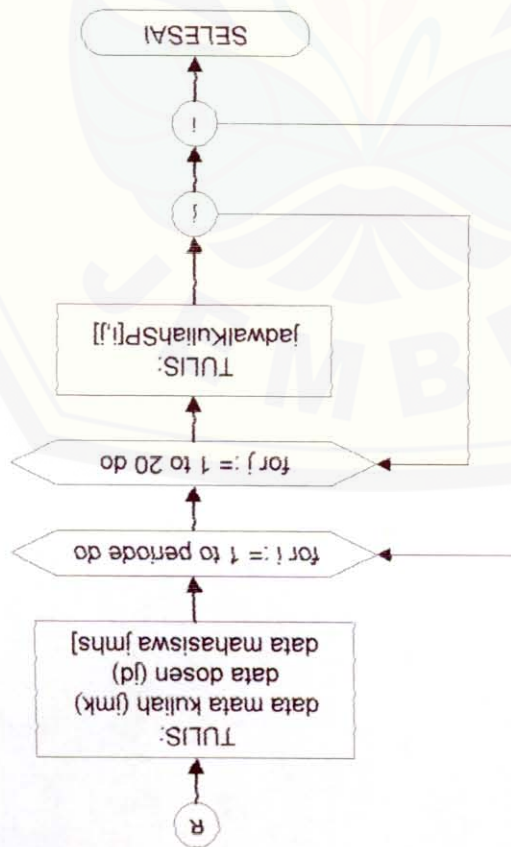












Lampiran 3 : List Program Penjadualan Semester Pendek dengan Algoritma Pewarnaan Titik Graf Menggunakan Turbo Pascal for Windows ver. 1.5.

```

Program PenjadualanSemesterPendek;

Uses
  WinCrt;

Const
  max = 110;
  ruang = 3;
  periode = 5;
  syaratminmhs = 0;

Type
  RekamData1 =
    record MataKuliah : string[10];
           Kode       : string[12];
           SKS        : integer;
    end;
  RekamData2 =
    record Dosen      : string[10];
           Kode       : string[12];
           MataKuliah1 : string[10];
           MataKuliah2 : string[10];
           MataKuliah3 : string[10];
    end;
  RekamData3 =
    record Nama       : string[10];
           NIM        : string[12];
           MataKuliah1 : string[10];
           MataKuliah2 : string[10];
           MataKuliah3 : string[10];
           MataKuliah4 : string[10];
    end;

  arrayDosen = Array[1..30,1..30] of integer;
  arrayKuliah = Array[1..30,1..30] of integer;
  arrayJadwal = array[1..5,1..3] of string[30];
  arrayJadwal1 = array[1..6,1..23] of string[10];
  arraySP = Array[1..30] of integer;

Var
  i, j, k, l, Pilih, imin,
  in1, in2, in3, in4, in5,
  Total1, Total2, Total3,
  warna, WarnaMK, JMK, JMHS, JD, JW,
  Temp1, Temp2, Temp3, Temp4, Temp5 : Integer;
  Jawab : Char;
  Daftar1 : array [1..20] of RekamData1;
  Daftar2 : array [1..20] of RekamData2;
  Daftar3 : array [1..max] of RekamData3;
  Dosen, DaftarSP, Mhs,
  Dosen1, Dosen2, Dosen3, Dosen4, Dosen5 : arrayDosen;
  Kuliah, DSP1, DSP2, DSP3, DSP4,
  Kuliah1, Kuliah2, Kuliah3, Kuliah4 : arrayKuliah;
  KuliahSP, KSP1, KSP2, JSP, JSP1, JSP0 : arraySP;
  jadwall1, Jadwal2, Jadwal3, Jadwal4 : arrayJadwal;
  JadwalKuliahSP : arrayJadwal1;

