

PENGELOLAAN JADWAL PENGGUNAAN LABORATORIUM KOMPUTER DENGAN ALGORITMA PEWARNAAN GRAF WELCH POWELL PADA UNIVERSITAS DIAN NUSWANTORO SEMARANG

Ikhsan Musmi Aji¹

¹ Sistem Informasi, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro
Jl. Nakula I No. 5 – 11 Semarang 50131, Telp (024) 3515261 Indonesia
E-mail : ic4an91@gmail.com

Abstrak

Universitas Dian Nuswantoro merupakan universitas swasta di kota Semarang yang unggul dalam dunia IT nya. Universitas Dian Nuswantoro memiliki beberapa Fakultas, diantaranya Fakultas Ilmu Komputer, Fakultas Ekonomi Bisnis, Fakultas Kesehatan, Fakultas Teknik. Pada Universitas Dian Nuswantoro memiliki Fakultas unggulan yaitu Fakultas Ilmu Komputer. Universitas Dian Nuswantoro di dalamnya memiliki fasilitas yaitu salah satunya laboratorium komputer. Pada laboratorium komputer digunakan untuk mata kuliah praktikum dari berbagai jurusan. Terdapatnya fasilitas laboratorium komputer sangat menunjang dalam proses belajar mengajar pada perkuliahan praktikum. Pada laboratorium komputer terdapat ruang LAB, jenis LAB, mata kuliah praktikum, serta jam penggunaannya. Perkuliahan praktikum di laboratorium komputer Universitas Dian Nuswantoro berlangsung pada hari senin sampai dengan hari jumat. Pada Universitas Dian Nuswantoro mempunyai masalah yang utama di dalam pemrosesan pembelajaran, yaitu salah satunya pada permasalahan pengelolaan jadwal penggunaan laboratorium komputer di Universitas Dian Nuswantoro kurang tertatanya jadwal penggunaan LAB dan kurang optimalnya penggunaan LAB. Permasalahan tersebut dapat diselesaikan dengan suatu metode guna mendapatkan hasil yang maksimal. Pada penelitian ini guna menyelesaikan suatu masalah dalam mengelola jadwal laboratorium komputer digunakan metode algoritma pewarnaan graf guna merancang jadwal yang nantinya akan di desain dengan tools pemrograman. Sehingga akan diperoleh hasil jadwal yang tertata rapi dan tidak akan terjadi tabrakan jadwal dan jadwal penggunaan LAB lebih optimal.

Kata Kunci: Penjadwalan, Graf, Pewarnaan Graf, Visual Basic 6.0, MySQL

Abstract

Dian Nuswantoro university is private universities in Semarang city excellence in IT the world. Dian Nuswantoro university having among the faculty, including the faculty computer science, the faculty of economics business, health faculty of, of engineering faculty. Dian Nuswantoro university inside having facility that is one of the computer laboratories. On computer laboratories used to lecture lab work of various routes. Across a laboratory facility computers extremely support in the process of learning and teaching lecture on lab work. On computer laboratories there is room lab, the lab, lecture lab work, and its hours. Lecture lab work in computer laboratories Dian Nuswantoro university held on monday to friday. At the Dian Nuswantoro university have a problem focus in in processing learning, namely one of them is problems management schedule the use of computer laboratories in Dian Nuswantoro university less a schedule the use of the lab and a less than optimal the use of the lab. These problems can be completed by a method to get the maximal result. To research this to resolve a problem in managing schedule computer laboratories used method of algorithm staining graf to design a schedule which will be in design with tools

programming. So will the results the schedule orderly and would not on a collision course schedule and schedule the use of the lab optimized.

Keywords: *Scheduling, Graph, Graph Coloring , Visual Basic 6.0, MySQL*

1. PENDAHULUAN

Penjadwalan perkuliahan merupakan sebuah masalah yang perlu di perhatikan dan di pertimbangkan di setiap pembelajaran terutama di Universitas. Karena penjadwalan melibatkan mata kuliah, ruang, dosen, dan mahasiswa. Demikian pula dalam penyusunan dan pengelolaan jadwal penggunaan laboratrium komputer Fakultas Ilmu Komputer di Universitas Dian Nuswantoro, menyesuaikan dengan mata kuliah dan jenis laboratorium komputernya sehingga tepat untuk mata kuliah itu dan jenis laboratorium komputer tersebut. Mata kuliah praktik memang sarana utama dalam proses pengajaran mata kuliah praktik pada Fakultas Ilmu Komputer, karena syarat adanya proses belajar mengajar pada fakultas yaitu harus adanya dosen yang mengajar, matakuliah, jam, ruang serta jurusannya dan mahasiswa.

