

**MENENTUKAN PELABELAN GRACEFUL
PADA GRAF LINTASAN (P_n) DENGAN PANJANG n
MENGUNAKAN PROGRAM PHP DAN JAVASCRIPT**

SKRIPSI

Oleh :

**BAMBANG RIADI
NIM. 02510010**



**JURUSAN MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG
MALANG
2009**

**MENENTUKAN PELABELAN GRACEFUL
PADA GRAF LINTASAN (P_n) DENGAN PANJANG n
MENGUNAKAN PROGRAM PHP DAN JAVASCRIPT**

SKRIPSI

Diajukan Kepada:

**Fakultas Sains dan Teknologi
Universitas Islam Negeri (UIN)
Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Sains (S.Si)**

Oleh :

**BAMBANG RIADI
NIM. 02510010**

**JURUSAN MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG
MALANG
2009**

SURAT PERNYATAAN ORISINALITAS PENELITIAN

Saya yang bertanda tangan dibawah ini:

Nama : BAMBANG RIADI

NIM : 02510010

Fakultas/Jurusan : Sains dan Teknologi/Matematika

Judul Penelitian : Menentukan Pelabelan Graceful pada Graf Lintasan (P_n)
dengan Panjang n Menggunakan Program PHP dan
Javascript.

Menyatakan dengan sebenar-benarnya bahwa hasil penelitian yang saya tulis ini tidak terdapat unsure-unsur penjiplakan karya penelitian atau karya ilmiah yang pernah dilakukan atau dibuat oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata hasil penelitian ini terbukti terdapat unsur-unsur jiplakan, maka saya bersedia untuk mempertanggung jawabkan, serta diproses sesuai peraturan yang berlaku.

Malang, 20 Juli 2009
Yang Membuat Pernyataan.

BAMBANG RIADI
NIM. 02510010

**MENENTUKAN PELABELAN GRACEFUL
PADA GRAF LINTASAN (F_n) DENGAN PANJANG n
MENGUNAKAN PROGRAM PHP DAN JAVASCRIPT**

SKRIPSI

Oleh :

**BAMBANG RIADI
NIM. 02510010**

Telah disetujui oleh:
Dosen Pembimbing ,

Abdussakir, M. Pd
NIP. 150 327 247

Tanggal: 22 Juli 2009

Mengetahui,
Ketua Jurusan Matematika

Sri Harini, M. Si
NIP. 150 318 321

**MENENTUKAN PELABELAN GRACEFUL
PADA GRAF LINTASAN (P_n) DENGAN PANJANG n
MENGUNAKAN PROGRAM PHP DAN JAVASCRIPT**

SKRIPSI

Oleh :

**BAMBANG RIADI
NIM. 02510010**

**Telah Dipertahankan di Depan Dewan Penguji Skripsi dan
Dinyatakan Diterima Sebagai Salah Satu Persyaratan Untuk
Memperoleh Gelar Sarjana Sains (S.Si)**

**Tanggal:
28 Juli 2009**

Susunan Dewan Penguji:

Tanda Tangan

- | | | | |
|--------------------|--|---|---|
| 1. Penguji Utama : | <u>Usman Pagalay, M.Si</u>
NIP.150 327 240 | (|) |
| 2. Ketua : | <u>Evawati Alisah, M.Pd</u>
NIP.150 291 271 | (|) |
| 3. Sekretaris : | <u>Abdussakir, M.Pd</u>
NIP.150 327 247 | (|) |

**Mengetahui,
Ketua Jurusan Matematika**

Sri Harini, M. Si
NIP. 150 318 321

PERSEMBAHAN

*Karya tulis ini penulis persembahkan untuk:
kedua orang tuaku
saudara-saudaraku
serta
seluruh insan yang telah membantu penulis
yang selalu memberikan dukungan moril dan spirituil.*

MOTTO

“Barang siapa ingin meraih dunia maka harus dengan ilmu,
barang siapa ingin meraih akherat maka harus dengan ilmu,
barang siapa ingin meraih keduanya maka harus dengan ilmu”.
(Al-Hadits)

Sesungguhnya sesudah kesulitan ada kemudahan
(QS Al-Insyirah : 6).

Ilmu menunjukkan kebenaran akal,
maka barang siapa yang berakal, niscaya dia berilmu
(Sayyidina Ali bin Abi Tholib).

KATA PENGANTAR

Assalamu'alaikum Wr.Wb.

Puji syukur kehadiran Allah SWT, karena atas taufik dan hidayah-Nya penulisan skripsi yang berjudul **“Menentukan Pelabelan Graceful pada Graf Lintasan (P_n) dengan Panjang n Menggunakan Program PHP dan Javascript”** dapat diselesaikan.

Sholawat serta salam semoga tetap tercurahkan kepada Nabi Muhammad SAW, yang telah membimbing umat manusia dari zaman kebodohan menuju zaman yang terang benderang, yaitu agama Islam.

Dalam penyusunan skripsi ini, penulis tidak dapat menyelesaikan sendiri tanpa bantuan dari berbagai pihak, untuk itu penulis mengucapkan rasa hormat dan terima kasih yang sebesar-besarnya kepada:

1. Prof. Dr. Imam Suprayogo, selaku Rektor Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
2. Prof. Dr. Sutiman Bambang Sumitro, SU., D.Sc., selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
3. Sri Harini, M.Si, selaku Ketua Jurusan Matematika Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
4. Abdussakir, M.Pd, selaku dosen pembimbing yang senantiasa sabar memberi arahan dan bimbingan dalam penyusunan skripsi ini.

5. Seluruh dosen dan staf Fakultas Sains dan Teknologi yang telah memberikan ilmunya selama ini dan yang selalu membimbing dan memberi motivasi agar penulis dapat menyelesaikan studi dengan baik.
6. Bapak dan Ibu tercinta dan seluruh keluarga, yang selalu memberikan semangat dan motivasi baik moril maupun spirituil dalam mendidik dan membimbing penulis hingga penulis dapat menyelesaikan skripsi ini.
7. Teman-teman mahasiswa Matematika.
8. Semua pihak yang telah membantu penulis, yang tidak dapat disebutkan satu persatu.

Penulis berdo'a semoga bantuan yang telah diberikan dicatat sebagai amal baik oleh Allah SWT dan mendapatkan balasan yang setimpal. Penulis berharap, semoga skripsi ini bermanfaat. Amin.

Malang, Juli 2009

Penulis

DAFTAR ISI

HALAMAN JUDUL

HALAMAN PERNYATAAN ORISINALITAS TULISAN

HALAMAN PERSETUJUAN

HALAMAN PENGESAHAN

HALAMAN PERSEMBAHAN

HALAMAN MOTTO

KATA PENGANTARi

DAFTAR ISIiii

DAFTAR GAMBARvi

DAFTAR LAMPIRANviii

ABSTRAKix

BAB I : PENDAHULUAN

1.1 Latar Belakang1

1.2 Rumusan Masalah3

1.3 Tujuan Penelitian3

1.4 Manfaat Penelitian3

1.5 Sistematika Penulisan4

BAB II : KAJIAN PUSTAKA**2.1 Graf**

2.1.1 Definisi Graf	6
2.1.2 Adjacent dan Incident	8
2.1.3 Derajat Titik	10
2.1.4 Graf Beraturan-r	12
2.1.5 Graf Komplit (<i>Complete Graph</i>)	13
2.1.6 Graf Bipartisi (<i>Bipartite Graph</i>)	13
2.1.7 Graf Bipartisi Komplit (<i>Complete Bipartite Graph</i>).....	14

2.2 Graf Terhubung	16
--------------------------	----

2.3 Graf Lintasan	18
-------------------------	----

2.4 Pelabelan Graf	19
--------------------------	----

2.5 Pelabelan Graceful	20
------------------------------	----

2.6 <i>Lexicographic Ordering</i>	21
---	----

2.7 Bahasa Pemrograman

2.7.1 Bahasa Pemrograman PHP	23
------------------------------------	----

2.7.2 Bahasa Pemrograman Javascript.....	24
--	----

2.7.3 Pemrograman HTML dan CSS	25
--------------------------------------	----

2.8 Langkah-langkah dalam Pemrograman Komputer	26
--	----

2.9 Dasar-dasar Pemrograman.....	28
----------------------------------	----

2.10 Keunggulan Aplikasi Menggunakan PHP dan Javascript	30
---	----

BAB III: METODE PENELITIAN

3.1 Waktu dan Tempat	32
----------------------------	----

3.2 Jenis Penelitian.....	32
---------------------------	----

3.3 Tahapan Penelitian	32
------------------------------	----

3.4 Alat Penelitian.....	34
--------------------------	----

BAB IV: PEMBAHASAN

4.1 Diagram Alir	35
------------------------	----

4.1.1 Diagram Alir Program secara Umum	37
--	----

4.1.2 Diagram Alir Metode Lexicographic Order.....	38
--	----

4.1.3 Diagram Alir Metode Random.....	39
4.1.4 Diagram Alir Proses Manual Input.....	40
4.2 Kode Program	41
4.3 Output Program.....	61
4.4 Pengujian Program	66
BAB V : PENUTUP	
5.1 Kesimpulan	73
5.2 Saran-saran.....	75

DAFTAR PUSTAKA

LAMPIRAN



DAFTAR GAMBAR

No.	Gambar	Halaman
2.1	Graf dan Multigraf	7
2.2	G_1 Graf Trivial dan G_2 Graf Non Trivial.....	8
2.3	Graf G	8
2.4	Graf G dengan 5 Titik dan 4 Sisi	9
2.5	Graf dengan Derajat Titik.....	10
2.6	Graf G_1 beraturan-1 dan G_2 beraturan-2.....	12
2.7	Graf Komplit.....	13
2.8	Graf Bipartisi	14
2.9	Graf Bipartisi Komplit	15
2.10	Graf dengan Jalan	16
2.11	Trail, Lintasan, Sirkuit dan Sikel	17
2.12	Graf Terhubung (<i>connected</i>).....	18
2.13	Graf Lintasan P_2, P_3 , dan P_n	19
2.14	Contoh Pelabelan Graceful pada Graf $K_{1,5}$	20
2.15	Contoh Graf Lintasan P_n dengan n Titik.....	21
4.1	Diagram Alir Program secara Umum	37
4.2	Diagram Alir Metode <i>Lexicographic Order</i>	38
4.3	Diagram Alir Metode Random	39
4.4	Diagram Alir Proses Manual Input	40
4.5	Output Program : Halaman Utama	61
4.6	Blok Define	62
4.7	Blok Option	62
4.8	Blok Proses	63
4.9	Blok Actions	63
4.10	Manual Textbox	64
4.11	Form Hasil	65

4.12 Menu Utama	67
4.13 Output Graceful	70
4.14 Menu Option.....	70
4.15 Output Graceful dan not Graceful.....	70
4.16 Output program Pelabelen Graceful pada Graf Lintasan P_n	71
4.17 Output Program Pelabelan Graf Lintasan P_{10} (Graceful)	72
4.18 Output Program Pelabelan Graf Lintasan P_{10} (Tidak Graceful)	72



DAFTAR LAMPIRAN

Lampiran 1 : kode program (Source Code)



ABSTRAK

Riadi, Bambang. 2009. *Menentukan Pelabelan Graceful Pada Graf Lintasan (P_n) Dengan Panjang n Menggunakan Program PHP dan Javascript*. Jurusan Matematika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
Pembimbing: Abdussakir, M. Pd.

Kata kunci: Pelabelan graceful, Graf lintasan (P_n) , Program PHP dan Javascript.

Pelabelan pada suatu graf adalah sebarang pemetaan (fungsi) yang memasangkan unsur-unsur graf (titik atau sisi) dengan bilangan (biasanya bilangan bulat). Jika domain dari fungsi adalah titik, maka pelabelan disebut pelabelan titik (*vertex labeling*). Jika domainnya adalah sisi, maka disebut pelabelan sisi (*edge labeling*), dan jika domainnya titik dan sisi, maka disebut pelabelan total (*total labeling*). Pelabelan graceful pada graf G dengan q sisi adalah fungsi injektif λ dari $V(G)$ ke $\{0, 1, 2, \dots, q\}$ sedemikian hingga, seandainya sisi (x, y) dilabeli dengan $|\lambda(x) - \lambda(y)|$, maka label sisi akan berbeda.

Pada penelitian ini akan dibahas tentang pelabelan graceful pada graf lintasan (P_n) dengan panjang n menggunakan program komputer. Adapun program yang digunakan adalah PHP dan Javascript.

Penelitian ini menghasilkan diagram alir, kode program serta output berupa graf lintasan P_n dengan n titik yang bersifat graceful atau yang tidak graceful, dimana setiap n yang diinput mempunyai jumlah iterasi sebanyak n faktorial, Nilai masing-masing iterasi didapat dengan metode lexicographic order dan metode random serta input manual.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Matematika berasal dari bahasa Yunani yaitu *mathema* yang diartikan sebagai sains, ilmu pengetahuan, atau belajar, dan juga *mathematikos* adalah suka belajar. Matematika dalam bahasa Belanda disebut *Wiskunde* atau ilmu pasti, yang kesemuanya berkaitan dengan penalaran. Sebagian besar masyarakat menganggap matematika hanya merupakan ilmu menghitung bilangan-bilangan dengan menggunakan beberapa operasi dasar; tambah, kurang, kali dan bagi. Seiring perkembangan zaman, ilmu matematika berkembang dan hadir sebagai hal yang mendasar dan perlu dipelajari pada setiap disiplin keilmuan (Wikipedia Indonesia, 2007:13)

Matematika sebagai disiplin ilmu dikenal sebagai *Queen of Science*, karena dalam konsep matematika banyak digunakan simbol yang mengosongkan arti yang juga bisa dipakai dan diterapkan di berbagai bidang keilmuan yang lain. Sehingga matematika dapat diterapkan kapanpun, dimanapun dan terbukti telah memberikan pengaruh yang cukup besar serta mempunyai peranan penting terhadap kemajuan disiplin ilmu lainnya, di antaranya ilmu statistika, perbankan, telekomunikasi dan lain sebagainya.