```

```

Procedure TulisFile1; {Mengisi Data Mata Kuliah}
Begin
  clrscr;
  writeln(' Menu : Mengisi Data Mata Kuliah ');
  writeln('-----');
  writeln;
  i := 1;
  writeln('Tulis Datanya Jika Selesai ketik titik(.) pada Mata Kuliah : ');
  writeln;
  write(i:3, '.Mata Kuliah : '); readln(Daftar1[i].MataKuliah);
  while Daftar1[i].MataKuliah <> '.' do
    begin
      write('':4, 'SKS           : '); readln(Daftar1[i].SKS);
      i := i + 1;
      writeln;
      write(i:3, '.Mata Kuliah : '); readln(Daftar1[i].MataKuliah);
    end;
  JMK := i-1;
  writeln;
  write('Jumlah Mata Kuliah adalah :', JMK);
  writeln;
End;

Procedure TulisFile2; {Mengisi Data Dosen}
Begin
  clrscr;
  writeln(' Menu : Mengisi Data Dosen ');
  writeln('-----');
  writeln;
  i := 1;
  writeln('Tulis Datanya Jika Selesai ketik titik(.) pada Dosen : ');
  writeln;
  write(i:3, '.Dosen : '); readln(Daftar3[i].MataKuliah1);
  while Daftar2[i].Dosen <> '.' do
    begin
      write('':4, 'Mata Kuliah-1   : '); readln(Daftar2[i].MataKuliah1);
      write('':4, 'Mata Kuliah-2   : '); readln(Daftar2[i].MataKuliah2);
      write('':4, 'Mata Kuliah-3   : '); readln(Daftar2[i].MataKuliah3);
      i := i + 1;
      writeln;
      write(i:3, '.Dosen : '); readln(Daftar2[i].Dosen);
    end;
  JD := i-1;
  writeln;
  write('Jumlah Dosen adalah :', JD);
  writeln;
End;

Procedure TulisFile3; {Mengisi Data Mahasiswa}
Begin
  clrscr;
  writeln(' Menu : Mengisi Data Mahasiswa ');
  writeln('-----');
  writeln;
  i := 1;
  writeln('Tulis Datanya Jika Selesai ketik titik(.) pada Nama Mahasiswa : ');
  writeln;
  write(i:3, '.Nama Mahasiswa : '); Readln(Daftar3[i].Nama);
  while Daftar3[i].Nama <> '.' do
    begin
      write('':4, 'Mata Kuliah 1   : '); readln(Daftar3[i].MataKuliah1);
      write('':4, 'Mata Kuliah 2   : '); readln(Daftar3[i].MataKuliah2);
      write('':4, 'Mata Kuliah 3   : '); readln(Daftar3[i].MataKuliah3);
    end;
  end;

```

```

        write('':4,'Mata Kuliah 4  : '); readln(Daftar3[i].MataKuliah4);
        i := i + 1;
        writeln;
        write(i:3,'.Nama Mahasiswa : '); readln(Daftar3[i].Nama);
    end;
    JMHS := i-1;
    writeln;
    write('Jumlah Mahasiswa adalah :', JMHS);
    writeln;
End;

Procedure ProsesData1;
Begin
    clrscr;
    writeln;
    writeln(' Menu : Proses Data Dosen ');
    writeln('-----');
    writeln;
    writeln('':3,'Data Mata Kuliah : ',JMK,' rekaman');
    writeln('':3,'Data Dosen      : ',JD,' rekaman');
    writeln;
    writeln(' Matrik Data Mata Kuliah dan Dosen ');
    writeln('-----');
    writeln;
    for in1 := 1 to JMK do
        begin
            for in2 := 1 to JD do
                begin
                    if Daftar2[in2].MataKuliah1 = Daftar1[in1].MataKuliah
                        then Dosen1[in1,in2] := 1
                        else Dosen1[in1,in2] := 0;
                    if Daftar2[in2].MataKuliah2 = Daftar1[in1].MataKuliah
                        then Dosen2[in1,in2] := 1
                        else Dosen2[in1,in2] := 0;
                    if Daftar2[in2].MataKuliah3 = Daftar1[in1].MataKuliah
                        then Dosen3[in1,in2] := 1
                        else Dosen3[in1,in2] := 0;
                    Dosen[in1,in2] := Dosen1[in1,in2]+Dosen2[in1,in2]+Dosen3[in1,in2];
                    write(Dosen[in1,in2]:3);
                end;
            writeln;
        end;
    writeln;
    readln;

    clrscr;
    writeln;
    writeln(' Menu : Proses Data Mahasiswa ');
    writeln('-----');
    writeln;
    writeln('':3,'Data Mata Kuliah : ',JMK,' rekaman');
    writeln('':3,'Data Mahasiswa   : ',JMHS,' rekaman');
    writeln;
    writeln(' Matrik Data Mata Kuliah dan Mahasiswa ');
    writeln('-----');
    writeln;
    for in3 := 1 to JMK do
        begin
            for in4 := 1 to JMHS do
                begin
                    if Daftar3[in4].MataKuliah1 = Daftar1[in3].MataKuliah
                        then Kuliah1[in3,in4] := 1
                        else Kuliah1[in3,in4] := 0;
                end;
            writeln;
        end;
    writeln;