Masalah penjadwalan penggunaan laboratorium komputer di UDINUS sering terjadi tabrakan jam mata kuliah di laboratorium komputer tersebut. Dari data yang didapat, dari jumlah 11 LAB komputer yang terdiri dari LAB A, LAB B, LAB C, LAB D, LAB E, LAB G, LAB H, LAB I, LAB J, LAB K, dan LAB HW diantaranya 4 LAB yaitu LAB A, LAB E, LAB K, dan LAB J kurang optimal dalam penggunaannya serta jadwal yang tertata kurang rapi masih sering terjadi. Jumlah presentase LAB yang kurang optimal yaitu sejumlah 27.5 % dari keseluruhan LAB yang ada di Fakultas Ilmu Komputer, itu merupakan angka yang cukup banyak. Maka dari itu pengelolaan penjadwalan penggunaan laboratorium komputer Fakultas Ilmu

Komputer sangat di perlukan guna menata jadwal yang tertata dan sesuai dengan jenis ruang laboratorium komputernya dan mengoptimalkan penggunaannya guna proses mengajar pada mata kuliah praktikum. Dalam mengelola dan menetapkan jadwal jam harus teliti dan harus di fikirkan secara tepat, agar tidak terjadi tabrakan jadwal atau jadwal yang sama. Selain itu mata kuliah praktik yang akan memakai ruang laboratorium komputer tidak hanya dari jurusan sistem informasi-SI saja, melainkan dari jurusan Teknik Informatika, jurusan DKV Fakultas Kesehatan, dan Fakultas Ekonomi.

Di dalam pengelolaan jadwal penggunaan laboratorium komputer di Universitas Dian Nuswantoro guna menetapkan jadwal yang baik dan teratur agar tidak terjadi tabrakan jadwal dan penggunaan secara optimal, maka di coba di gunakan metode Algorithma Pewarnaan Graf Welch powell. Didalamnya terdapat mata kuliah yang akan di cocokkan dengan sesuai jenis laboratorium komputernya, selanjutnya di atur jamnya agar tidak terjadi mata kuliah praktik yang tidak bisa menempati laboratorium komputer yang sama diwaktu yang sama dengan kelompok mata kuliah yang sama. Salah satunya cabang teori graf yang biasa digunakan dalam memodelkan permasalahan adalah pewarnaan graf (*graph colouring*). Pewarnaan graf, yaitu dengan memberikan warna pada elemen graf yang akan dijadikan subjek dalam memahami *constraint* permasalahan. Ada tiga macam persoalan pewarnaan graf (*graph colouring*), yaitu pewarnaan titik (*vertex*), pewarnaan sisi (*edge*), dan pewarnaan wilayah (*region*). Salah satu contoh permasalahan yang dapat

dimodelkan dengan pewarnaan graf adalah permasalahan penyusunan jadwal mata kuliah dikampus khususnya di masing-masing jurusan. Untuk sebagai alat bantu, aplikasinya nanti akan di gunakan pemrograman aplikasi Visual Basic 6.0 dengan mysql guna mendukung dalam penyelesaian permasalahan tersebut.

2. KAJIAN TEORI

2.1 Definisi Penjadwalan

Jadwal adalah pembagian waktu berdasarkan rencana pengaturan kerja. Sedangkan penjadwalan adalah proses, cara, perbuatan menjadwalkan atau memasukkan di jadwal [4]. Salah satu tingkatan pendidikan yang mengalami permasalahan penjadwalan adalah perguruan tinggi dalam hal ini penjadwalan perkuliahan. Komponen dari masalah penjadwalan perkuliahan [5] adalah :

1. Kelompok dari mahasiswa
2. Kelompok dari pengajar
3. Kelompok dari mata kuliah
4. Kelompok dari ruangan
5. Kelompok dari slot waktu yang telah ditentukan

Aktivitas perkuliahan merupakan gabungan dari kelompok mahasiswa, dosen yang mengajar, dan mata kuliah yang diajarkan. Permasalahan yang sebenarnya dihadapi dalam kasus ini adalah permasalahan penempatan beberapa aktivitas perkuliahan yang ada ke dalam dimensi slot waktu dan ruangan yang telah ditentukan sehingga diperoleh solusi hasil yang optimal.

2.2 Graf

Teori Graf merupakan salah satu bahasan dalam Matematika Diskrit yang

menarik untuk dibahas karena berkaitan dengan permasalahan yang banyak ditemui di dunia nyata Graf G didefinisikan sebagai pasangan himpunan (V, E) , ditulis dengan notasi $G = (V, E)$, dimana V adalah himpunan tidak kosong dari simpul- simpul (verteks) dan E adalah himpunan isi (Edges) yang menghubungkan sepasang simpul [1].

Berdasarkan arahnya Graf dibedakan menjadi 2 yaitu :

- a. Graf Tak Berarah : Graf yang sisinya tidak mempunyai orientasi arah
- b. Graf Berarah : Graf yang sisinya mempunyai orientasi arah

Pada tulisan ini hanya dibahas Graf Tak Berarah saja.

2.3 Pewarnaan Graf

Dalam teori graf, pewarnaan graf merupakan suatu bentuk pelabelan graf, yaitu dengan memberikan warna pada elemen graf yang akan dijadikan subjek dalam memahami constrain permasalahan. Ada tiga macam persoalan pewarnaan graf (graph colouring), yaitu pewarnaan Simpul, pewarnaan sisi, dan pewarnaan wilayah (region). Pada tulisan ini hanya akan membahas pewarnaan untuk elemen graf yang paling sederhana yaitu pewarnaan simpul graf.

a. Pewarnaan Simpul Graf

Pewarnaan simpul adalah memberi warna pada simpul-simpul di dalam graf sedemikian sehingga setiap dua simpul bertetangga mempunyai warna yang berbeda [6]. Contoh kasus yang merepresentasikan permasalahan ini diantaranya adalah penjadwalan ujian mata kuliah.

b. Derajat Verteks

Derajat dari suatu verteks v dalam graf G , yang dituliskan $\text{deg}(v)$, sama dengan banyaknya sisi dalam G yang memiliki v , artinya yang insiden pada v . Karena setiap sisi terhitung dua kali dalam perhitungan derajat verteks dari G , kita memiliki hasil yang sederhana tapi penting [7].