Gafur (2008) mengatakan bahwa ilmu pengetahuan dan teknologi tidak lepas dari peran serta ilmu matematika. Aplikasi ilmu matematika sangat banyak sekali dalam ilmu pengetahuan lain, salah satunya adalah teori graf. *Teori graf* adalah salah satu cabang matematika diskret yang penting dan banyak manfaatnya, antara lain dalam komunikasi, transportasi, sistem antrian, penjadwalan dan lain sebagainya.

Graf adalah himpunan yang tidak kosong yang memuat elemen-elemen yang disebut titik, dan suatu daftar pasangan tidak terurut elemen itu yang disebut sisi. Dalam teori graf ada yang disebut pelabelan graf, yaitu pemberian nilai pada titik, sisi atau titik dan sisi. Pelabelan graf sudah banyak dikaji mulai tahun 60-an. seperti valuasi- β yang diperkenalkan oleh Rosa pada tahun 1967 [5]. Sejak saat itu, sekitar 250 tulisan mengenai pelabelan banyak bermunculan.

Suatu graf dikatakan graceful jika terdapat fungsi injektif dari himpunan titik ke himpunan bilangan bulat tak negatif $\{0, 1, 2, \dots, q\}$ sedemikian hingga jika sisinya mendapat label harga mutlak dari selisih pelabelan kedua titik yang terhubung langsung (*adjacent*) maka hasilnya berbeda. Bilangan q adalah banyak sisi pada graf.

Graf lintasan P_n dengan panjang lintasan n titik, adakalanya merupakan graf graceful atau graf tidak graceful. Untuk menentukannya secara manual, hal itu dapat dilakukan dengan cara mencoba-coba, membalik-balik angka pada suatu titik sampai didapatkan suatu pelabelan graceful. Cara tersebut memakan waktu cukup lama dan pemikiran yang tidak mudah.

Pada zaman modern ini sudah ada komputer yang salah satu fungsinya adalah sebagai alat untuk memudahkan proses menghitung, menganalisis data, mengolah data, dan sebagainya. Pada intinya, fungsi komputer adalah memudahkan dan mempercepat pekerjaan. Dalam hal ini, ada beberapa bahasa pemrograman yang dapat digunakan untuk melakukan perintah pada sistem komputer tersebut, tergantung kemampuan, bahasa apa yang dikuasai serta cocok untuk diterapkan dalam menyelesaikan suatu persoalan menggunakan komputer.

Penulis bermaksud melakukan penelitian tentang penentuan pelabelan graceful khususnya pada graf lintasan (P_n) dengan panjang n menggunakan program komputer. Oleh karena itu, penulis merumuskan judul skripsi: **“Menentukan Pelabelan Graceful pada Graf Lintasan (P_n) dengan Panjang n Menggunakan Program PHP dan Javascript”**

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah yang dapat dikemukakan adalah bagaimana cara menentukan pelabelan graceful suatu graf lintasan (P_n) dengan panjang n menggunakan program PHP dan Javascript?

1.3 Tujuan Penulisan

Berdasarkan rumusan masalah, maka tujuan penulisan ini adalah menentukan pelabelan graceful suatu graf lintasan (P_n) dengan panjang n menggunakan program PHP dan Javascript.

1.4 Manfaat Penulisan

Skripsi ini diharapkan dapat bermanfaat bagi berbagai pihak, di antaranya:

a. Bagi Penulis

Dalam penulisan skripsi ini, penulis diharapkan dapat menemukan algoritma pemrograman untuk menentukan pelabelan graceful suatu graf lintasan (P_n) dengan panjang n menggunakan program PHP dan Javascript.

b. Bagi Pembaca

Diharapkan dapat menambah wawasan pengetahuan tentang graf graceful serta dapat mengoperasikan dan mengaplikasikan program pelabelan graceful yang telah dibuat oleh penulis untuk suatu graf lintasan (P_n) dengan n titik. Kemudian, diharapkan hasil penulisan ini dapat merangsang peneliti lain untuk melakukan pemrograman lebih lanjut mengenai pelabelan graceful pada beberapa jenis graf lainnya.

c. Bagi Lembaga

Bagi lembaga, penulisan skripsi ini dapat bermanfaat sebagai tambahan perbendaharaan karya tulis ilmiah.

1.5 Sistematika Penulisan

Untuk mempermudah penulis sekaligus pembaca dalam mengkaji skripsi ini, maka sistematika penulisannya dibagi menjadi empat bagian yaitu:

BAB I: PENDAHULUAN

Pada bab ini dijelaskan tentang latar belakang, rumusan masalah, tujuan penulisan, manfaat penulisan, dan sistematika penulisan.

BAB II: KAJIAN PUSTAKA

Pada bab ini dijelaskan tentang definisi graf, *adjacent* / *incident*, derajat dalam suatu graf, graf beraturan, graf komplit, graf bipartisi, graf bipartisi

komplit, graf terhubung, graf lintasan, pelabelan graf, pelabelan graceful
lexicographic order, serta bahasa pemrograman.



BAB III: METODE PENELITIAN

Meliputi waktu dan tempat penelitian, jenis penelitian, tahapan penelitian, dan alat penelitian.

BAB IV: PEMBAHASAN

Pada bab ini, dijelaskan tentang pembahasan mengenai algoritma pemrograman tentang cara mencari permutasi dari n faktorial yang akan digunakan untuk menentukan apakah suatu graf lintasan dengan n titik termasuk graf graceful atau tidak, dimana akan dilengkapi dengan tampilan listing program, input program, beserta hasilnya.

BAB V: PENUTUP

Pada bab ini dijelaskan tentang kesimpulan dari pembahasan yang telah diuraikan pada bab sebelumnya dan saran-saran yang berkaitan dengan pembahasan.

BAB II

KAJIAN PUSTAKA

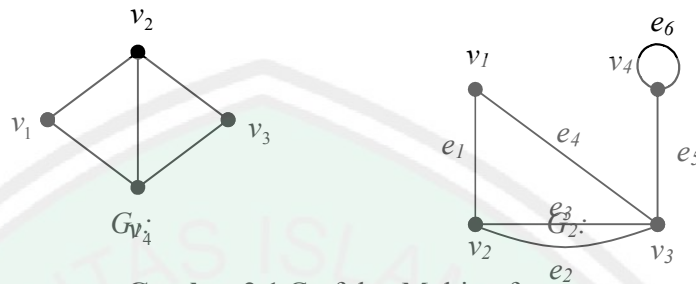
2.1 GRAF

2.1.1 Definisi Graf

Graf G adalah pasangan himpunan (V, E) dengan V adalah himpunan tidak kosong dan berhingga dari obyek-obyek yang disebut sebagai titik dan E adalah himpunan (mungkin kosong) pasangan tak berurutan dari titik-titik berbeda di G yang disebut sebagai sisi. Himpunan titik di G dinotasikan dengan $V(G)$ dan himpunan sisi dinotasikan dengan $E(G)$. Sedangkan banyaknya unsur di V disebut order dari G dan dilambangkan dengan $p(G)$ dan banyaknya unsur di E disebut ukuran dari G dan dilambangkan dengan $q(G)$. Jika graf yang dibicarakan hanya graf G , maka order dan ukuran dari G tersebut cukup ditulis dengan p dan q (Chartrand dan Lesniak, 1986:4).

Dari definisi di atas, maka suatu graf tidak boleh mempunyai sisi rangkap (*multiple edges*) dan loop. Sisi rangkap dari suatu graf adalah jika dua titik yang dihubungkan oleh lebih dari satu sisi. Sedangkan yang disebut dengan loop adalah suatu sisi yang menghubungkan suatu titik dengan dirinya sendiri (Suryanto, 1986:14). Graf yang mempunyai sisi rangkap dan loop disebut multigraf.

Contoh:



Gambar 2.1 Graf dan Multigraf

Pada Gambar 2.1 G_1 merupakan graf, memuat himpunan titik $V(G_1)$ dan sisi $E(G_1)$ yaitu :

$$V(G_1) = \{v_1, v_2, v_3, v_4\}$$

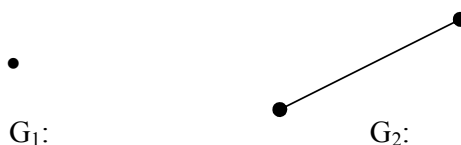
$$E(G_1) = \{(v_1, v_2), (v_2, v_3), (v_2, v_4), (v_3, v_4), (v_4, v_1)\}$$

Graf G_1 mempunyai order 4 atau $p = 4$ dan size 5 atau $q = 5$.

Sedangkan G_2 merupakan multigraf karena mempunyai sisi rangkap e_2, e_3 dan loop pada titik v_4 yaitu e_6 .

Graf G disebut *finite* atau berhingga jika himpunan titik adalah berhingga, atau graf yang jumlah titiknya adalah n berhingga. Graf *infinite* atau tak berhingga adalah graf yang jumlah titiknya tidak berhingga. Graf *trivial* adalah graf berorder satu dengan himpunan sisinya merupakan himpunan kosong. Graf *non trivial* adalah graf yang berorder lebih dari satu (Bondy and Murthy, 1976:3).

Contoh:



Gambar 2.2 G_1 Graf Trivial dan G_2 Graf Non Trivial

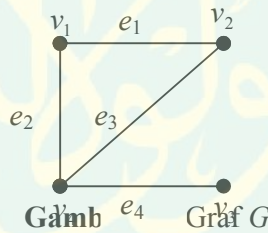
Pada Gambar 2.2 G_1 merupakan graf trivial karena G_1 hanya memuat satu titik atau berorder satu dan himpunan sisinya merupakan himpunan kosong. Graf G_2 merupakan graf non trivial karena berorder lebih dari satu.

2.1.2 Adjacent / Incident

Definisi

Sisi $e = (u, v)$ dikatakan menghubungkan titik u dan v . Jika $e = (u, v)$ adalah sisi di graf G , maka u dan v disebut terhubung langsung (adjacent), u dan e serta v dan e disebut terkait langsung (incident). Untuk selanjutnya, sisi $e = (u, v)$ akan ditulis $e = uv$ (Chartrand dan Lesniak, 1986:4).

Contoh:

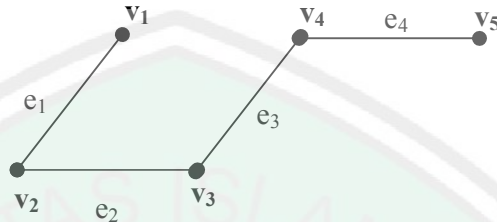


Dari Gambar 2.3, titik yang terhubung langsung adalah titik v_1 dan v_2 , titik v_1 dan v_4 , titik v_2 dan v_3 , titik v_3 dan v_4 . Titik v_2 dan v_4 tidak terhubung langsung karena tidak terdapat sisi (v_2, v_4) pada graf tersebut. Titik yang terkait langsung adalah sebagai berikut:

1. Pada sisi e_1 yang terkait langsung v_1 dan v_2
2. Pada sisi e_2 yang terkait langsung v_1 dan v_4
3. Pada sisi e_3 yang terkait langsung v_2 dan v_3
4. Pada sisi e_4 yang terkait langsung v_3 dan v_4

Contoh lain:

G :



Gambar 2.4 Graf G dengan 5 Titik dan 4 Sisi

Pada graf G Gambar 2.4, titik yang terhubung langsung adalah titik v_1 dan v_2 , titik v_2 dan v_3 , titik v_3 dan v_4 , titik v_4 dan v_5 . Titik v_1 dan v_4 tidak terhubung langsung karena tidak terdapat sisi (v_1, v_4) pada graf tersebut. Titik yang terkait langsung adalah sebagai berikut:

1. Pada sisi e_1 yang terkait langsung v_1 dan v_2
2. Pada sisi e_2 yang terkait langsung v_2 dan v_3
3. Pada sisi e_3 yang terkait langsung v_3 dan v_4
4. Pada sisi e_4 yang terkait langsung v_4 dan v_5

2.1.3 Derajat Titik

Definisi

Derajat dari titik v pada graf G , ditulis $\deg_G(v)$, adalah banyaknya sisi di G yang terkait langsung (incident) dengan v . Jika dalam konteks pembicaraan hanya terdapat satu graf G , maka tulisan $\deg_G(v)$ disingkat menjadi $\deg(v)$. Titik yang berderajat genap sering disebut titik genap (even vertices) dan titik yang berderajat ganjil disebut titik ganjil (odd vertices). Titik yang berderajat nol disebut titik terisolasi (isolated

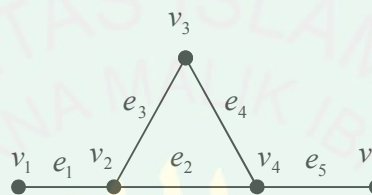
vértices) dan titik yang berderajat satu disebut titik ujung (end vertices) (Chartrand dan Lesniak, 1986:7).

Contoh:

Perhatikan graf G berikut yang mempunyai himpunan titik

$V(G) = \{v_1, v_2, v_3, v_4, v_5\}$ dan himpunan sisi $E(G) = \{e_1, e_2, e_3, e_4, e_5\}$

G :



Berdasarkan Gambar 2.5, maka derajat titik

$$\deg(v_1) = 1$$

$$\deg(v_2) = 3$$

$$\deg(v_3) = 2$$

$$\deg(v_4) = 3$$

$$\deg(v_5) = 1$$

Titik v_2 dan v_4 adalah titik ganjil, titik v_3 adalah titik genap, titik v_1 dan v_5 adalah titik ujung. Hubungan antara jumlah derajat semua titik dalam suatu graf G dengan banyak sisi, yaitu q , adalah

$$\sum_{v \in G} \deg(v) = 2q.$$

Hal ini dinyatakan dalam teorema berikut:

Teorema

Jika G graf dengan $V(G) = \{v_1, v_2, \dots, v_p\}$

maka $\sum_{i=1}^p \deg(v_i) = 2q$ (Chartrand dan Lesniak, 1986:7).