```

```

    if Daftar3[in4].MataKuliah2 = Daftar1[in3].MataKuliah
    then Kuliah2[in3,in4] := 1
    else Kuliah2[in3,in4] := 0;
    if Daftar3[in4].MataKuliah3 = Daftar1[in3].MataKuliah
    then Kuliah3[in3,in4] := 1
    else Kuliah3[in3,in4] := 0;
    if Daftar3[in4].MataKuliah4 = Daftar1[in3].MataKuliah
    then Kuliah4[in3,in4] := 1
    else Kuliah4[in3,in4] := 0;
    Mhs[in3,in4] := Kuliah1[in3,in4]+Kuliah2[in3,in4]+
    Kuliah3[in3,in4]+Kuliah4[in3,in4];
    write(Mhs[in3,in4]:3);
  end;
  writeln;
end;
writeln;
readln;

clrscr;
writeln;
writeln(' Menu : Proses Data ');
writeln('-----');
writeln;
writeln('':3,'Data Mata Kuliah : ',JMK,' rekaman');
writeln('':3,'Data Dosen      : ',JD,' rekaman');
writeln('':3,'Data Mahasiswa   : ',JMHS,' rekaman');
writeln;

{Matriks Data Mata kuliah dan Dosen-Mahasiswa}
for in3 := 1 to JMK do
  begin
    for in4 := 1 to JD do
      begin
        Kuliah[in3,in4] := Dosen[in3,in4];
      end;
    end;
  for in3 := 1 to JMK do
    begin
      for in4 := 1 to JMHS do
        begin
          Kuliah[in3,in4+JD] := Mhs[in3,in4];
        end;
      end;
    end;

  writeln;
  writeln(' Matriks Data Mata Kuliah dan Dosen-Mahasiswa ');
  writeln('-----');
  writeln;
  for in3 := 1 to JMK do
    begin
      for in4 := 1 to JD+JMHS do
        begin
          write(Kuliah[in3,in4]:3);
        end;
      writeln;
    end;
  writeln;
  readln;

```



```

clrscr;
{Matriks Jumlah Data Peserta Mata Kuliah}
writeln;
writeln(' Matriks Jumlah Data Peserta Mata Kuliah ');
writeln('-----');
writeln;
  for in3 := 1 to JMK do
    begin
      Totall :=0;
      for in4 := 1 to JMHS do
        begin
          templ := 0;
          templ := Mhs[in3,in4];
          Totall := Totall + templ;
          KSP1[in3] := Totall;
        end;
      writeln('':3,KSP1[in3]);;
    end;
writeln;
readln;

{Mata Kuliah setelah persyaratan minimum Mahasiswa}
clrscr;
writeln;
writeln(' Mata Kuliah setelah persyaratan minimum Mahasiswa');
writeln('-----');
writeln;
for in3 := 1 to JMK do
  begin
    if KSP1[in3] >= syaratminmhs
      then begin in3 := in3; writeln(' Mata Kuliah ',in3,' tidak dihapus'); end
    else
      begin
        templ := in3;
        for in4 := 1 to JMHS do
          begin
            Mhs[templ,in4] := 0;
          end;
        for in5 := 1 to JD do
          begin
            Dosen[templ,in5] := 0;
          end;
        writeln('Mata Kuliah ',in3,' dihapus');
      end;
  end;
writeln;
readln;

{Matriks Data Mata Kuliah dan Dosen-Mahasiswa setelah syarat}
clrscr;
writeln;
writeln(' Matriks Data Mata Kuliah dan Dosen-Mahasiswa ');
writeln(' setelah persyaratan minimum Mahasiswa');
writeln('-----');
writeln;
for in3 := 1 to JMK do
  begin
    for in4 := 1 to JD do
      begin
        Kuliah[in3,in4] := Dosen[in3,in4];
      end;
  end;
end;