2.4 Algoritma Welch-Powell

Algoritma Welch-Powell merupakan salah satu pewarnaan algoritma pewarnaan graf yang melakukan pewarnaan berdasarkan derajat tertinggi dari simpul-simpulnya atau disebut Largest Degree Ordering (LOD). Algoritma Welch-Powell dapat digunakan untuk mewarnai sebuah graf G secara efisien. Algoritma ini tidak selalu memberikan jumlah warna minimum yang diperlukan untuk mewarnai G , namun cukup praktis untuk digunakan dalam pewarnaan simpul sebuah graf. Algoritma Welch-Powell hanya cocok digunakan untuk graf dengan orde yang kecil [8].

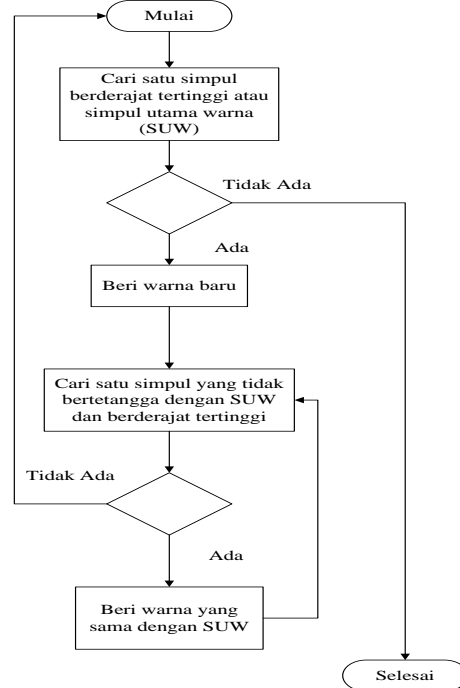
Berikut algoritmanya:

1. Urutkan simpul-simpul dari G dalam derajat yang menurun (urutan seperti ini mungkin tidak unik karena beberapa simpul mungkin berbeda derajat sama).
2. Gunakan satu warna untuk mewarnai simpul pertama (yang mempunyai derajat tertinggi) dan simpul-simpul lain (dalam urutan yang berurutan) yang tidak bertetangga dengan simpul pertama ini.
3. Mulai lagi dengan simpul berderajat tertinggi berikutnya didalam daftar terurut yang belum diwarnai dan ulangi proses pewarnaan simpul

dengan menggunakan warna kedua.

4. Ulangi penggunaan warna-warna sampai semua simpul telah di warnai.

Flowchart Algoritma Welch-Powell adalah sebagai berikut:



Gambar 0.1 Flowchart Algoritma Welch-Powell

2.5 Penerapan Pewarnaan Graf Dalam Penyusunan Jadwal Perkuliahan

Salah satu penerapan dari metode pewarnaan graf dalam kehidupan sehari-hari adalah dalam penyusunan jadwal. Pertama-tama, untuk mengaplikasikan metode pewarnaan graf dalam membuat jadwal ini adalah dengan menggambarkan graf yang menyatakan penjadwalan. Masing-masing pewarnaan memiliki ciri-cirinya masing-masing. Untuk pewarnaan titik pada sebuah G , dua titik yang berbeda yang dihubungkan oleh sebuah sisi harus memiliki warna yang berbeda. Sedangkan untuk pewarnaan sisi pada sebuah G , sisi-sisi dari sebuah titik yang

menghubungkan dengan titik-titik yang lain harus memiliki warna yang berbeda. Untuk memudahkan dalam penyusunan jadwal dengan pewarnaan graf, data-data yang berhubungan dengan penjadwalan diklasifikasikan ke dalam beberapa himpunan. Dalam penyusunan jadwal perkuliahan, komponen-komponen yang harus diperhatikan antara lain :

1. Mata kuliah pada setiap semester
2. Jumlah kelas pada setiap semester
3. Ketersediaan ruang kuliah
4. Waktu perkuliahan
5. Jumlah mata kuliah dalam sehari dari masing-masing kelas
6. Tim pengajar masing-masing mata kuliah

Untuk poin 1 sampai 3 di dalam graf akan dinyatakan dalam bentuk himpunan titik-titik. Sedangkan untuk poin 4, 5 dan 6 di dalam graf akan dinyatakan dalam bentuk sisi.