Bukti:

Setiap sisi adalah terkait langsung dengan 2 titik. Jika setiap derajat titik dijumlahkan, maka setiap sisi dihitung dua kali.

Corollary 1.

Pada sebarang graf, banyak titik ganjil adalah genap.

Bukti:

Misalkan graf G dengan banyak sisi (*size*) q . Misalkan W himpunan yang memuat titik ganjil pada G serta U himpunan yang memuat titik genap di G . Dari teorema 1 maka diperoleh:

$$\sum_{v \in V(G)} \deg(v) = \sum_{v \in W} \deg(v) + \sum_{v \in U} \deg(v) = 2q$$

Dengan demikian karena $\sum_{v \in U} \deg(v)$ genap, maka $\sum_{v \in W} \deg(v)$ (jumlah derajat titik ganjil) juga genap. Sehingga $|W|$ adalah genap.

2.1.4 Graf Beraturan - r

Definisi

Graf beraturan-r adalah graf yang semua titiknya berderajat r , atau $\deg(v) = r, \forall v \in V(G)$ (Chartrand dan Lesniak, 1986:9).

Contoh:



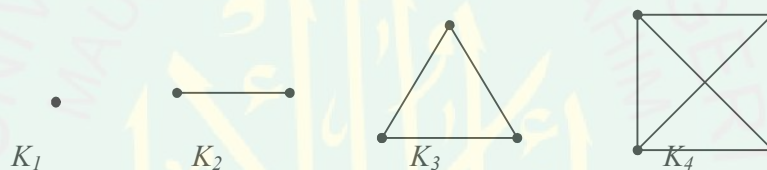
Gambar 2.6 Graf G_1 Beraturan-1 dan G_2 Beraturan-2

Dari Gambar 2.6, graf G_1 disebut graf beraturan-1 karena derajat tiap titiknya adalah 1. Graf G_2 disebut graf beraturan-2 karena derajat tiap titiknya adalah 2.

2.1.5 Graf Komplit

Definisi 5

Graf komplit (*Complete Graph*) adalah graf dengan setiap pasang titik yang berbeda dihubungkan oleh satu sisi. Graf komplit dengan n titik dinyatakan dengan K_n (Purwanto, 1998:21).



Gambar 2.7 Graf Komplit

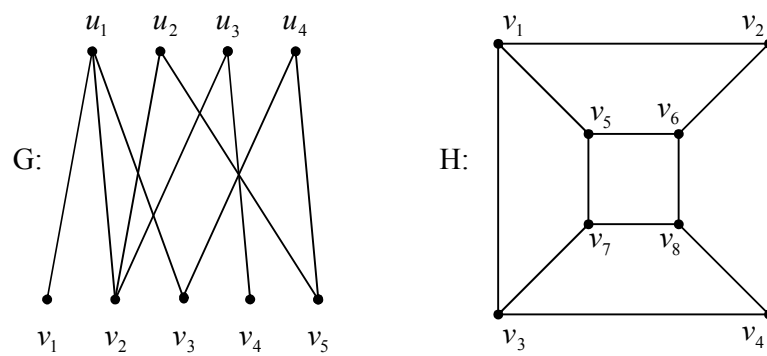
Dari Gambar 2.7, K_1 , K_2 , K_3 dan K_4 adalah graf komplit karena tiap titik dalam graf tersebut selalu *adjacent* dengan semua titik yang lain.

2.1.6 Graf Bipartisi

Definisi

Graf bipartisi (*bipartite graph*) adalah graf yang himpunan titiknya dapat dipisahkan menjadi dua himpunan tak kosong X dan Y sehingga masing-masing sisi di graf tersebut menghubungkan satu titik di X dan satu titik di Y . X dan Y disebut himpunan partisi (Purwanto, 1998:21).

Contoh:



Gambar 2.8 Graf Bipartisi

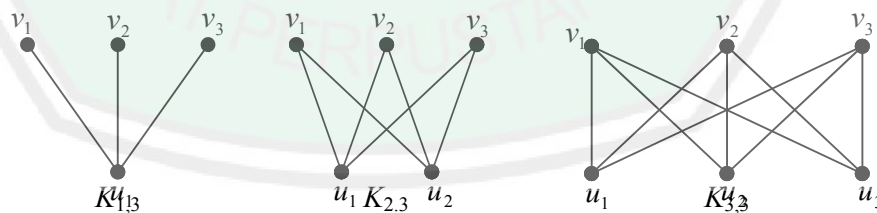
Pada gambar 2.8 G adalah graf bipartisi dengan himpunan partisi $X = \{u_1, u_2, u_3, u_4\}$ dan $Y = \{v_1, v_2, v_3, v_4, v_5\}$ demikian juga H adalah graf bipartisi dengan himpunan partisi $X = \{v_1, v_2, v_3, v_4\}$ dan $Y = \{v_5, v_6, v_7, v_8\}$.

2.1.7 Graf Bipartisi Komplit

Definisi

Graf bipartisi komplit (*complete bipartite graph*) adalah graf bipartisi dengan himpunan partisi X dan Y sehingga masing-masing titik di X dihubungkan dengan masing-masing titik di Y oleh tepat satu sisi. Jika $|X| = m$ dan $|Y| = n$, maka graf bipartisi tersebut dinyatakan dengan $K_{m,n}$ (Purwanto, 1998:22).

Contoh:



Gambar 2.9 Graf Bipartisi Komplit

Pada Gambar 2.9 akan dijelaskan sebagai berikut:

$K_{1,3}$ adalah graf bipartisi komplit dengan:

$$X = \{u_1\}, \quad |X| = 1$$

$$Y = \{v_1, v_2, v_3\}, \quad |Y| = 3$$

$K_{2,3}$ adalah graf bipartisi komplit dengan:

$$X = \{u_1, u_2\}, \quad |X| = 2$$

$$Y = \{v_1, v_2, v_3\}, \quad |Y| = 3$$

$K_{3,3}$ adalah graf bipartisi komplit dengan:

$$X = \{u_1, u_2, u_3\}, \quad |X| = 3$$

$$Y = \{v_1, v_2, v_3\}, \quad |Y| = 3$$

2.2 Graf Terhubung

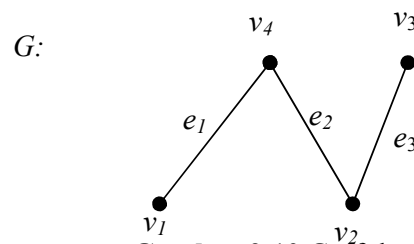
Definisi

Sebuah jalan (*walk*) $u - v$ di graf G adalah barisan berhingga (tak kosong).

$W : u = v_0, e_1, v_1, e_2, v_2, \dots, e_n, v_n = v$ yang berselang seling antara titik dan sisi, yang dimulai dari titik dan diakhiri dengan titik sedemikian hingga untuk $0 < i \leq n$. Dengan $e_i = v_{i-1}v_i$ adalah sisi di G .

v_0 disebut titik awal, v_n disebut titik akhir, $v_1, v_2, \dots, v_{n-1}$ disebut titik internal, dan n menyatakan panjang dari W (Chartrand dan Lesniak, 1986:26).

Contoh:



Gambar 2.10 Graf dengan Jalan

Jalan pada graf G di Gambar 2.10 adalah $W: v_1, e_1, v_4, e_2, v_2, e_3, v_3$

Definisi

Jalan $u - v$ yang semua sisinya berbeda disebut *Trail* $u - v$ (Chartrand dan Lesniak, 1986:26).

Definisi

Trail $u - v$ yang semua titiknya berbeda disebut *path* (lintasan) $u - v$. Dengan demikian, semua lintasan adalah *Trail* (Chartrand dan Lesniak, 1986:26).

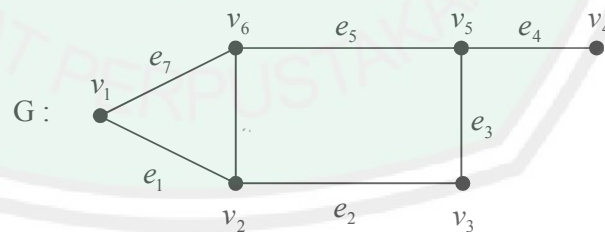
Definisi

Trail tertutup (*closed trail*) yang tak-trivial pada graf G disebut Sirkuit G (Chartrand dan Lesniak, 1986:28).

Definisi

Sirkuit $v_1, v_2, \dots, v_n, v_1$ ($n \geq 3$) dimana v_i adalah titik-titik berbeda untuk $1 \leq i \leq n$ disebut Sikel (*cycle*). (Chartrand dan Lesniak, 1986:28).

Contoh:



Trail pada graf G di Gambar 2.11 adalah Lintasan, Sirkuit dan Sikel.

Lintasan pada graf G di Gambar 2.11 adalah : $v_3, e_2, v_2, e_1, v_1, e_7, v_6, e_5, v_5, e_4, v_4$.

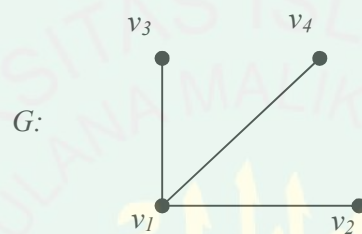
Sirkuit pada graf G di Gambar 2.11 adalah : $v_5, e_5, v_6, e_7, v_1, e_1, v_2, e_2, v_3, e_3, v_5$

Sikel pada graf G di Gambar 2.11 adalah : $v_5, e_5, v_6, e_6, v_2, e_2, v_3, e_3, v_5$

Definisi

Misalkan u dan v titik berbeda pada graf G . Maka titik u dan v dapat dikatakan terhubung (*connected*), jika terdapat lintasan $u - v$ di G . Sedangkan suatu graf G dapat dikatakan terhubung (*connected*), jika untuk setiap titik u dan v di G terhubung (Chartrand dan Lesniak, 1986:28).

Contoh:



Gambar 2.12 Graf Terhubung (*connected*)

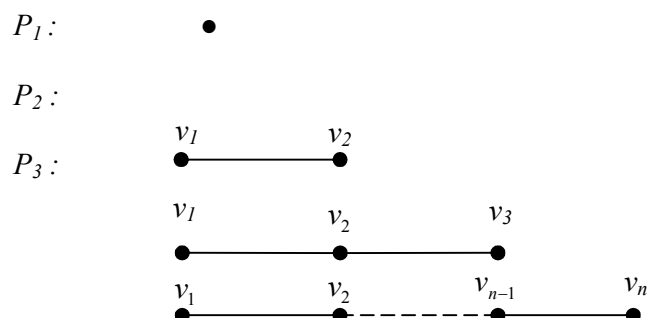
Graf G pada Gambar 2.12 dikatakan terhubung karena setiap titiknya terhubung dengan titik yang lain.

2.3 Graf Lintasan

Graf yang terdiri dari satu lintasan disebut graf lintasan. (Purwanto, 1998:22). Sumber lain mengatakan bahwa graf berbentuk lintasan dengan titik sebanyak n dinamakan graf lintasan dan ditulis P_n (Chartrand & Leniak, 1986:28).

Graf lintasan dengan n titik dinotasikan dengan P_n , dengan n bilangan asli.

Contoh:



P_n :

Gambar 2.13 Graf Lintasan P_2 , P_3 dan P_n

Jadi secara umum graf lintasan P_n mempunyai n titik dan $n - 1$ sisi.

2.4 Pelabelan Graf

Pelabelan pada suatu graf adalah sebarang pemetaan (fungsi) yang memasangkan unsur-unsur graf (titik atau sisi) dengan bilangan (biasanya bilangan bulat). Jika domain dari fungsi adalah titik, maka pelabelan disebut pelabelan titik (*vertex labeling*). Jika domainnya adalah sisi, maka disebut pelabelan sisi (*edge labeling*), dan jika domainnya titik dan sisi, maka disebut pelabelan total (*total labeling*).

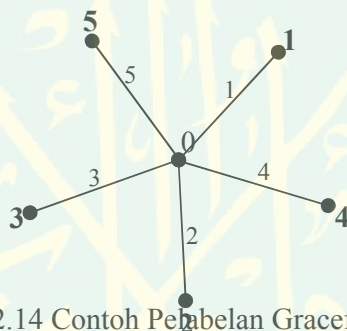
Pelabelan pada graf G adalah pemberian nilai pada titik atau sisi di G . Pada tahun 1967 Rosa menyebutkan λ adalah *valuasi- β* pada graf G jika λ fungsi satu-satu dari himpunan titik di G ke himpunan $\{0, 1, 2, \dots, E(G)\}$ sedemikian hingga, setiap sisi uv di G mendapat label $|\lambda(u) - \lambda(v)|$ yang berbeda semua. Selanjutnya tahun 1972 Golomb menamakannya *pelabelan graceful*.

2.5. Pelabelan Graceful

Misal G graf dengan himpunan titik $V(G)$ dan himpunan sisi $E(G)$. Pelabelan Graceful adalah fungsi injektif f dari $V(G)$ ke $\{0, 1, 2, \dots, |E(G)|\}$ sedemikian hingga jika sisi uv dilabeli $|f(u) - f(v)|$ hasilnya berbeda untuk semua sisi di G (Gallian, 2007: 4).