```

```

for in3 := 1 to JMK do
  begin
    for in4 := 1 to JMHS do
      begin
        Kuliah[in3,in4+JD] := Mhs[in3,in4];
      end;
    end;
  for in3 := 1 to JMK do
    begin
      for in4 := 1 to JD+JMHS do
        begin
          write(Kuliah[in3,in4]:3);
          DaftarSP[in3,in4] := Kuliah[in3,in4];
        end;
      writeln;
    end;
  writeln;
  readln;

  {Matrik adjacent Mata Kuliah}
  clrscr;
  writeln;
  writeln(' Matrik adjacent Mata Kuliah ');
  writeln('-----');
  writeln;
  for i := 1 to JD+JMHS do
    begin
      for j := 1 to JMK do
        begin
          for k := 1 to JMK do
            begin
              if {DaftarSP[j,i] = 1} and {DaftarSP[k,i]=1}
                then DSP1[j,k] := 1;
              if j = k then DSP1[j,k] := 0
            end;
          end;
        end;
      end;
    for i := 1 to JMK do
      begin
        for k := 1 to JMK do
          begin
            write(DSP1[i,k]:3);
            DSP2[i,k] := DSP1[i,k];
          end;
        writeln;
      end;
    writeln;
    readln;

  {Penjumlahan Matrik adjacent Mata Kuliah}
  clrscr;
  writeln;
  writeln(' Penjumlahan Matrik adjacent Mata Kuliah ');
  writeln('-----');
  writeln;
  temp1 := JMK + 1;
  temp2 := JMK + 2;
  for in3 := 1 to JMK do
    begin
      Total2 := 0;
      For in4 := 1 to JMK do
        Begin
          temp3 := 0;

```

```

        temp3 := DSP1[in3,in4];
        Total2 := Total2 + temp3;
        JSP[in3] := Total2;
        DSP1[in3,temp2] := Total2;
        DSP1[in3,temp1] := in3;
    end;
end;
for i := 1 to JMK do
begin
    for k := 1 to temp2 do
    begin
        write(DSP1[i,k]:3);
        DSP3[i,k] := DSP1[i,k];
    end;
    writeln;
end;
writeln;
readln;

{Proses Pengurutan derajat titik}
clrscr;
temp1 := JMK + 1;
temp2 := JMK + 2;
temp3 := JMK;
for i := 1 to JMK - 1 do
begin
    imin := 1;
    for j := 2 to temp3 do
    begin
        if JSP[j] < JSP[imin] then
        begin
            imin := j;
        end;
    end;
    temp4 := JSP[temp3];
    JSP[temp3] := JSP[imin];
    JSP[imin] := temp4;
    for k := 1 to temp2 do
    begin
        temp5 := DSP3[temp3,k];
        DSP3[temp3,k] := DSP3[imin,k];
        DSP3[imin,k] := temp5;
    end;
    temp3 := temp3 - 1;
end;

{Cetak Matrik jumlah}
writeln;
writeln(' Matrik Jumlah Titik : ');
writeln('-----');
writeln;
for i := 1 to JMK do
begin
    writeln('':3,JSP[i]);
end;
writeln;
readln;