Metode pewarnaan graf yang cocok untuk penyusunan jadwal perkuliahan adalah pewarnaan sisi. Dari proses metode pewarnaan graf tersebut, dapat dibuat jadwal yang tepat sehingga jadwal dari mata kuliah yang sama dan dosen yang sama yang dibuat tidak pada waktu yang sama/tidak terjadi tumpang tindih jadwal selain itu dapat diatur jumlah maksimal mata kuliah dari masing-masing kelas. Dari sebuah graf yang dibuat dengan pewarnaan sisi, dari warna sisi, dapat diketahui apabila sisi yang berwarna sama, menunjukkan apabila sisi-sisi tersebut dapat dibuat pada waktu yang sama. Namun apabila warna yang digunakan berbeda, maka jadwal tersebut tidak dibuat pada waktu yang sama [2].

2.6 Use Case Modelling

Melalui pemodelan kasus penggunaan (Use-Case Modelling),

kebutuhan dasar aktor diidentifikasi dan bagaimana kebutuhan tersebut akan dipenuhi oleh sistem dijelaskan. Aktor mempresentasikan kelas pengguna dari sistem. Setiap kelas pengguna memiliki alasan yang spesifik mengapa ia menggunakan sistem. Alasan ini juga mewakili suatu goal dari setiap orang pengguna ketika ia menggunakan sistem. Hubungan interaksi antara aktor dan sistem inilah yang kita sebut sebagai kasus penggunaan atau Use Cases. Kasus penggunaan menjelaskan fungsi-fungsi atau layanan-layanan yang harus dilakukan sistem dari perspektif aktor-aktor tertentu.

Kasus-kasus penggunaan tersebut menangkap kebutuhan-kebutuhan dan membantu melacak realisasi dari kebutuhan tersebut selama proses pengembangan perangkat lunak. Jadi sekarang anda tahu ada dua notasi utama dalam pemodelan kasus penggunaan, yaitu : aktor dan kasus penggunaan (Use Case).

a. Aktor

Seorang aktor merupakan suatu peran yang dimainkan oleh sekumpulan objek yang berada diluar sistem yang berinteraksi secara langsung dengan sistem. Suatu objek tunggal boleh memerankan beberapa peran dan karenanya dapat direpresentasikan dengan sejumlah simbol aktor dalam model ini. Instansiasi dari suatu aktor tidak selalu merupakan objek dari manusia, tetapi bisa merupakan objek dari suatu sistem lain diluar sistem yang dikembangkan.

Seorang aktor dapat terlibat secara aktif maupun secara pasif dalam suatu interaksi dengan suatu kasus penggunaan. Ketika seorang aktor terlibat secara aktif, itu berarti aktor yang memicu atau menginisiasi terjadinya kasus penggunaan

tersebut. Sedangkan aktor disebut terlibat secara pasif berarti aktor tersebut bukan bertindak sebagai pemicu atau inisiator, tetapi aktor lain atau sistem itu sendiri. Gambar 2.3 menunjukkan notasi UML untuk merepresentasikan aktor. Aktor digambarkan sebagai orang dan memiliki nama. Nama dari suatu aktor hendaknya merepresentasikan peran dari aktor tersebut.



Gambar 0.2 Notasi grafis untuk aktor

b. Kasus penggunaan (Use Case)

Kasus penggunaan (Use Case) merupakan deskripsi dari interaksi antara aktor dan sistem yang digunakan untuk memenuhi kebutuhan pengguna. Suatu kasus penggunaan merupakan deskripsi langkah-langkah interaksi antara aktor dan sistem untuk mencapai hasil tertentu. Kasus penggunaan juga merefleksikan tujuan atau goal seorang aktor ketika menggunakan sistem.

Gambar 2.3 menunjukkan notasi UML (Unified Modelling Language) untuk merepresentasikan suatu kasus penggunaan. Kasus penggunaan direpresentasikan dengan sebuah elips dan diberi nama. Nama dari suatu kasus penggunaan merepresentasikan fungsionalitas maupun tujuan atau goal yang hendak dicapai ketika interaksi selesai. Biasanya nama kasus penggunaan diungkapkan sebagai suatu kata kerja ataupun kata benda yang dibentuk dari kata kerja (gerund). Deskripsi langkah-langkah interaksi antara aktor dan sistem dibahasakan secara formal maupun tidak formal. Deskripsi ini digunakan lebih untuk media

komunikasi dengan pemangku kepentingan dari sistem dan bertindak sebagai kontrak antar pemangku kepentingan.



Gambar 0.3 Notasi grafis untuk use case

2.7 Deskripsi Use Case

Deskripsi suatu kasus penggunaan mengandung beberapa hal, antara lain :

1. Deskripsi singkat tentang kasus penggunaan tersebut. Bagian ini memberikan gambaran ruang lingkup, masukan, dan luaran yang dihadapkan dari kasus penggunaan tersebut.
2. Mendefinisikan alur normal kejadian dari kasus penggunaan. Bagian ini mendefinisikan bagaimana dan kapan suatu kasus penggunaan dimulai dan berakhir. Bagian ini juga mendefinisikan bagaimana dan kapan kasus penggunaan tersebut akan menyimpan data dalam sistem, atau membutuhkan data dari sistem.
3. Kejadian eksepsional. Bagian ini mendefinisikan setiap kejadian eksepsional yang dapat terjadi beserta langkah-langkah alternatif dari alur normal yang ada.
4. Pre-kondisi. Bagian ini mendefinisikan kondisi awal yang harus terpenuhi sebelum kasus penggunaan ini dimulai.
5. Pra-kondisi. Bagian ini mendefinisikan kondisi akhir yang harus tercapai setelah kasus penggunaan ini berakhir.