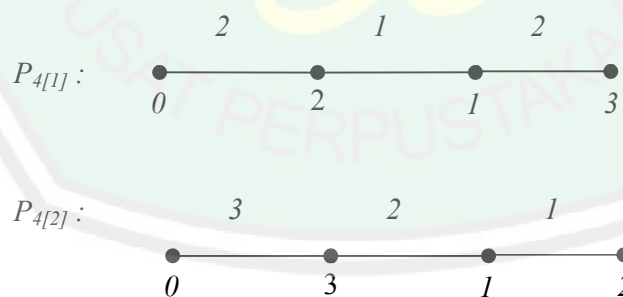
Pelabelan graceful didefinisikan sebagai pemberian label pada titik suatu graf G yang memenuhi fungsi injektif dari himpunan titik ke himpunan bilangan bulat

tak negatif $\{0, 1, 2, \dots, q\}$ sedemikian sehingga jika sisinya mendapat label harga mutlak dari selisih pelabelan kedua titik yang terhubung langsung (adjacent) maka hasilnya berbeda. Dengan demikian, pelabelan graceful merupakan salah satu bentuk pelabelan pada titiknya saja sedangkan label sisinya menjadi akibat dari adanya label titik. Atau pengertian lain, pelabelan graceful pada graf G dengan q sisi adalah fungsi injektif λ dari $V(G)$ ke $\{0, 1, 2, \dots, q\}$ sedemikian hingga, seandainya sisi (x, y) dilabeli dengan $|\lambda(x) - \lambda(y)|$, maka label sisi akan berbeda. Berikut ini adalah contoh pelabelan graceful pada graf star $K_{1,5}$.



Gambar 2.14 Contoh Pelabelan Graceful pada Graf $K_{1,5}$

Contoh lain untuk graf lintasan P_n dengan n Titik



Gambar 2.15 Contoh Pelabelan Graf Lintasan P_n dengan panjang n

Pelabelan pada $P_{4[1]}$ tidak graceful, karena nilai absolut dari masing-masing selisih sisi tidak berbeda

Pelabelan pada $P_{4[2]}$ adalah graceful, karena nilai absolut dari masing-masing selisih sisi berbeda.

2.6 Lexicografic Ordering (Pengurutan Lexicografic)

Secara bahasa, *lexicografic* berarti pengurutan kata atau huruf seperti pada penyusunan kamus. Dalam matematika lexicografic order (LO), biasa juga disebut dengan *alphabetic order* atau *dictionary order*, ialah struktur urutan asli dari Cartesian product dua himpunan terurut. *Lexicografic order* terkadang disebut juga sebagai *dictionary order*.

Seandainya diberikan dua himpunan terurut parsial, A dan B, LO pada Cartesian product $A \times B$ didefinisikan sebagai $(a,b) \leq (a',b')$ jika dan hanya jika $a < a'$ atau $(a = a' \text{ dan } b \leq b')$.

Lexicografic Order: $(a_1, a_2) < (b_1, b_2)$ jika

$$a_1 < b_1 \quad \text{atau}$$

$$a_1 = b_1 \quad \text{dan} \quad a_2 < b_2$$

Contoh:

$$\leq : (Z \times Z) \rightarrow (Z \times Z)$$

$$(3, 5) < (4, 8); (3, 8) < (4, 5); (4, 9) < (4, 11)$$

Ilustrasinya, $a_1, a_2, \dots, a_k$ muncul sebelum $b_1, b_2, \dots, b_k$ jika dan hanya jika a_i pertama, yang mana berbeda dengan b_i muncul sebelum a_i dalam alfabet. Diasumsikan barisan-barisan huruf (kata) tersebut mempunyai panjang yang sama.

Secara umum:

$$\leq : (A_1 \times A_2 \times A_3 \dots \times A_n) \rightarrow (A_1 \times A_2 \times A_3 \dots \times A_n)$$

Lexicografic Order:

$$(a_1, a_2, a_3, \dots, a_n) < (b_1, b_2, b_3, \dots, b_n) \text{ jika } a_1 < b_1$$

$$a_1, a_2, a_3, \dots, a_i = b_1, b_2, b_3, \dots, b_i \text{ dan } a_{i+1} < b_{i+1}$$

Ketika diaplikasikan ke permutasi, menurut Skiena (1990:4) *lexicografic order* adalah pengurutan angka ke atas (*alphabetic order for lists of symbols*). Sebagai contoh, permutasi dari {1,2,3} dalam *lexicografic order* adalah 123, 132, 213, 231, 312, and 321.

2.7 Bahasa Pemrograman

2.7.1 Bahasa Pemrograman PHP

PHP (*PHP Hypertext Proccessor*) adalah *server-side programming* yang populer digunakan untuk membuat *web-based application*. PHP saat ini menjadi salah satu dari *server-side programming* yang paling banyak disukai karena kemudahan penggunaan, tersedianya ratusan *built-in function* serta fleksibilitas modul-modul yang bisa dikembangkan.

PHP pada awalnya bernama PHP/FI (*Personal Home Page/Form Interpreter*) dibuat oleh Rasmus Lerdorf di tahun 1995. Pada saat itu Rasmus Lerdorf membuat bahasa ini untuk digunakan di *website* personalnya, menggunakan script perl. Kemudian, olehnya, PHP dibuat menjadi semakin fungsional dengan menambahkan fungsi-fungsi yang lebih kompleks seperti fungsi akses ke database, dan lain sebagainya. Rasmus memilih untuk me-release source code PHP/FI kepada publik dengan tujuan agar semua orang dapat

menggunakan, menambahkan fungsi-fungsi baru, serta memperbaiki *bugs-bugs* yang ada.

Pada tahun 1997, di release PHP/FI 2.0 menggunakan implementasi bahasa C. Pada saat itu, diperkirakan sekitar ribuan pengguna telah menggunakan PHP/FI serta menginstallnya di server mereka. Pada saat itu pula sudah banyak kontributor yg ikut mengembangkan PHP/FI, walaupun sebenarnya proyek ini masih dimiliki secara individu oleh Rasmus.

Pada Akhir tahun 1997, Andi Gutmans dan Zeev Suraski merombak total implementasi PHP/FI, menambahkan fungsi-fungsi dan kemampuan baru yang akhirnya di release sebagai PHP 3.0. Yang pada akhirnya mereka bertiga setuju untuk mengumumkan kerjasama dimana PHP 3.0 adalah official release dari PHP/FI 2.0 sekaligus menghentikan pengembangan PHP/FI 2 untuk selanjutnya berkonsentrasi pada pengembangan PHP 3.

Sampai hari ini PHP 5 telah dikembangkan dengan Zend Engine 2.0 dengan kemampuan jauh lebih powerfull dibandingkan PHP 4, terutama di sisi OOP (Object Oriented Programming), sehingga bahasa pemrograman PHP tidak hanya digunakan sebagai website namun juga banyak digunakan untuk membuat aplikasi program yang berjalan di PC layaknya program desktop lainnya.

2.7.2. Bahasa Pemrograman Javascript

Javascript diperkenalkan pertama kali oleh Netscape pada tahun 1995. Pada awalnya bahasa ini dinamakan “LiveScript” yang berfungsi sebagai bahasa sederhana untuk browser Netscape Navigator 2. Pada masa itu bahasa ini banyak di kritik karena kurang aman, pengembangannya yang terkesan buru buru dan

tidak ada pesan kesalahan yang di tampilkan setiap kali kita membuat kesalahan pada saat menyusun suatu program. Kemudian sejalan dengan sedang giatnya kerjasama antara Netscape dan Sun (pengembang bahasa pemrograman “Java”) pada masa itu, maka Netscape memberikan nama “JavaScript” kepada bahasa tersebut pada tanggal 4 desember 1995. Pada saat yang bersamaan Microsoft sendiri mencoba untuk mengadaptasikan teknologi ini yang mereka sebut sebagai “Jscript” di browser Internet Explorer 3.

Dengan adanya JavaScript sebuah halaman web akan menjadi lebih dinamis dan interaktif terhadap user, karena halaman web mampu berfungsi sebagai sebuah program aplikasi yang dapat memproses masukan yang diberikan user dan memberikan hasil sesuai dengan yang telah diprogramkan.

Javascript bisa mengakses server dengan menggunakan suatu object yang disebut dengan XMLHttpRequest, metode request dilakukan secara asinkron (di belakang layar yang artinya proses akses tidak terlihat oleh user).

Akibat dari proses kerja JavaScript di atas, maka beban server akan menjadi lebih ringan, dan halaman program akan jauh menjadi lebih cepat merespon.

2.8. Langkah-langkah dalam Pemrograman Komputer

Menurut Pranata (2000 : 4-7) Langkah-langkah yang harus dilakukan dalam pemrograman komputer adalah sebagai berikut :

a. Mendefinisikan Masalah

Sebelum menginjak ke langkah yang kedua, mendefinisikan masalah merupakan langkah yang sangat penting yang berisi menentukan masalahnya

seperti apa, apa saja yang harus dipecahkan dengan komputer, yang terakhir adalah apa masukkannya dan bagaimana keluarannya.

b. Menentukan Solusi

Setelah masalah didefinisikan dengan jelas, masukkan apa yang diberikan sudah jelas, keluaran apa yang diinginkan sudah jelas langkah selanjutnya adalah menentukan bagaimana masalah tersebut diselesaikan. Apabila masalah yang dihadapi terlalu kompleks, kita bisa membaginya ke dalam beberapa bagian kecil agar lebih mudah dalam penyelesaiannya.

c. Memilih Algoritma

Dalam memilih algoritma untuk sebuah program kita harus menentukannya dengan tepat. Karena pemilihan program yang salah akan menyebabkan program memiliki unjuk kerja yang kurang baik.

d. Menulis Program

Dalam langkah ini kita sudah mulai menuliskan program komputer untuk memecahkan masalah yang ada. Dalam menulis program kita juga akan melakukan pemilihan terhadap bahasa pemrograman yang akan digunakan. Ada beberapa hal yang harus dipertimbangkan saat memilih bahasa pemrograman, antara lain masalah yang dihadapi, bahasa pemrograman yang dikuasai, dan sebagainya.

e. Menguji Program

Setelah program selesai ditulis, langkah selanjutnya adalah mengujinya. Pengujian pertama adalah apakah program berhasil dikompilasi dengan baik,

Pengujian berikutnya apakah program dapat menampilkan keluaran yang diinginkan.

Program juga harus diuji untuk kasus yang berbeda, sering terjadi suatu program berjalan baik untuk kasus A, B, C tetapi menghasilkan sesuatu yang tidak diinginkan untuk kasus X, Y, dan Z.

f. Menulis Dokumentasi

Langkah ini biasanya dilakukan bersamaan dengan langkah menulis program tetapi tidak menutup kemungkinan ditulis pada setelah langkah terakhir. Menulis dokumentasi artinya pada setiap beberapa baris program ditambahkan komentar yang menjelaskan kegunaan dari suatu pernyataan.

g. Merawat Program

Langkah ini dilakukan setelah program selesai dibuat dan sudah digunakan oleh pengguna. Hal yang sering terjadi adalah munculnya bug yang sebelumnya tidak terdeteksi. Atau mungkin juga pengguna menginginkan tambahan suatu fasilitas baru. Apabila hal-hal ini terjadi, maka harus dilakukan revisi terhadap program.

2.9. Dasar-Dasar Pemrograman

Pada dasarnya, semua bahasa pemrograman yang berbasis objek mempunyai struktur yang sama, biasanya yang membedakan hanya sintak programnya saja.

Berikut ini beberapa dasar istilah yang ada pada pemrograman:

a. Pembuatan fungsi

Contoh:

```
Function perkalian(a,b){  
    Return a*b;  
}
```

b. Operasi dasar aritmatika

Perkalian : $\text{val1} * \text{val2}$

Pembagian : $\text{val1} / \text{val2}$

Penjumlahan : $\text{val1} + \text{val2}$

Pengurangan : $\text{val1} - \text{val2}$

Modulus : $\text{val1} \% \text{val2}$

c. Operasi relational

$\text{val1} == \text{val2}$

$\text{val1} != \text{val2}$

$\text{val1} < \text{val2}$

$\text{val1} > \text{val2}$

d. Seleksi kondisi (if..else)

```
if (kondisi){  
    expresi jika kondisi bernilai true  
}  
else {  
    expresi jika kondisi bernilai false  
}
```

e. Penggunaan operator switch untuk seleksi kondisi

```
Switch(kondisi){
```

Case '1':

 Expresi jika kondisi = 1

Break;

Case '2':

 Expresi jika kondisi = 2

Break;

Default:

 Expresi jika kondisi tidak terpenuhi

Break;

}

f. Pemakaian looping (for)

for (x=0;x<=10;x++){

 ekspresi perulangan sampai kondisi bernilai false

}

g. Pemakaian looping (do..while)

do{

 ekspresi jika kondisi bernilai true

 x++;

}

while (kondisi)

h. Pemakaian looping (while)

while (kondisi){

ekspresi jika kondisi bernilai true

```
x++;  
}
```

2.10. Keunggulan Aplikasi Menggunakan PHP dan Javascript

Kedua program ini mempunyai fungsi yang berbeda, PHP akan bekerja pada server sedangkan javascript bekerja pada sisi klien. Pemrosesan data dilakukan PHP sedangkan javascript melakukan *request* dengan tanpa *mereload* halaman dan proses dilakukan di belakang layar/ *asynchronous* sehingga sangat efektif digunakan untuk melakukan pemrosesan data matematis khususnya pada simulasi bilangan.

Beberapa kelebihan PHP dan Javascript:

1. Aplikasi lebih mudah dibuat secara dinamis.
2. PHP dapat berjalan dalam beberapa web server: Apache, IIS, Microsoft Personal Web Server, dll.
3. Dapat berjalan dalam beberapa Sistem Operasi yang berbeda, windows, linux, Macintosh, dll
4. Bersifat opensource, diterbitkan secara gratis.
5. Dapat diakses lewat jaringan internet, intranet, bahkan personal computer
6. Bersifat multi user, cukup di install di server jaringan, program dapat dijalankan dari semua komputer yang terkoneksi.
7. Tidak ada virus yang menginfeksi program PHP.