```

```

{Cetak Matrik terurut dan jumlah}
clrscr;
writeln;
writeln(' Matrik Data dan Jumlah Titik Terurut : ');
writeln('-----');
writeln;
for i := 1 to JMK do
  begin
    for k := 1 to temp2 do
      begin
        Write(DSP3[i,k]:3);
        DSP2[i,k] := DSP3[i,k];
      end;
    writeln;
  end;
writeln;
readln;

{Proses Pewarnaan}
clrscr;
writeln;
for i := 1 to JMK do
  begin
    JSP1[i] := 0;
  end;
total3 := 0;
warna := 1;
for i := 1 to JMK do
  begin
    for j := 1 to JMK do
      begin
        if DSP3[i,j] = 0 then
          begin
            DSP4[i,j] := warna;
            if JSP1[j] = 0 then JSP1[j] := warna;
          end;
        end;
        warna := warna + 1;
        total3 := total3 + 1;
      end;
    end;
  end;

{cetak matrik daftar warna}
writeln(' Matrik Daftar Warna : ');
writeln('-----');
writeln;
for i := 1 to JMK do
  begin
    for j := 1 to JMK do
      begin
        write(DSP4[i,j]:3);
      end;
    writeln;
  end;
writeln;
readln;

{Cetak Matrik terurut dan jumlah}
clrscr;
writeln;
writeln(' Matrik Data Warna Mata Kuliah: ');
writeln('-----');
writeln;

```

```

for i := 1 to JMK do
  begin
    DSP3[i,temp2+1] := JSP1[i];
  end;
writeln;
for i := 1 to JMK do
  begin
    for k := 1 to temp2+1 do
      begin
        DSP3[i,temp1] := i;
        write(DSP3[i,k]:3);
      end;
      writeln;
    end;
  end;
writeln;
readln;

{Cetak matrik warna titik}
clrscr;
writeln;
writeln(' Matrik Warna Titik : ');
writeln('-----');
writeln;
for i := 1 to JMK do
  begin
    writeln('':5,JSP1[i]);
  end;
writeln;

{Proses Pengurutan warna}
temp1 := JMK + 1;
temp2 := JMK + 2;
temp3 := JMK;
for i := 1 to JMK-1 do
  begin
    imin := 1;
    for j := 2 to temp3 do
      Begin
        if JSP1[j] > JSP1[imin] then
          begin
            imin := j;
          end;
      end;
    end;
    temp4 := JSP1[temp3];
    JSP1[temp3] := JSP1[imin];
    JSP1[imin] := temp4;
    for k := 1 to temp2+1 do
      begin
        temp5 := DSP3[temp3,k];
        DSP3[temp3,k] := DSP3[imin,k];
        DSP3[imin,k] := temp5;
      end;
    end;
    temp3 := temp3-1;
  end;

{Cetak Matrik Warna Mata Kuliah}
clrscr;
writeln;
writeln(' Matrik urutan warna Mata Kuliah : ');
writeln('-----');
for i := 1 to JMK do
  begin

```

```
writeln('':3,JSP1[i]);
warnaMK := JSP1[JMK]
end;
writeln;
readln;

{Cetak Matrik Data Warna Mata Kuliah}
clrscr;
writeln;
writeln(' Matrik Data Warna Mata Kuliah : ');
writeln('-----');
for i := 1 to JMK do
begin
for k := 1 to temp2+1 do
begin
Write(DSP3[i,k]:3);
end;
writeln;
end;
writeln;
writeln;
writeln(' Total Warna : ',warnaMK);
writeln('-----');
readln;

{Proses PengaturanPeriode}
clrscr;
writeln;
for i := 1 to JMK do
begin
for k := 1 to JMK do
begin
DSP1[i,k]:=0;
end;
end;

{Pengaturan Periode-Ruang}
in4 := 1;
totall := 0;
k:= 1;
while totall <= JMK do
begin
for i := 1 to JMK do
begin
if {DSP3[i,JMK+3] = k}
then
begin
DSP1[k,in4] := k;
totall := totall + 1;
in4 := in4 + 1;
end
else
begin
in4 := 1;
end;
end;
k := k + 1;
end;
end;
```

```

{Jadwal 1 hari}
writeln;
writeln(' Jadwal 1 hari : ');
writeln('-----');
for i := 1 to periode do
  begin
    for j := 1 to ruang do
      begin
        write(DSP1[i,j]:3);
      end;
      writeln;
    end;
  writeln;
  readln;

{Jadwal Mata Kuliah 1 hari}
in4 := 1;
totall := 0;
k:= 1;
while totall <= JMK do
  begin
    for i := 1 to JMK do
      begin
        if (DSP3[i,JMK+3] = k)
          then
            begin
              l := DSP3[i,JMK+1];
              jadwall[k,in4] := Daftar1[l].MataKuliah;
              totall := totall + 1;
              in4 := in4 + 1;
            end
          else
            begin
              in4 := 1;
            end;
        end;
        k := k + 1;
      end;
    end;

  clrscr;
  writeln;
  writeln(' Jadwal Mata Kuliah 1 hari : ');
  writeln('-----');
  for i := 1 to periode do
    begin
      for j := 1 to ruang do
        begin
          write(Jadwall[i,j]:3);
        end;
        writeln;
      end;
    writeln;
    readln;

begin {Pengaturan Periode}{ untuk 3 SKS}
writeln;
JW := warnaMK;
writeln;
for i := 1 to JW do
  begin
    for j := 1 to ruang do
      begin