Deskripsi dari suatu kasus penggunaan hendaknya berfokus pada sudut pandang aktor, bukan pada ide

atau kebutuhan dari perekayasa. Biasanya deskripsi dari kasus penggunaan ini dinyatakan dalam bahasa alamiah. Suatu kasus penggunaan hendaknya dibuat serinci yang dibutuhkan. Dalam artian, sampai pada tingkat dimana pemangku kepentingan memahami bagaimana sistem nantinya akan digunakan. Dengan demikian kita tahu sekarang bahwa suatu kasus penggunaan hanyalah mendeskripsikan apa yang dipersepsi oleh seorang aktor ketika ia berinteraksi dengan sistem untuk menggunakan suatu fungsionalitas yang hendak disediakan sistem.

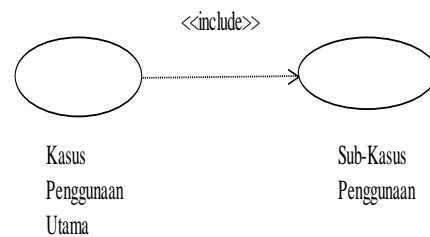
2.8 Relasi dalam Use Case

Suatu kasus penggunaan dapat dihubungkan dengan kasus penggunaan lain melalui relasi kebergantungan, yaitu include, extends, dan alternative atau specialization. Suatu relasi kebergantungan include dari suatu kasus penggunaan A terhadap B mengindikasikan bahwa kasus penggunaan A merujuk dan melibatkan atau mengandung perilaku atau langkah-langkah yang dispesifikasikan pada kasus penggunaan B. Kasus penggunaan B sering juga disebut sub-kasus-penggunaan (Sub Use-Case) A.

Relasi include antara dua kasus penggunaan memodelkan kasus penggunaan utama (Base Use-Case) secara eksplisit melibatkan perilaku dari sub-kasus penggunaan utama. Relasi ini biasa digunakan untuk menghindari pendeskripsian alur kejadian yang sama berulang-ulang, dimana perilaku tersebut muncul pada lebih dari satu kasus penggunaan.

Gambar 2.4 menunjukkan notasi garis relasi include, dimana kasus penggunaan utama melibatkan sub-kasus penggunaan. Relasi ini digambarkan dengan garis putus-putus

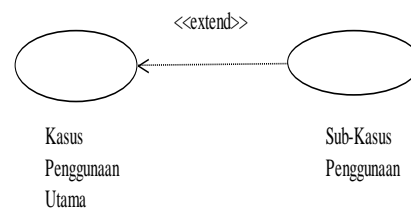
dengan tanda panah mengarah kepada sub-kasus penggunaannya.



Gambar 0.4 Contoh relasi kebergantungan include

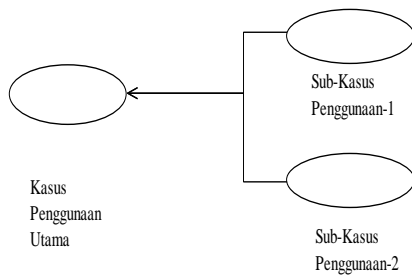
Relasi kebergantungan yang kedua adalah extend. Jika kasus penggunaannya A dinyatakan memiliki relasi kebergantungan extend dengan kasus penggunaan B, berarti kasus penggunaan B dapat diekstensi dengan perilaku yang dispesifikasikan oleh kasus penggunaan A.

Relasi extend antara dua kasus penggunaan memodelkan kasus penggunaan utama (Base Use-Case) yang secara eksplisit diekstensi dengan perilaku dari sub-kasus penggunaan pada lokasi yang ditentukan dalam kasus penggunaan utama. Relasi ini biasa digunakan untuk memisahkan antara alur kejadian utama atau default dengan alur kejadian yang bersifat optional. Gambar 2.5 menunjukkan notasi grafis relasi extend, dimana kasus penggunaan utama mengekstensi sub-kasus penggunaan. Relasi ini digambarkan dengan garis putus-putus dengan tanda panah mengarah kepada kasus penggunaan utamanya.



Gambar 0.5 Contoh relasi kebergantungan extends

Relasi kebergantungan yang ketiga adalah relasi alternative atau specialization. Relasi kebergantungan alternative atau specialization memodelkan suatu kasus penggunaan yang lain. Gambar 2.6 menunjukkan notasi grafis relasi kebergantungan alternative atau specialization. Kasus penggunaan utama memiliki sejumlah alternatif, yaitu sub-kasus penggunaan-1 dan -2.



Gambar 0.6 Contoh relasi kebergantungan alternative atau specialization

2.9 Normalisasi

Normalisasi adalah proses mengelola data dalam database. Ini termasuk membuat tabel dan membangun hubungan antara tabel tersebut berdasarkan aturan yang dirancang untuk melindungi data dan untuk membuat database yang lebih fleksibel dengan menghilangkan redundansi dan dependensi tidak konsisten [10].

Normalisasi bertujuan untuk mengurangi ketidak normalan rancangan tabel yang redundansi atau memiliki struktur atau nilai ganda [11].