BAB III

METODE PENELITIAN

3.1 Waktu dan Tempat Penelitian

Penelitian ini dilaksanakan mulai tanggal 23 Maret sampai 10 Juli 2009 di Laboratorium Matematika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.

3.2 Jenis Penelitian

Penelitian ini menggunakan metode studi literatur. Penelitian dilakukan dengan pertama kali melakukan kajian terhadap buku-buku teori graf dan jurnal-jurnal atau makalah-makalah yang memuat topik tentang graf graceful graf lintasan dengan n titik yang kemudian diimplementasikan ke dalam program komputer menggunakan bahasa pemrograman PHP dan Java Script.

3.3 Tahapan Penelitian

Langkah-langkah yang dilakukan dalam pembuatan program pada penelitian ini adalah sebagai berikut :

1. Mengidentifikasi Masalah

Masalah pada penelitian ini adalah mengetahui apakah suatu graf lintasan (P_n) dengan panjang n bersifat graceful atau tidak dengan melakukan iterasi hasil permutasi bilangan sebanyak n factorial graf lintasan yang berbeda.

2. Menentukan Solusi

Untuk mengetahui apakah suatu graf lintasan (P_n) dengan panjang n bersifat graceful atau tidak digunakan algoritma lexicographic order atau randomizer untuk mencari setiap nilai permutasi bilangan n factorial graf lintasan yang berbeda.

3. Membuat Diagram Alir

Sebelum peneliti membuat program maka terlebih dahulu dibuat diagram alir. Diagram alir dibuat untuk mempermudah urutan langkah-langkah pembuatan program.

4. Menulis Program

Berdasarkan diagram alir yang telah dibuat sebelumnya, maka permasalahan yang ada diimplementasikan ke dalam program komputer dengan menggunakan bahasa pemrograman PHP dan Javascript.

5. Menguji Program

Setelah program ditulis, program diuji dengan cara menjalankan (*running*) program untuk mengetahui kesalahan yang ada.

6. Menulis Dokumentasi

Menambahkan dokumentasi berupa komentar-komentar pada program. Manfaatnya adalah agar pembaca program (*user*) dapat mengetahui lebih jelas kegunaan dari setiap pernyataan yang ditulis dalam program.

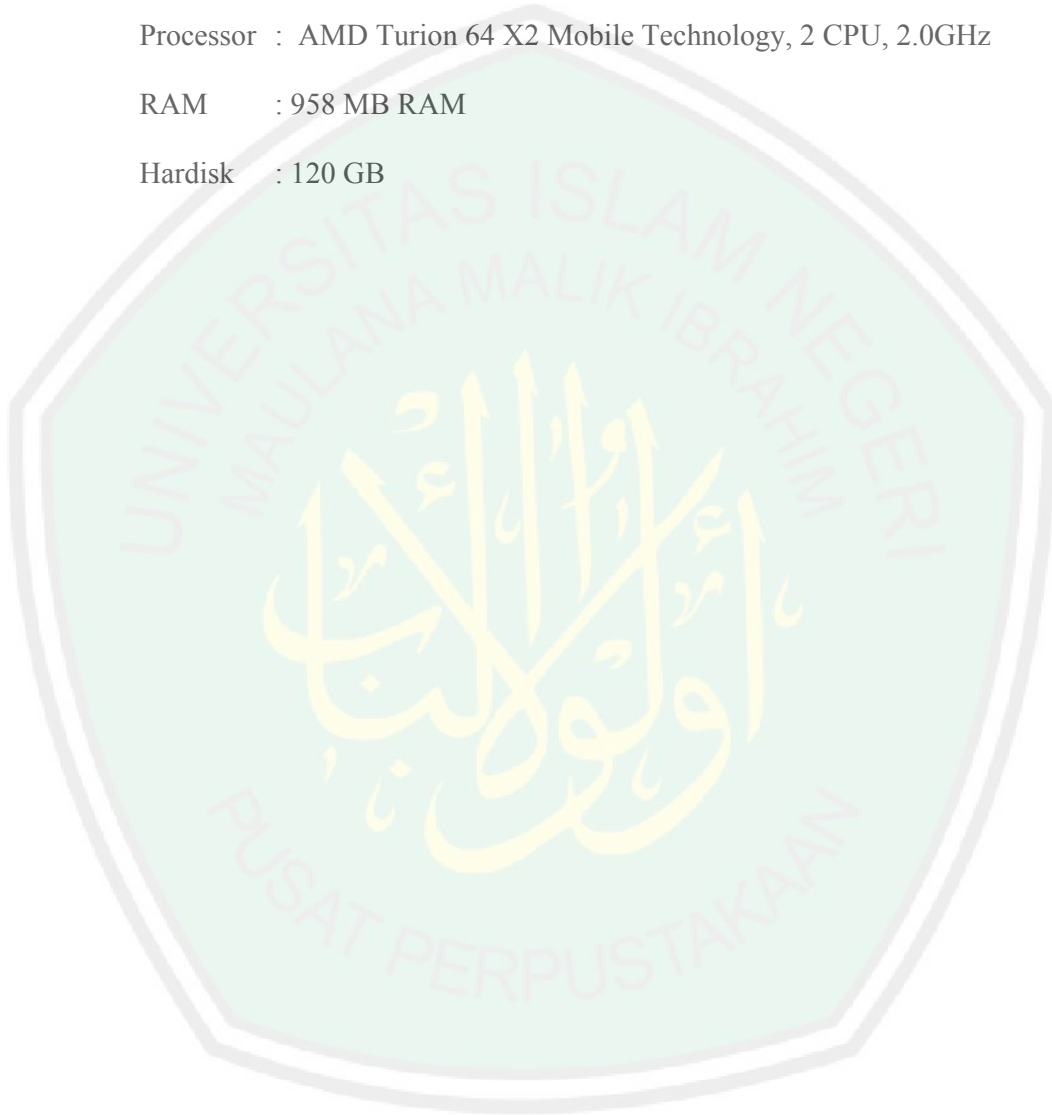
3.4 Alat Penelitian

Sedangkan alat yang digunakan dalam penelitian ini adalah satu unit komputer dengan spesifikasi :

Processor : AMD Turion 64 X2 Mobile Technology, 2 CPU, 2.0GHz

RAM : 958 MB RAM

Hardisk : 120 GB



BAB IV

PEMBAHASAN

Penelitian ini menghasilkan diagram alir, kode program serta output berupa graf lintasan  dengan n titik yang bersifat graceful atau yang tidak graceful, dimana setiap n yang *diinput* mempunyai jumlah iterasi sebanyak n faktorial, Nilai masing-masing iterasi didapat dengan metode lexicographic order dan metode random serta input manual

4.1 Diagram Alir

Diagram alir dibuat untuk mempermudah urutan langkah-langkah penyelesaian masalah. Secara garis besar, diagram alir yang dihasilkan sebagaimana ditampilkan pada halaman selanjutnya.

Setelah program dijalankan, tampilan utama akan muncul, selanjutnya *form input* diisi dengan nilai yang valid. Setelah mendefinisikan form input, masukan dari user akan divalidasi, jika valid maka akan dijalankan pemrosesan data, jika tidak valid maka dikembalikan ke awal form input.

Pemrosesan masukan dari user, ada tiga metode. jika *user* memilih metode pertama yaitu lexicographic order, proses akan di arahkan pada kode program yang berisi algoritma permutasi dengan metode lexicographic order, dimana nilai permutasi diurutkan dari nilai yang paling kecil ke nilai yang paling besar.

Kemudian jika user memilih metode kedua, yaitu random, program akan diarahkan pada kode yang berisi pengacakan nilai *array n* digit untuk menghasilkan nilai permutasinya.

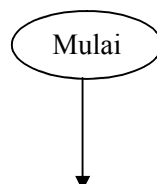
Metode yang ketiga adalah, user memilih input manual pada textarea yang ada beberapa baris bilangan ke *n* yang nantinya bilangan-bilangan tersebut akan diproses dan dicek apakah graceful atau tidak.

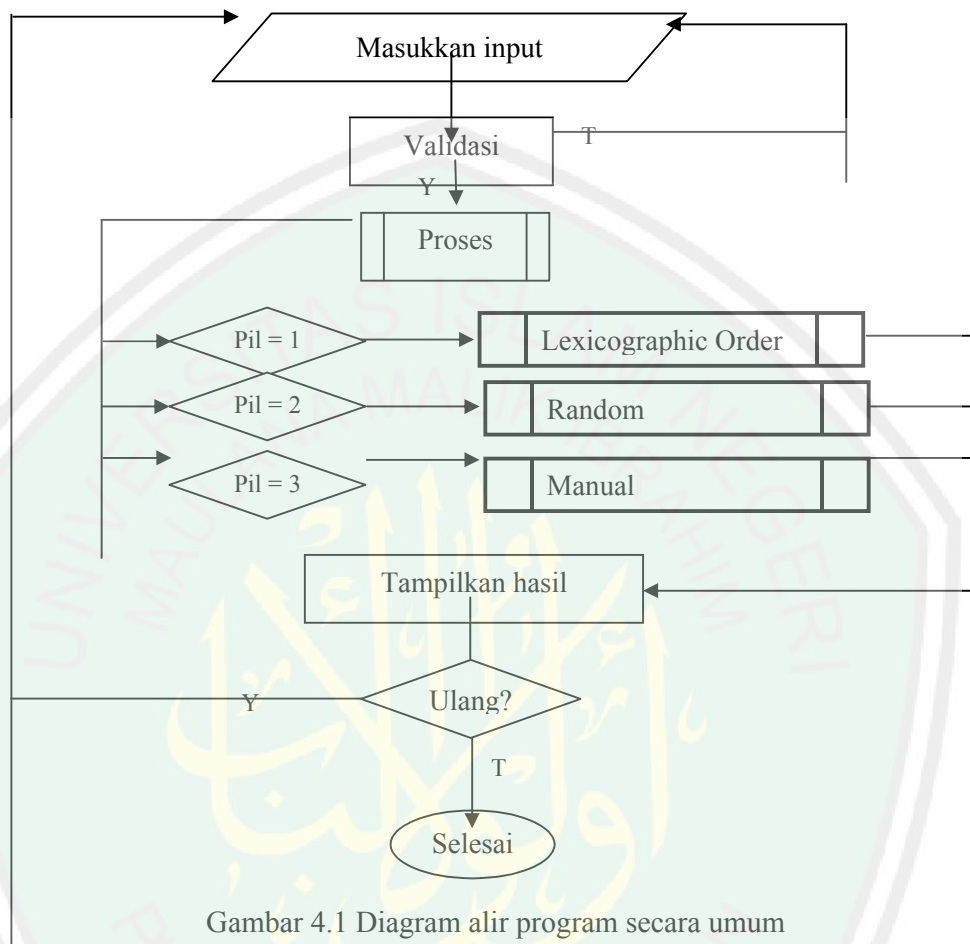
Setelah memilih metode penyelesaian permutasi dan pengecekan bilangan *n* digit, maka program akan mengembalikan hasil pemrosesan data, jika options show graceful only dipilih, maka yang ditampilkan hanya baris bilangan yang graceful saja, namun jika show graceful only tidak dipilih, semua keluaran nilai iterasi ke *n*, baik yang graceful atau tidak akan ditampilkan ke layar.

Demikian juga dengan options display picture, jika dipilih maka seluruh keluaran akan disertai gambar, jika tidak dipilih, hasil yang ditampilkan hanya angka dan teks keterangan.

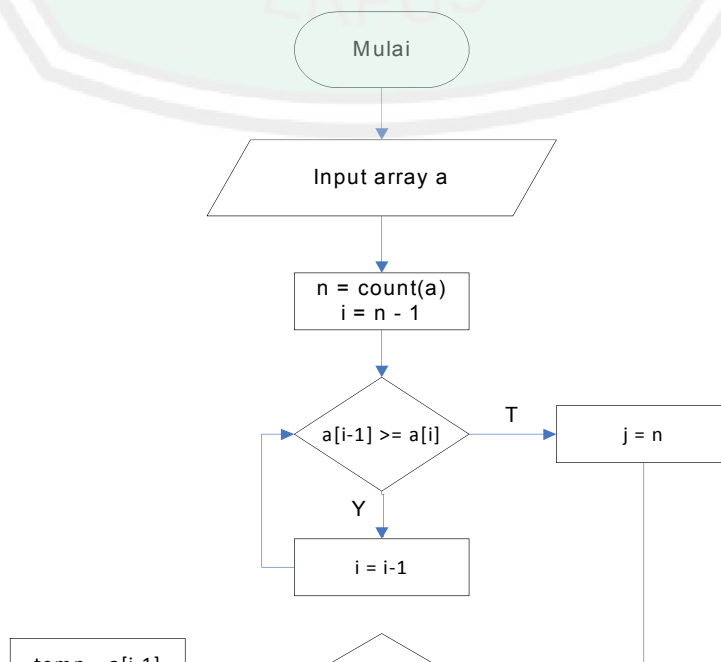
Setelah pemrosesan data *n* digit selesai, maka program akan berhenti melakukan pemrosesan data, dengan menampilkan hasil berapa jumlah pemrosesan data dan berapa jumlah yang graceful serta berapa waktu yang diperlukan dari awal sampai akhir proses.