```

```

        Kuliaah2[i+1,j] := DSP1[i,j];
        Kuliaah2[JW+1,j] := 0;
        Kuliaah2[1,j] := DSP1[JW,j];
        Jadwal2[i+1,j] := Jadwal1[i,j];
        Jadwal2[JW+1,j] := '';
        Jadwal2[1,j] := Jadwal1[JW,j];
    end;
    writeln;
end;
writeln;
for i := 1 to JW do
    begin
        for j := 1 to ruang do
            begin
                Kuliaah3[i+1,j] := Kuliaah2[i,j];
                Kuliaah3[JW+1,j] := 0;
                Kuliaah3[1,j] := Kuliaah2[JW,j];
                Jadwal3[i+1,j] := Jadwal2[i,j];
                Jadwal3[JW+1,j] := '';
                Jadwal3[1,j] := Jadwal2[JW,j];
            end;
            writeln;
        end;
        writeln;
        for i := 1 to periode do
            begin
                for j := 1 to 20 do
                    begin
                        JadwalKuliaahSP[i+1,2] := '';
                        JadwalKuliaahSP[i+1,6] := '';
                        JadwalKuliaahSP[i+1,10] := '';
                        JadwalKuliaahSP[i+1,14] := '';
                        JadwalKuliaahSP[i+1,18] := '';
                    end;
                    writeln;
                end;
                for i := 0 to 3 do
                    begin
                        J := 4*i;
                        JadwalKuliaahSP[1,3+j] := 'R1';
                        JadwalKuliaahSP[1,4+j] := 'R2';
                        JadwalKuliaahSP[1,5+j] := 'R3';
                    end;
                end;
                writeln;
                JadwalKuliaahSP[2,1] := 'I';
                JadwalKuliaahSP[3,1] := 'II';
                JadwalKuliaahSP[4,1] := 'III';
                JadwalKuliaahSP[5,1] := 'IV';
                JadwalKuliaahSP[6,1] := 'V';
            end;
            writeln;
            for i := 1 to periode do
                begin
                    for j := 1 to ruang do
                        begin
                            JadwalKuliaahSP[i+1,j+2] := Jadwal1[i,j];
                            JadwalKuliaahSP[i+1,j+6] := Jadwal2[i,j];
                            JadwalKuliaahSP[i+1,j+10] := Jadwal3[i,j];
                        end;
                    end;
                    writeln;
                end;
            end;
        end;
    end;
end; {Pengaturan Periode}

```



```

Begin {Pengaturan Periode untuk SKS}
writeln;
for i := 1 to JMK do
  begin
    if Daftar1[i].SKS = 3 then
      begin
        {writeln('Mata Kuliah dengan 3 SKS adalah Mata Kuliah ',i);}
        for j := 1 to JMK do
          begin
            if (DSP3[j,JMK+1] = i)
              then
                begin
                  DSP3[j,JMK+1] := 0;
                end;
            end;
          end;
        end;
      end;
    writeln;
  end;
  {Jadwal Mata Kuliah 1 hari}
  for i := 1 to 5 do
    begin
      for j := 1 to 3 do
        begin
          kuliah4[i,j] := 0;
        end;
      end;
      writeln;
    end;
  end;
  in4 := 1;
  totall := 0;
  k:= 1;
  while totall <= JMK do
    begin
      for i := 1 to JMK do
        begin
          if (DSP3[i,JMK+3] = k)
            then
              begin
                l := DSP3[i,JMK+1];
                kuliah4[k,in4] := 1;
                if l = 0
                  then
                    begin
                      kuliah4[k,in4] := 0;
                    end;
                totall := totall + 1;
                in4 := in4 + 1;
              end
            else
              begin
                in4 := 1;
              end;
          end;
        end;
      end;
      k := k + 1;
    end;
  end;
  writeln;
  for j := 1 to 4 do
    begin
      for i := 1 to 5 do
        begin
          if (kuliah4[i,1] = kuliah4[i,2]) and (kuliah4[i,1] = kuliah4[i,3])
            then