Redundansi data membuang ruang disk dan membuat masalah pemeliharaan. Apabila data yang ada di lebih dari satu tempat harus diubah, data harus diubah dengan cara yang sama di semua lokasi. Perubahan alamat pelanggan akan lebih mudah untuk menerapkan apabila data yang disimpan

di tabel pelanggan dan tempat lain di dalam database.

Ada beberapa aturan untuk normalisasi pangkalan data. Setiap aturan yang disebut "bentuk normal." Jika aturan pertama diamati, database dikatakan berupa "pertama normal." Jika aturan tiga diamati, database dianggap berupa "ketiga normal." Meskipun tingkat normalisasi lain mungkin, bentuk normal ketiga dianggap terbaik yang diperlukan untuk sebagian besar aplikasi.

Dengan banyak resmi aturan dan spesifikasi, skenario dunia nyata tidak selalu memungkinkan untuk kepatuhan sempurna. Secara umum, normalisasi memerlukan tabel tambahan dan beberapa pelanggan menemukan ini rumit. Jika Anda memutuskan untuk melanggar salah satu dari tiga aturan normalisasi, pastikan bahwa aplikasi mengantisipasi semua masalah yang dapat terjadi, seperti data yang berlebihan dan dependensi tidak konsisten.

2.10 Tujuan Normalisasi Database

Tujuan normalisasi database adalah untuk menghilangkan dan mengurangi redundansi data dan tujuan yang kedua adalah memastikan dependensi data (Data berada pada tabel yang tepat). Jika data dalam database tersebut belum di normalisasi maka akan terjadi 3 kemungkinan yang akan merugikan sistem secara keseluruhan.

1. INSERT Anomali : Situasi dimana tidak memungkinkan memasukkan beberapa jenis data secara langsung di database.
2. DELETE Anomali: Penghapusan data yang tidak sesuai dengan yang diharapkan, artinya data yang harusnya tidak terhapus mungkin ikut terhapus.

3. UPDATE Anomali: Situasi dimana nilai yang diubah menyebabkan inkonsistensi database, dalam artian data yang diubah tidak sesuai dengan yang diperintahkan atau yang diinginkan.

a. Bentuk Tidak Normal (UnNormalize)

Bentuk tidak normal merupakan bentuk data yang di rekam, tetapi tidak memiliki format tertentu. Data mungkin terduplikasi atau tidak lengkap, tetapi tidak jadi masalah. Pada bentuk tidak normal, kita masih melihat record yang tidak lengkap karena datangnya atribut dengan apa adanya. Apabila dokumen memiliki bentuk lebih kompleks maka kita akan sulit membentuk record. Contoh bentuk tidak normal di gambarkan pada table sebagai berikut :

Tabel 0.1 Tabel Tidak Normal

Student #	Adv isor	Iklan- Ruang	Kls 1	Kls 2	Kls 3
1022	Jones	412	101-07	143-01	159-02
4123	Smith	216	201-01	211-02	214-01

b. Bentuk Normal Kesatu (1 NF)

Sekarang pada bentuk normal kesatu, atribut-atribut yang tepat dan memiliki nilai atomic membentuk suatu table yang flat dan disebut flat table atau flat file. Apabila ada atribut rangkap maka salah satu harus dibuang. Jadi, hanya ada atribut dengan satu arti. Bentuk normal kesatu belumlah normal karena ada atribut yang memiliki

ketergantungan pada atribut yang lain.

Syarat bentuk Normal kesatu

- Buat tabel terpisah untuk menetapkan nilai-nilai yang berlaku untuk beberapa data.
- Tabel ini berhubungan dengan kunci asing.

Data tidak akan tergantung pada apa pun selain Tabel kunci primer (gabungan kunci, jika diperlukan). Sebagai contoh, pertimbangkan alamat pelanggan di sistem akuntansi. Alamat yang dibutuhkan oleh pelanggan tabel, tetapi juga oleh pesanan, pengiriman, faktur, piutang dan koleksi tabel. Alih-alih menyimpan pelanggan alamat sebagai entri terpisah di setiap tabel ini, Simpan di satu tempat, atau tabel pelanggan tabel alamat terpisah. Contoh bentuk normal kesatu di gambarkan pada table sebagai berikut :

Tabel 0.2 Tabel Normal Kesatu

Student #	Advisor	Iklan- Ruang	Kelas #
1022	Jones	412	101-07
1022	Jones	412	143-01
1022	Jones	412	159-02
4123	Smith	216	201-01
4123	Smith	216	211-02
4123	Smith	216	214-01

Tabel harus memiliki dua dimensi. Karena salah satu siswa memiliki beberapa kelas, kelas ini harus terdaftar di tabel terpisah. Kelas 1 kolom, Class2 dan Class3 dalam catatan di atas adalah indikasi desain masalah.