4.1.1 Diagram alir program secara umum



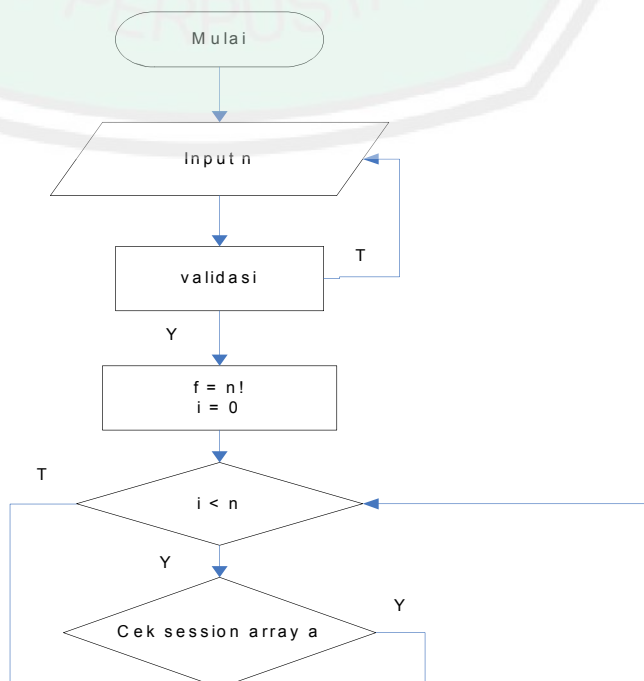


4.1.2 Diagram alir metode lexicographic order



Gambar 4.2 Diagram alir metode lexicographic order

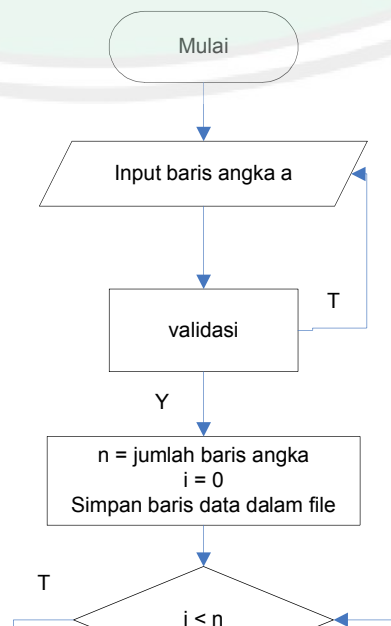
4.1.3 Diagram alir metode Random





Gambar 4.3 Diagram alir metode Random

4.1.4 Diagram alir proses Manual Input





Gambar 4.4 Diagram alir proses Manual Input

4.2 Kode Program

Kode program yang dihasilkan dari penelitian ini adalah beberapa fungsi program yang digunakan untuk mengetahui apakah suatu barisan angka Graf Lintasan P_n dengan n titik bersifat graceful atau tidak.

Kode program PHP dan Javascript bisa tergabung dalam satu file, bisa juga ditulis secara terpisah. Secara garis besar kode program ini terdiri dari :

1. Tampilan Program

Untuk membuat tampilan program, disini penulis menggunakan program HTML dan CSS, konsep tampilan program ini adalah dengan membuat table dan iframe, dimana setiap blok halaman dibuat kolom dan kolom hasil di isi dengan iframe.

2. Fungsi-Fungsi

a) Fungsi startTime dan finishTime

StartTime digunakan untuk menjalankan waktu yang dibutuhkan dalam menyelesaikan pemrosesan data, sedangkan finishTime digunakan untuk menghentikan waktu yang sedang berjalan, baik saat pause maupun saat stop.

```
function startTime(){
    jumWaktu++;
    $('jumWaktu').innerHTML = jumWaktu/100;
    w=setTimeout('startTime()',5);
}

function finishTime(){
    clearTimeout(w);
}
```

b) Fungsi getRandomNumber

Fungsi ini digunakan untuk mengacak angka, dimana angka hasil random tersebut digunakan untuk membuat request data menjadi unik.

```
function getRandomNumber(){
    random = Math.random();
    return random;
}
```

c) Fungsi Validasi

Fungsi ini digunakan untuk memvalidasi input data dari user sebelum pemrosesan data dilakukan, fungsi ini mengembalikan nilai true dan false.

```
function validasi(){
    if ($('#cekListData').checked==false){
        jumDigitVal = $("#jumDigit").value;
        waktuVal = $("#waktu").value;
        if(jumDigitVal == "" || isNaN(jumDigitVal)){
            alert('Enter with the valid number');
            $("#jumDigit").focus();
            return false;
        }
        if (jumDigitVal < 2){
            alert('Number must be greater than one (1).');
            $("#jumDigit").focus();
            return false;
        }
        if(waktuVal == "" || isNaN(waktuVal)){
            alert('Enter with the valid number');
            $("#waktu").focus();
            return false;
        }
        if (waktuVal < 0){
            alert('Number must be greater than zero (0).');
            $("#waktu").focus();
            return false;
        }
    }
    return true;
}
```

d) Fungsi Factorial

Fungsi ini untuk melakukan penghitungan faktorial, metode yang digunakan adalah dengan pendekatan rekursif.

$n! = n \times n-1 \times n-2 \times \dots \times n-1$ untuk $n > 1$

```
function faktorial(bil){
    if (bil < 0) return -1;
    if (bil == 0) return 1;
    if (bil <= 2) return bil;
    return bil*faktorial(bil - 1);
}
```

e). Fungsi cekData

Fungsi ini digunakan untuk melakukan pengecekan metode dan identifikasi beberapa variabel sebelum diproses ke fungsi request.

```
function cekData(){
    if($('prosesData').value == 'CONTINUE'){
        lanjutkan();
    }else{
        URL='stop.php?rd='+getRandomNumber();
        new Ajax.Request(URL,{
            method: 'get',
            onSuccess: function(transport){
            }, onFailure: function(transport){
            }, onLoading: function(transport){
            }
        });

        $('prosesData').value = 'PROCESS';
        $('jumDigit').readOnly = true;
        $('waktu').readOnly = true;
        $('status').innerHTML = 'Identifying...';
        $('prosesData').disabled = true;
        $('status').innerHTML = 'Processing...';
    }
}
```

```

$('jumGraceful').innerHTML = '0';

$('jumCek').innerHTML = '0';

$('jumWaktu').innerHTML = '0';

$('progressPersen').innerHTML = '0';

var myIFrame =
document.getElementById("iframeku").contentWindow;

myIFrame.document.getElementById("hasilProses").innerHTML =
'';

tblHasilGracefulClear();

//mulai proses detal
var bil = $("jumDigit").value;
fact = faktorial(bil);
jumWaktu = 0;
selectMethod = $("tampilSelect").value;
$("tampilSelect").disabled=true;
if(selectMethod==2){
URL='deleteFile.php?jumDigit='+bil+'&rd='+getRandomNumber();

    new Ajax.Request(URL,{
        method: 'get',
        onSuccess: function(transport){
        }, onFailure: function(transport){
        }, onLoading: function(transport){
        }
    });
}

if (listData()!=false){
    fact = listData();

```

```

        selectMethod = 3;

        bil = fact;

        dataGenerated = 0;

    }

    $('jumPermutasi').innerHTML = fact;

    getContent(bil, fact);
    startTime();
}
}

```

f) Fungsi listData

Digunakan apabila pemrosesan menggunakan metode yang ketiga, dimana user menginput baris nilai kedalam textarea, fungsi ini akan mengembalikan jumlah baris namun jika baris kosong, akan mengembalikan nilai false, dan pemrosesan akan dilanjutkan menggunakan metode pertama atau kedua.

```

function listData(){
    if ($('#cekListData').checked==true){
        var dListData = $('#listData').value;

        var text = dListData.replace(/\s+$/g, "");
        var splitq = text.split("\n");

        var pattern = /[0-9]-[0-9]+/;

        if(pattern.test(dListData)){
            var jumData = splitq.length;

            URL='writeTextFile.php?jumDigit='+jumData+'&dListData='+splitq+'&rd='+getRandomNumber();

```

```

new Ajax.Request(URL,{
method: 'get',
onSuccess: function(transport){
}, onFailure: function(transport){
}, onLoading: function(transport){
}
});
return jumData;
}else{
return false;
}
}
return false;
}

```

g) Fungsi lanjutkan

Fungsi ini digunakan setelah user menekan tombol pause, lalu melanjutkan proses dengan menekan tombol continue..

```

function lanjutkan(){
    $('#prosesData').value = 'PROCESS';
    if ($('#cekListData').checked==false){
        $('#jumDigit').readOnly = true;
        $('#waktu').readOnly = true;
        $('#status').innerHTML = 'Identifying...';
        $('#prosesData').disabled = true;
        $('#status').innerHTML = 'Processing...';

        //mulai proses detal
    }
}

```

```

var bil = $("jumDigit").value;

fact = faktorial(bil);

$('jumPermutasi').innerHTML = fact;

```

```

    startTime();

    getContent(bil, fact);
}
}

```

h) Fungsi selesai

Fungsi ini di eksekusi hanya jika pemrosesan data telah selesai dilakukan.

```

function selesai(){
    clearTimeout(t);
    dataGenerated=0;
    URL='stop.php?rd='+getRandomNumber();
    new Ajax.Request(URL,{
        method: 'get',
        onSuccess: function(transport){
        }, onFailure: function(transport){
        }, onLoading: function(transport){
        }
    });
}

```

```

$("tampilSelect").disabled=false;

$('prosesData').value = 'PROCESS';

$('status').innerHTML = 'Finished';

$('prosesData').disabled = false;

$('jumDigit').readOnly = false;

$('waktu').readOnly = false;

```

```
}
```

i) Fungsi stop

Fungsi ini dieksekusi ketika user menekan tombol stop, yaitu untuk menghentikan proses yang sedang dijalankan.

```
function stop(){
    if( $('#prosesData').disabled == true ){
        if(confirm('Are you sure you want to stop this
process?')){
            clearTimeout(t);
            finishTime();
            dataGenerated=0;
            URL='stop.php?rd='+getRandomNumber();
            new Ajax.Request(URL,{
                method: 'get',
                onSuccess: function(transport){
                }, onFailure: function(transport){
                }, onLoading: function(transport){
                }
            });
            $("tampilSelect").disabled=false;
            $('status').innerHTML = 'Stoped';
            $('prosesData').value = 'PROCESS';
            $('prosesData').disabled = false;
            $('jumDigit').readOnly = false;
            $('waktu').readOnly = false;
        }
    }
}
```

j) Fungsi pause

Fungsi ini dipanggil ketika user menekan tombol pause, yang artinya program dihentikan sementara dan dapat dilanjutkan kembali dengan menekan tombol continue.

```
function pause(){
    if( $('#prosesData').disabled == true ){
        if(confirm('Are you sure you want to pause this
process?')){
            clearTimeout(t);
            finishTime();
            if ($('#prosesData').disabled == true){
                $('#status').innerHTML = 'Paused';
                $('#prosesData').disabled = false;
                $('#jumDigit').readOnly = true;
                $('#waktu').readOnly = true;
                $('#prosesData').value = 'CONTINUE';
            }
        }
    }
}
```

k) Fungsi reset

Fungsi ini digunakan untuk melakukan reset dari awal, window akan di reload atau di refresh.

```
function resetAll(){
    window.location.reload();
}
```

l) Fungsi Close

Fungsi ini dijalankan ketika user menekan tombol close, yang artinya akan menutup program window.

```
function tutupWindow(){
  if (confirm("Are you sure you want to close this
window?\nif this button not work, please use Alt + F4")) {
    window.opener='';window.close();
    window.open( '', '_parent' , '' );
    if (!window.opener) {
      window.opener='';
    }
    window.close();
    //alert('cls');
  }
}
```

m) Fungsi getContent

Fungsi ini adalah script yang berisi perintah untuk merequest data beserta hasil pemrosesan data, kode ini dijalankan dengan melakukan perulangan sampai n faktorial.

```
function getContent(bil,fact){
  dataGenerated++;

  var waktuInScond = $('waktu').value;
  var waktu = waktuInScond;

  var myIFrame =
document.getElementById("iframeku").contentWindow;

  if ($('showGracefulOnly').checked==true){
```

```

        var showGracefulOnly = 1;
    }else{
        var showGracefulOnly = 0;
    }
    if ($('#displayPicture').checked==true){
        var displayPicture = 1;
    }else{
        var displayPicture = 0;
    }

    URL='content.php?jumDigit='+bil+'&factor='+fact+'&ke='+dataGenerated+
    '&tampilSelect='+selectMethod+'&showGracefulOnly='+showGracefulOnly+
    '&displayPicture='+displayPicture+'&rd='+getRandomNumber();

    new Ajax.Request(URL,{
        method: 'get',
        onSuccess: function(transport){
            if(transport.responseText){
                var responseData =
transport.responseText;

                var hasilx = new Array();
                hasilx = responseData.split("::");

myIframe.document.getElementById("hasilProses").innerHTML =
hasilx[1];

myIframe.document.getElementById("hasilProses").innerHTML +=
'<br><br>';

```

```

        $('jumCek').innerHTML =

dataGenerated;

        var persen = dataGenerated/fact*100;

        $('progressPersen').innerHTML =

persen;

        if (hasilx[0]==1) { //getGracefull

jumGraceful

                if(hasilx[2]=='g'){

                tabelId =

myIFrame.document.getElementById("tblHasilGraceful");

                baris = tabelId.insertRow(0);

baris.insertCell(0).innerHTML = hasilx[3];

                $('jumGraceful').innerHTML =

parseFloat($('jumGraceful').innerHTML) + 1;

                }else if(hasilx[2]=='ng' &&

$('showGracefulOnly').checked==false){

                tabelId =

myIFrame.document.getElementById("tblHasilGraceful");

                baris = tabelId.insertRow(0);

baris.insertCell(0).innerHTML = hasilx[3];

                }

        if ( dataGenerated >= fact ){

                selesai();

                finishTime();