```

```

        begin
            kuliah4[i,1] := kuliah4[i+1,1];
            kuliah4[i,2] := kuliah4[i+1,2];
            kuliah4[i,3] := kuliah4[i+1,3];
            kuliah4[i+1,1] := 0;
            kuliah4[i+1,2] := 0;
            kuliah4[i+1,3] := 0;
        end;
    end;
end;
writeln;
for i := 1 to periode do
    begin
        for j := 1 to ruang do
            begin
                if {Kuliah4[i,j] <> 0}
                then
                    begin
                        l := Kuliah4[i,j];
                        jadwal4[i,j] := Daftarl[l].MataKuliah;
                    end
                end;
            end;
        end;
        writeln;
    end;
    writeln;
    for i := 1 to periode do
        begin
            for j := 1 to ruang do
                begin
                    JadwalKuliahSP[i+1,j+14] := Jadwal4[i,j];
                end;
            end;
        end;
    end;
    writeln;
End; {Pengaturan Periode untuk SKS}

clrscr;
writeln;
writeln(' Proses Data Selesai ');
writeln('-----');
writeln;
writeln('':3,'Data Mata Kuliah : ',JMK,' rekaman');
writeln('':3,'Data Dosen      : ',JD,' rekaman');
writeln('':3,'Data Mahasiswa   : ',JMHS,' rekaman');
writeln;
writeln('-----');
writeln('Tekan Return <Enter> untuk melanjutkan');
readln;
clrscr;
writeln('-----');
writeln('          Jadwal Kuliah Semester Pendek ');
writeln('-----');
for i := 1 to periode+1 do
    begin
        for j := 1 to 20 do
            begin
                write(JadwalKuliahSP[i,j]:3);
            end;
        end;
        writeln;
    end;
end;
readln;
End;

```

```

Procedure Buka;
Begin
  clrscr;
  writeln;
  writeln(' Pembuka - Program Penjadwalan ');
  writeln('-----');
  writeln;
  writeln;
  writeln('-----');
  writeln('Tekan Return <Enter> untuk melanjutkan');
End;

Procedure Tutup;
Begin
  clrscr;
  writeln;
  writeln(' Penutup - Program Penjadwalan ');
  writeln('-----');
  writeln;
  writeln(':5, 'Terima Kasih, ');
  writeln(':5, 'telah mencoba program ini, kritik dan saran selalu kami
nantikan. ');
  writeln;
  writeln('-----');
End;

{Program Utama}
Begin
  clrScr;
  Buka;
  readln;
  Jawab := 'y';
  while (Jawab = 'y') or (Jawab = 'Y') or (Jawab = '.') do
  begin
    clrScr;
    writeln('Menu Utama :');
    writeln('-----');
    writeln;
    writeln('1. Mengisi Data Mata Kuliah');
    writeln('2. Mengisi Data Dosen');
    writeln('3. Mengisi Data Mahasiswa');
    writeln('4. Proses Data SP');
    writeln('0. Keluar Program');
    writeln;
    write('Pilih salah satu menu diatas : '); readln(Pilih);
    case Pilih of
      1: Begin TulisFile1 End;
      2: Begin TulisFile2 End;
      3: Begin TulisFile3 End;
      4: Begin ProsesData1 End;
      0: Begin
          clrScr;
          writeln('Anda Akan Keluar Program. ');
          end;
      else Write('Anda tidak memilih menu'); readln; Tutup;
    end;
    writeln;
    write('Kembali ke menu utama ? (y/t)'); readln(Jawab);
  end;
  Tutup;
End.

```