Spreadsheet sering menggunakan dimensi ketiga, tetapi tidak boleh tabel. Cara lain untuk melihat masalah ini dengan satu-ke-banyak hubungan, jangan meletakkan satu

sisi dan banyak dalam tabel yang sama. Sebaliknya, Buat tabel lain dalam bentuk normal pertama dengan menghilangkan grup berulang (kelas #), seperti yang ditunjukkan table diatas.

c. Bentuk Normal Kedua (2NF)

Membuat bentuk Normal Kedua yaitu dengan cara menghilangkan kolom yang tidak tergantung pada kunci. Nilai-nilai dalam dokumen yang bukan merupakan bagian dari kunci data yang tidak termasuk di dalam tabel. Secara umum, saja konten grup kolom mungkin hanya berlaku untuk lebih dari satu dokumen dalam tabel, pertimbangkan untuk menempatkan kolom tersebut dalam tabel terpisah. Misalnya, dalam tabel rekrutmen karyawan, calon Universitas nama dan alamat mungkin disertakan. Tetapi Anda perlu daftar lengkap Universitas untuk grup menyurat. Jika Universitas informasi yang disimpan di dalam tabel calon, ada cara untuk Daftar Universitas dengan saat ini calon. Buat tabel Universitas terpisah dan link ke tabel calon dengan Universitas kode kunci.

PENGECEUALIAN: Mengikuti bentuk ketiga normal, sementara secara teoritis diinginkan, ini tidak selalu praktis. Jika Anda memiliki tabel pelanggan dan Anda ingin menghilangkan semua dependensi interfield mungkin, Anda harus membuat tabel terpisah untuk kota, kode pos, perwakilan penjualan, pelanggan kelas, dan faktor lain yang mungkin diduplikasi di banyak data. Dalam teori, normalisasi bernilai mengerucutkan. Namun, banyak tabel kecil dapat menurunkan kinerja atau melebihi Buka berkas dan kapasitas memori.

Mungkin lebih layak ke bentuk normal ketiga hanya berlaku untuk data yang sering berubah. Jika beberapa bidang tergantung tetap, desain aplikasi Anda untuk meminta pengguna untuk memverifikasi semua terkait kolom apabila salah satu diubah. Perhatikan beberapa kelas # nilai untuk setiap Student # nilai dalam tabel di atas. Kelas # bukanlah fungsional tergantung pada Student # (kunci primer), jadi hubungan ini bukan berupa kedua normal.

Contoh bentuk normal kedua di gambarkan pada table sebagai berikut :

Tabel 0.3 Tabel Normal Kedua

Tabel Siswa :

Student #	Advisor	Iklan-Ruang
1022	Jones	412
4123	Smith	216

Tabel Pendaftaran :

Student #	Kelas #
1022	101-07
1022	143-01
1022	159-02
4123	201-01
4123	211-02
4123	214-01

d. Bentuk Normal Ketiga (3NF)

Pada bentuk normal ketiga, hal yang harus dilakukan yaitu menghilangkan data tidak tergantung pada kunci. Dalam contoh terakhir, iklan-ruang (nomor kantor advisor) fungsional tergantung pada atribut Advisor. Solusinya adalah untuk memindahkan bahwa atribut dari tabel siswa ke tabel fakultas. Contoh bentuk normal ketiga di gambarkan pada table sebagai berikut :

Tabel 0.4 Tabel Normal Ketiga

Tabel Siswa :

Student #	Advisor
1022	Jones
4123	Smith

Tabel Fakultas :

Nama	Ruang	Bagian
Jones	412	42
Smith	216	42

2.11 Pengertian Microsoft Visual Basic

Microsoft Visual Basic adalah sebuah aplikasi yang digunakan untuk pengembangan dengan memanfaatkan keistimewaan konsep-konsep antar muka grafis dalam Microsoft Windows. Aplikasi yang di hasilkan Visual Basic berkaitan erat dengan windows itu sendiri sehingga di butuhkan pengetahuan bagaimana cara kerja windows. Dalam pemrograman Visual banyak istilah dan konsep untuk menyebut sesuatu yang membentuk sebuah aplikasi. Istilah-istilah tersebut memiliki arti yang sama dalam lingkungan pemrograman Visual lainnya, seperti misalnya Objek, Property dan event [12].

2.12 Pemrograman Event Driven

Pemrograman Event Driven telah ada sebelum GUI (Graphical User Interface) atau pemrograman antar muka dibuat dan di implementasikan dalam beberapa cara. Dengan diperkenalkannya mouse, Pemrograman Event Driven lebih banyak disukai oleh para pemakai dan perancang program. Sebelum ada Pemrograman Event Driven, Pemrograman top-down digunakan untuk membuat kode yang menangani banyak pemrosesan. Selain itu aplikasi yang di rancang dengan teknik ini mudah untuk diperbaiki, di

samping kodenya menjadi luwes. Namun aplikasi yang di buat dengan metode ini biasanya memiliki menu yang kompleks yang tergabung erat pada proses yang ada dalam program.