```

```

        exit();
    }

    t=setTimeout("getContent("+bil+","+fact+")",waktu);
}

if (hasilx[0]==2) {
    dataGenerated--;

    t=setTimeout("getContent("+bil+","+fact+")",waktu);
}

if (hasilx[0]==3) {
    selesai();
    finishTime();
    exit();
}

if (hasilx[0]==4) {
    $('jumCek').innerHTML = 0;
    $('jumPermutasi').innerHTML = 0;

    $('progressPersen').innerHTML = 0;
    selesai();
    finishTime();
    exit();
}

}else{
    alert('respon gagal');
}

}, onFailure: function(transport){

```

```

        alert('gagal');

    }, onLoading: function(transport){

    }

});
}

```

n) Fungsi tblHasilGracefulClear

Fungsi ini digunakan untuk membersihkan iframe, yaitu tempat hasil pemrosesan data.

```

function tblHasilGracefulClear(){
    var myIFrame =
    document.getElementById("iframeku").contentWindow;
    tabelId =
    myIFrame.document.getElementById("tblHasilGraceful");
    barisTerakhir = tabelId.rows.length;
    if(barisTerakhir>0 && barisTerakhir<1000){
        for (iRows=barisTerakhir-1;iRows>=0;iRows--) {
            tabelId.deleteRow(iRows);
        }
    }
    if(barisTerakhir>1000){
        alert('To clean up the form, the window will
reloaded. ');
        window.location.reload();
    }
}

```

o) Fungsi Petunjuk

Fungsi bantuan menampilkan petunjuk pemakaian program. Fungsi ini memanggil dokumen petunjuk.

p) Fungsi swapNum

Fungsi ini digunakan untuk menukarkan nilai dari dua buah variabel.

```
function swapNum(&$x, &$y) {
    $x ^= $y ^= $x ^= $y;
}
```

q) Fungsi penentuan nilai permutasi dengan lexicographic order

Fungsi ini digunakan untuk mendapatkan nilai permutasi selanjutnya yang dibentuk kedalam sebuah *array* dengan memakai metode lexicographic order.

```
function nextArray($a){
    $n = count($a);
    $i = $n - 1;

    while( $a[$i-1] >= $a[$i] ){
        $i = $i - 1;
    }

    $j = $n;
    while( $a[$j-1] <= $a[$i-1] ){
        $j = $j - 1;
    }

    swapNum($a[$i-1], $a[$j-1], $a);
}
```

```
$i++;
```

```
$j = $n;
```

```
while ($i < $j) {
```

```
    swapNum($a[$i-1], $a[$j-1], $a);
```

```
    $i++;
```

```
    $j--;
```

```
}
```

```
return $a;
```

```
}
```

r) Fungsi Penyelesaian Randomizer

Fungsi ini digunakan untuk mendapatkan nilai permutasi selanjutnya yang di bentuk kedalam sebuah *array* dengan mengacak nilai dari *array* sebelumnya.

```
shuffle($_SESSION['numbers']);
```

s) Fungsi pilihan show graceful dan display picture

Fungsi ini digunakan untuk mengecek apakah suatu angka n digit (graf lintasan P_n dengan n Titik) berupa graceful atau tidak. Kemudian jika pilihan display picture di pilih, program akan menggenerate barisan angka tersebut menjadi sebuah gambar yang akan ditampilkan di iframe hasil.

```
function isGracefull($angka) {
```

```
    global $showGracefulOnly, $displayPicture, $stampilSelect;
```

```
    if(!empty($angka)) {
```

```
        $pecah = explode("-", $angka);
```

```
        $jumPecah = count($pecah);
```

```

$selisih = array();

for($i=0;$i<$jumPecah-1;$i++){

$selisih[$i] = abs($pecah[$i] - $pecah[$i+1]);

}

$selisihUnik = array_unique($selisih);
$jumselisihUnik = count($selisihUnik);
$jumSelisih = count($selisih);

if($jumSelisih==$jumselisihUnik){
$return = "::g::";

$return .= $angka.'<br>';

$return .= '&nbsp; ' .implode("-", $selisih).'
&raquo; Graceful';
}else{
$return = "::ng::";

if ($showGracefulOnly!=1) {
$return .= $angka.'<br>';

$return .= '&nbsp; ' .implode("-", $selisih).'
&raquo; <font color="#FF0000">Not Graceful</font>';
}
}

if($displayPicture==1 AND $jumSelisih!=1) {
$return .= '

<table id="Table_01" width="200" border="0"
cellpadding="0" cellspacing="0">

<tr>

<td align="left" valign="top"></td>';

```

```

        for($i=0;$i<$jumSelisih;$i++){//aslinya 09

            $return .= '<td align="center"
valign="middle" background="images/charts_08.png" width="40"
height="34"><b>'. $selisih[$i]. '</b></td>';

            $return .= '<td align="left"
valign="top"></td>';

        }

        $return .= '</tr>';

        $return .= '<tr>';
        for($i=0;$i<$jumSelisih;$i++){
            if($i==0){
                $return .= '<td align="left"
valign="top"></td>';
            }else{
                $return .= '<td align="left"
valign="top"></td>';
            }

            $return .= '<td align="left"
valign="top"></td>';

        }
    }

```

```

        $return .= '<td align="left"
valign="top"></td>';

        $return .= '</tr>';

        $return .= '<tr>';
        for($i=0;$i<$jumSelisih;$i++){
            $return .= '<td align="center"
valign="middle" background="images/charts_17.png" width="40"
height="27">'. $specah[$i]. '</td>';

            $return .= '<td align="left"
valign="top"></td>';
        }
        $return .= '<td align="center"
valign="middle" background="images/charts_17.png" width="40"
height="27">'. $specah[$jumSelisih]. '</td>';

        $return .= '</tr>';

        $return .= '</table>';
    }
    $return .= '<br><br>';
}

return $return;
}

```

t) Kode program hasil proses

Kode ini digunakan sebagai tempat menampilkan proses generate data.

```
<div id="hasilProses">&nbsp;</div>
```

u) Kode program hasil graceful

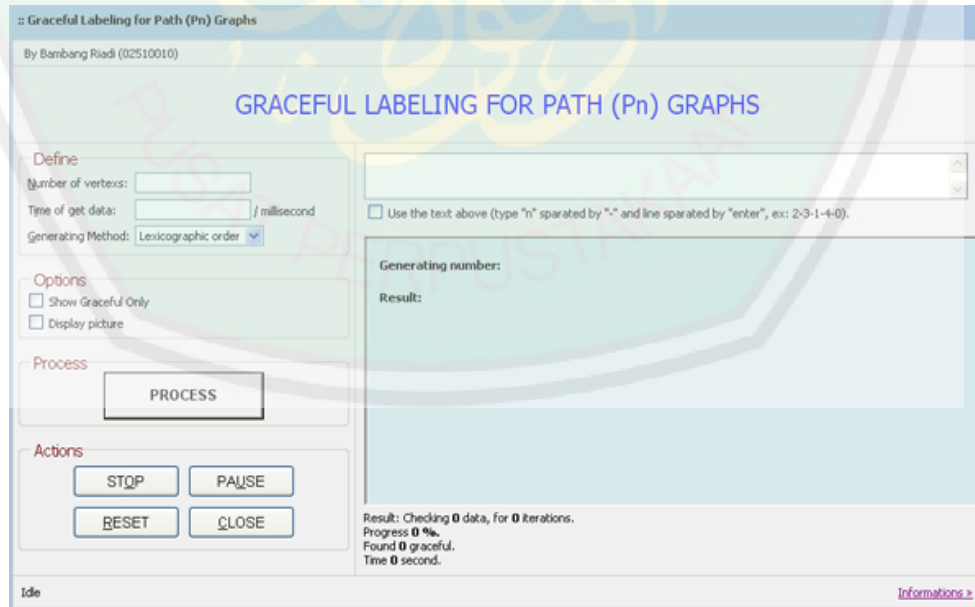
Setiap hasil generate angka, akan ditampilkan di tabel ini. Baik hanya berupa teks maupun yang disertai gambar.

```
<table id="tblHasilGraceful">
  <tr>
    <td>&nbsp;</td>
  </tr>
</table>
```

Adapun kode program secara keseluruhan dapat dilihat pada lampiran 2.

4.3 Output Program

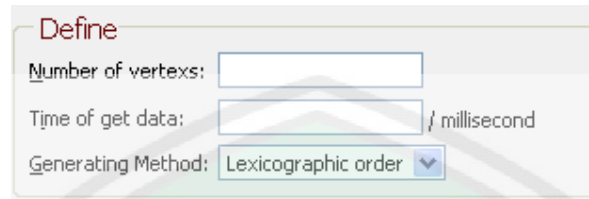
Pada saat pemanggilan pertama program, tampak Halaman Utama :



Gambar 4.5 Output Program: Halaman Utama

Berikut beberapa keterangan blok program:

a. Blok Define



Gambar 4.6 Blok Define

Blok ini digunakan untuk mendefinisikan data yang akan diproses. *Number of vertices* diisi dengan bilangan asli > 1 karena himpunan titik dalam graf tidak mungkin berupa himpunan kosong. *Time of get data* diisi dengan angka ≥ 0 . *Generating method* adalah pilihan untuk memilih metode generate data, pakai *lexicographic order* atau *randomizer*.

b. Blok Options



Gambar 4.7 Blok Options

Blok option digunakan sebagai pilihan, yang terdiri dari dua pilihan, yaitu *show graceful only* dan *display picture*.

Jika *show graceful only* dipilih, maka hasil yang akan ditampilkan hanya yang berlabel graceful, sebaliknya jika tidak dipilih hasil akan ditampilkan semuanya.

c. Blok Proses





Gambar 4.8 Blok Proses

Blok proses berisi tombol untuk proses data. Jika tombol ini dipilih maka proses data akan dimulai dan secara otomatis tombol menjadi disabled, jika user menekan tombol pause maka value dari process akan berubah menjadi continue. Jika tombol continue di tekan maka proses data akan dilanjutkan dan secara otomatis tombol menjadi disabled.

d. Blok Actions



Gambar 4.9 Blok Actions

Blok action terdiri dari empat tombol, yaitu:

Stop : button ini digunakan untuk menghentikan proses data yang sedang berjalan.

Pause : button ini digunakan untuk menghentikan sementara proses yang sedang berjalan, proses akan dilanjutkan lagi setelah user menekan tombol continue.

Reset : button ini digunakan untuk mereset program dari awal, dimana halaman program akan direload/di refresh

Close : button ini digunakan untuk menutup program window.

e. Manual Textbox

2-6-0-5-3-4-1
2-6-0-5-4-1-3
3-0-6-1-5-4-2

☒ Use the text above (type "n" sparated by "-" and line sparated by "enter", ex: 2-3-1-4-0).

Gambar 4.10 Manual Texbox

Text area ini di isi jika user ingin mendefinisikan barisan angka graf lintasan secara manual, data akan diproses jika textarea tidak kosong dan checkbox dibawahnya dipilih.

Format penulisan barisan graf lintasan adalah dengan memisahkan masing-masing angka dengan tanda “-” dan setiap baris dipisahkan dengan menekan tombol “enter” di keyboard.

f. Form hasil

Generating number:
3-0-6-1-5-4-2

Result:
3-0-6-1-5-4-2
3-6-5-4-1-2 » Graceful

2-6-0-5-4-1-3
4-6-5-1-3-2 » Graceful

2-6-0-5-3-4-1
4-6-5-2-1-3 » Graceful

2-5-4-0-6-1-3
3-1-4-6-5-2 » Graceful



Gambar 4.11 form Hasil

Form ini untuk menampung hasil dari generate data, hasil dapat berupa teks saja atau dengan gambar.

g. Summary

Result: Checking **120** data, for **120** iterations.
Progress **100 %**.
Found **8** graceful.
Time **2.39** second.

Hasil dari proses data:

Misalkan $n = 5$

Result : program telah melakukan pengecekan data sebanyak 120 dari 120 total iterasi.

Progress : proses telah dijalankan 100%

Ditemukan 8 graceful dalam waktu 2.39 detik.

Result: Checking **3199** data, for **5040** iterations.
Progress **63.472222222222 %**.
Found **27** graceful.
Time **60.94** second.

$N = 7$

Result : program telah melakukan pengecekan data sebanyak 3199 dari 5040 total iterasi.

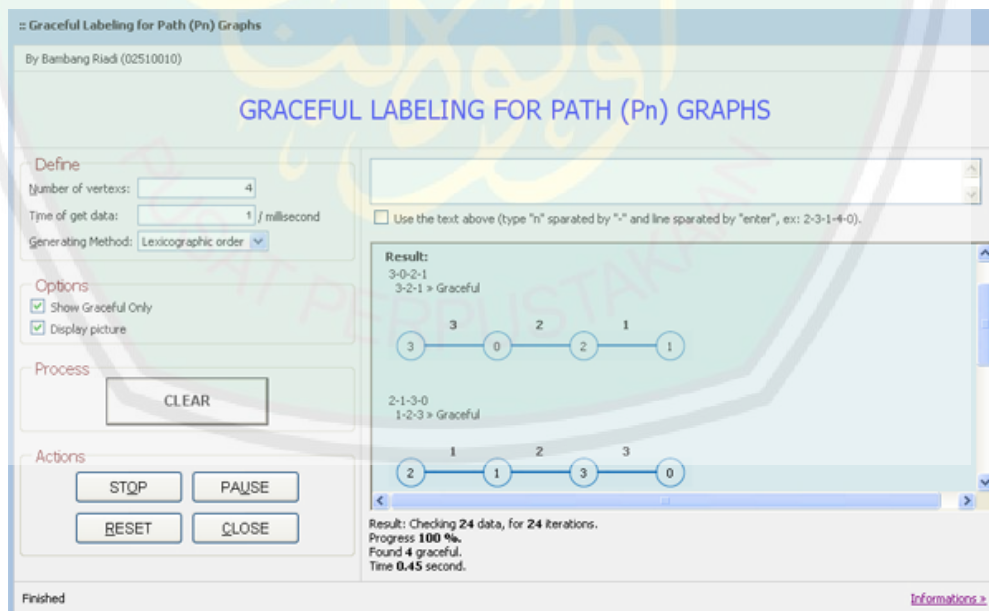
Progress : proses telah dijalankan 63.47%

Ditemukan 27 graceful dalam waktu 60.94 detik.

4.4 Pengujian Program

Number of vertexs dimasukkan nilai integer > 1 , yang artinya jumlah iterasi adalah jumlah faktorial dari digit yang dimasukkan. *Time of get data* adalah waktu *request* saat hasil telah didapat (dalam *millisecond*).

Pilih metode generate permutasi data (*lexicographic* atau *random*). *Check show graceful only*, Jika ingin hanya graceful yang ditampilkan. *Check display picture*, jika ingin menampilkan gambar dari angka yang diproses. Lalu setelah selesai klik tombol *PROCESS*.



Gambar 4.12 Menu Utama

Proses detailnya adalah sebagai berikut:

Cek nilai *vertex* yang diinput, $n=4$

Cek waktu *request* data, $t=1$

Dikirimkan informasi tersebut ke proses.php

Jumlah iterasi = $n! = 4! = 24$

Karena menggunakan *lexicographic order* maka setiap iterasi di ambil satu persatu.

Iterasi akan dilakukan mulai dari $i=1$ sampai $i=n!$

Jika $i=1$ akan dikembalikan nilai 0-1-2-3

1. 0-1-2-3 di proses dengan cara memecah masing angka menjadi *array* $x[0]=0$, $x[1]=1$, dst
2. Kemudian selisih dari masing-masing angka dimasukkan dalam *array* $y[i] = \text{abs}(x[i] - x[i+1])$
 $y[0] = \text{abs}(x[0] - x[0+1]);$
 $y[0] = \text{abs}(x[0] - x[1]);$
 $y[0] = \text{abs}(0 - 1);$
 $y[0] = 1;$
3. Hitung jumlah *array* selisih $y[i] \Rightarrow$
 $p = \text{count}(y[0], y[1], y[2]);$
 $p = \text{count}(1, 1, 1);$
 $p = 3;$
4. Hapus nilai dalam *array* y yang mempunyai duplicate sehingga menghasilkan *array* yang unik (dalam setiap selisih tidak boleh sama).
5. Hitung jumlah *array* unik $\Rightarrow$
 $u = \text{array_unique}(y[0], y[1], y[2]);$

```
u = array_unique (1, 1, 1);
```

```
u = count(1);
```

```
u = 1;
```

6. Bandingkan nilai p dan u, jika nilai p = u maka graf graceful, jika p > u maka graf tidak graceful. Dari hasil i = 1 dapat dilihat bahwa p > u yang artinya i = 1 tidak graceful.

Jika pilihan show graceful only di pilih, maka hasil tidak ditampilkan dan lanjut ke iterasi berikutnya.

Jika i=2 akan dikembalikan nilai 0 1 3 2

Jika i=3 akan dikembalikan nilai 0 2 1 3

Jika i=5 akan dikembalikan nilai 0 3 1 2

1. 0-3-1-2 di proses dengan cara memecah masing angka menjadi array x[0]=0, x[1]=3, dst

2. Kemudian selisih dari masing-masing angka dimasukkan dalam array y[i] = abs(x[i]- x[i+1])

```
y[0] = abs(x[0]- x[0+1]);
```

```
y[0] = abs(x[0]- x[1]);
```

```
y[0] = abs(0- 3);
```

```
y[0] = 3;
```

3. Hitung array selisih y[i] =>

```
p = count(y[0], y[1], y[2]);
```

```
p = count(3, 2, 1);
```

```
p = 3;
```

4. Hapus nilai dalam *array* *y* yang mempunyai *duplicate* sehingga menghasilkan *array* yang unik (dalam setiap selisih tidak boleh sama).

hitung jumlah *array* unik =>

```
u = array_unique (y[0], y[1], y[2]);
```

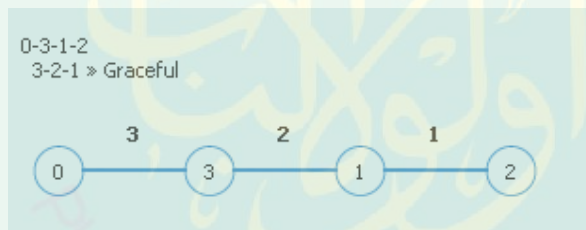
```
u = array_unique (3, 2, 1);
```

```
u = count(3, 2, 1);
```

```
u = 3;
```

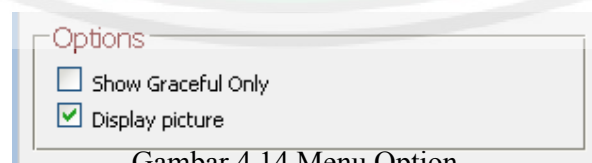
5. Bandingkan nilai *p* dan *u*, jika nilai $p = u$ maka graf graceful, jika $p > u$ maka graf tidak graceful. Dari hasil $i = 5$ dapat dilihat bahwa $p = u$ yang artinya $i = 5$ graceful.

Pilihan *show graceful only* dipilih atau tidak, hasil akan ditampilkan dan dilanjutkan ke iterasi berikutnya.

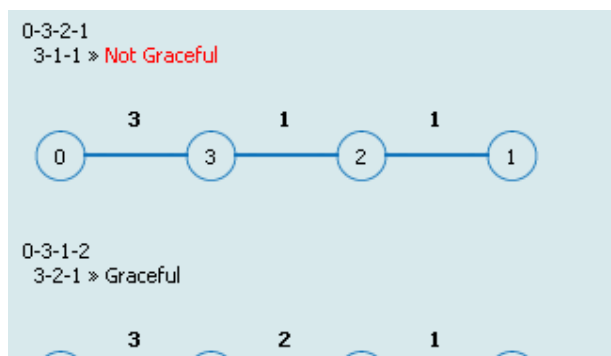


Gambar 4.13 output graceful

Kemudian jika pilihan *show graceful only* tidak dipilih, maka semua hasil proses data akan ditampilkan.



Gambar 4.14 Menu Option



Gambar 4.15 Output non Graceful dan Graceful

Jika pilihan *display picture* tidak dipilih yang ditampilkan hanya teks dan gambar tidak ikut ditampilkan, hal ini untuk mempercepat proses *request* data.

Result:

3-2-1-0
1-1-1 » Not Graceful

3-2-0-1
1-2-1 » Not Graceful

3-1-2-0
2-1-2 » Not Graceful

3-1-0-2
2-1-2 » Not Graceful

Hasil dari pengujian ini adalah :

3-0-2-1
3-2-1 » Graceful
Untuk pelabelan graceful pada graf lintasan P_n dengan $n = 4$, diperoleh jumlah graceful = 4 baris dari 24 iterasi dalam waktu 0.57 detik.

3-0-2-1



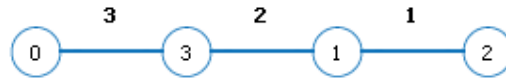
2-1-3-0



1-2-0-3

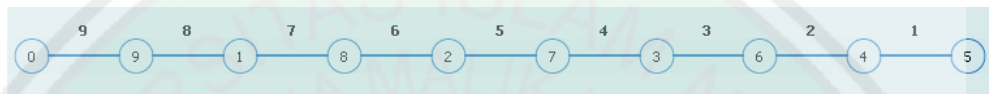


0-3-1-2

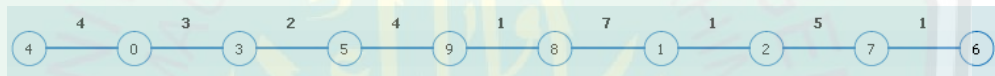


Gambar 4.16 Output Program Pelabelan Graf pada Graf Lintasan P_4

Contoh lain, untuk pelabelan graceful pada graf lintasan P_n dengan $n = 10$, seperti gambar dibawah ini:



Gambar 4.17 Output Program Pelabelan Graf Lintasan P_{10} (Graceful)



Gambar 4.18 Output Program Pelabelan Graf Lintasan P_{10} (Tidak Graceful)

Dari beberapa hasil pengujian, pemrosesan data akan mendapatkan hasil yang akurat pada banyaknya pelabelan graf lintasan n titik. Namun dibutuhkan waktu yang lama untuk melakukan pemrosesan data pelabelan graceful pada graf lintasan dengan jumlah n titik yang besar. Sebagai alternatif, jika pada graf lintasan yang akan diproses mempunyai jumlah titik yang besar, digunakan metode random untuk melakukan iterasi secara acak atau melakukan input manual dengan memasukkan baris angka lintasannya.

BAB V

PENUTUP

5.1 Kesimpulan

Dari penelitian yang telah dilakukan diperoleh kesimpulan bahwa telah dibuat program komputer yang menentukan pelabelan graceful suatu graf lintasan (P_n) dengan panjang n titik.

Adapun langkah-langkah untuk menentukan pelabelan graceful pada graf lintasan (P_n) dengan program komputer, dimana n bilangan asli dan $n > 1$ adalah sebagai berikut:

1. Menjadikan n menjadi barisan angka
 $(a_1, a_2, a_3, \dots, a_n) < (b_1, b_2, b_3, \dots, b_n)$ jika $a_1 < b_1$
 $a_1, a_2, a_3, \dots, a_i = b_1, b_2, b_3, \dots, b_i$ dan $a_{i+1} < b_{i+1}$
2. Nilai a disimpan menjadi sebuah array $a[n]$.
3. Nilai pada $a[n]$ di proses dengan cara memecah masing angka menjadi array $x[n], x[n+1], \dots$.
4. Kemudian selisih dari masing-masing angka dimasukkan dalam array $y[i] = \text{abs}(x[i] - x[i+1])$, dimana $i = n-1$.
5. Hitung array selisih $y[i] \Rightarrow$
 $p = \text{count}(y[0], y[1], y[i]);$
6. Hapus nilai dalam array y yang mempunyai duplicate sehingga menghasilkan array yang unik (dalam setiap selisih tidak boleh sama).

7. Hitung jumlah array unik =>

`u = array_unique (y[0], y[1], y[i]);`

8. Bandingkan nilai p dan u, jika nilai $p = u$ maka graf graceful, jika $p > u$ maka graf tidak graceful.

Hasil pengecekan di atas didapat keluaran barisan angka graceful atau tidak graceful beserta gambar pelabelannya.

5.2 Saran

Berdasarkan kesimpulan di atas, penulis menyarankan beberapa hal sebagai berikut:

1. Disarankan kepada peneliti lain untuk melanjutkan pembuatan program dengan memodifikasi dan mengembangkan untuk pelabelan graceful pada jenis graf yang lain.
2. Disarankan bagi peneliti lain, dalam pembuatan metode yang digunakan lebih diorientasikan pada pengambilan nilai yang bersifat graceful saja, sehingga pemrosesan data bisa lebih cepat.

DAFTAR PUSTAKA

- Abdussakir. 2005. Edge-Magic Total Labeling pada Graph mP_2 (m Bilangan Asli Ganjil). Jurnal *Saintika*, Edisi Khusus Dies Natalis I UIN Malang, Juni. Halaman 22-27.
- Achour, Mehdi, dkk. 2007. *PHP Manual* {book on-line}, Mass: The PHP Documentation Group. <http://www.php.net/docs.php>. Diakses tanggal 09 Desember 2007.
- Azis, Farid. 2004. *Belajar Sendiri Pemrograman PHP 4*. Jakarta: Elexmedia Komputindo.
- Bondy, J.A. & Murty, U.S.R., 1976. *Graph Theory with Applications*. London: The Macmillan Press.
- Chartrand, G. & Lesniak, L. 1986. *Graph and Digraph 2nd Edition*. California: Wadsworth, Inc.
- Flanagan, David. 2001. JavaScript: *The Definitive Guide, 4th Edition* {book on-line}. Mass: O'Reilly. www.davidflanagan.com. Diakses pada tanggal 20 Mei 2009.
- Gallian, Joseph. A. 2007. A Dynamic Survey of Graph Labeling. (Online): (<http://www.combinatorics.com>). Diakses Tanggal 17 Juni 2009).
- McLeod, Raymond Jr., George Shell. 2004. *Sistem Informasi Manajemen terjemahan Edisi Kedelapan*. Jakarta: Indeks.
- Miller, Mirka. 2000. *Open Problems in Graph Theory: Labelings and Extremal Graphs*. Prosiding Konferensi Nasional Himpunan Matematika Indonesia X di Institut Teknologi Bandung, 17-20 Juli.
- Pranata, Antony. 2000. Algoritma dan Pemrograman. Yogyakarta : J&J Learning
- Prasetyo, Didik Dwi. 2006. *101 Tip dan Trik Pemrograman PHP*. Jakarta : PT Elex Media Komputindo.
- Purwanto, 1998. Matematika Diskrit. Malang: IKIP Malang.
- Siang, J.J.. 2002. *Matematika Diskrit dan Aplikasinya pada Ilmu Komputer*. Yogyakarta: Andi Offset.

Skiena, S.Steven. 1990. *The Algorithm Design Manual*. Jakarta: Springer

Suryanto. 1986. Materi Pokok Pengantar Teori Graph. Karunia Universitas Terbuka: Jakarta

Wijaya, K dan Baskoro, E.T. 2000. *Pelabelan Total-Sisi Ajaib pada Gabungan Graf-graf Lingkaran*. Prosiding Konferensi Nasional Himpunan Matematika Indonesia X di Institut Teknologi Bandung, 17-20 Juli.