Pemrograman Event Driven tidak mencoba menggantikan pemrograman procedural, namun melengkapi dengan kerangka yang dapat membedakan antara user interface dan proses tertentu dalam aplikasi. Visual Basic dan aplikasi Event Driven lainnya menyediakan kerangka tersebut sehingga programmer lebih terkonsentrasi pada masalah aplikasinya.

a. Pengertian dari objek

Definisi dari sebuah objek adalah suatu yang dapat melakukan hal-hal tertentu. Di dalam visual basic contoh dari objek misalnya, Command Button, Label, List Box, Check Box, dan lain-lain. Objek adalah komponen-komponen yang membuat aplikasi.

b. Pembuatan objek

Pemrograman visual merupakan dimensi baru dalam pembuatan aplikasi karena dapat langsung menggambarkan objek-objek ke layar sebelum di eksekusi. Dalam lingkungan pengembangan visual, sekarang objek yang di buat dapat langsung di tampilkan di layar. Objek yang dibuat tersebut akan sama hasilnya pada saat program di jalankan, dengan demikian tidak perlu lagi melakukan perubahan kode program secara manual. Setelah objek di letakkan dalam form, maka semua atribut objek tersebut akan di simpan sebagai suatu kode program yang dapat langsung di jalankan.

2.13 Prosedure dan fungsi

Pada dasarnya procedure dan fungsi mempunyai tugas yang sama hanya perbedaan terdapat pada apakah prosedur tersebut dapat mengembalikan sebuah nilai atau tidak, jika mengembalikan sebuah nilai di sebut dengan fungsi, sedangkan jika tidak mengembalikan sebuah nilai maka disebut prosedur [13].

a. Prosedur

Sebuah prosedur dapat memiliki satu atau lebih parameter yang tujuannya untuk berhubungan dengan daerah luar prosedur dengan isi dari prosedur tersebut, dimana prosedur tersebut di panggil. Prosedur dibagi menjadi beberapa macam :

- **General Procedure (Prosedur Umum)**

Digunakan untuk member tahu aplikasi bagaimana mengolah pernyataan yang spesifik. Jadi jika semua bagian spesifik dari program telah terdefinisi maka semua bagian spesifik dari program telah terpanggil.

- **Event Procedure (Prosedur Even)**

Setiap form maupun control telah memiliki beberapa event. Jika ingin agar setiap control diatur oleh event maka kita harus menuliskan rutin pada sebuah event yang di sebut Event Procedure.

Contoh :

```
Private sub Form_Load()  
Print "Look at me !!!"  
End sub
```

b. Function Procedure (Prosedur Fungsi)

Prosedur fungsi ini atau fungsi ini untuk mengembalikan sebuah nilai.

Contoh :

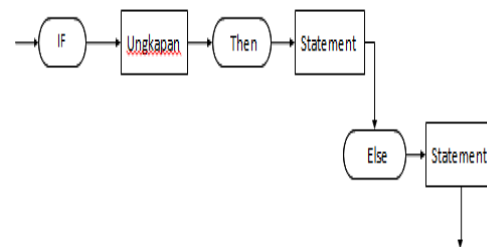
```
Function tengah(f As Form)  
f.Left = (Screen.Width -  
f.Width) / 2  
f.Top = (Screen.Height -  
f.Height) / 2  
End Function
```

2.14 Fungsi Logika

Dalam hamper semua program yang kompleks mengandung suatu penyeleksi kondisi. Dengan menyeleksi suatu kondisi, program dapat menentukan tindakan apa yang harus dikerjakan, tergantung dari hasil kondisi yang di seleksi tersebut. Untuk menyeleksi suatu kondisi, di dalam viusual basic dapat dipergunakan statement if

a. Statement IF

Struktur dari statement if dapat berupa if-Then-Else dengan sintak diagram sebagai berikut :



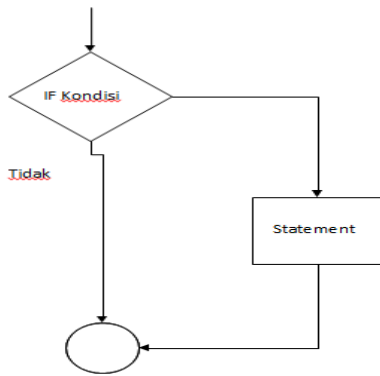
Gambar 0.7 Diagram sintak statement IF

b. Statement IF-Then

Bentuk dari struktur If-Then adalah sebagai berikut:

If Ungkapan Then Statement

Ungkapan adalah kondisi yang akan di seleksi oleh statement If. Bila kondisi yang diseleksi terpenuhi, maka statemen yang mengikuti Then akan dip roses. Sebaliknya, bila kondisi tidak terpenuhi, maka yang akan diproses adalah statement berikutnya.



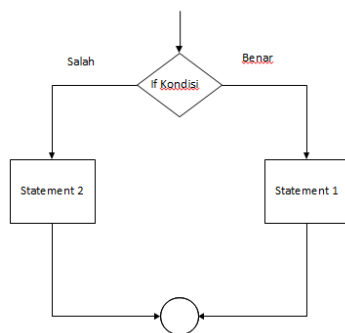
Gambar 2.8 Diagram alir struktur If-Then

c. Statement IF-Then..Else

Struktur If-Then..Else merupakan pengembangan dari struktur If-Then dengan struktur sebagai berikut:

If Kondisi Then
Statement1
Else
Statement2

Statement1 atau dapat juga berupa blok statement akan diproses bilamana kondisi yang diseleksi benar (terpenuhi) sedang statement2 atau berupa blok statement akan diproses bilamana kondisi yang di seleksi tidak terpenuhi. Bila struktur ini digambarkan dalam bentuk bagan alir akan tampak sebagai berikut:



Gambar 0.9 Diagram alir struktur If-Then-Else

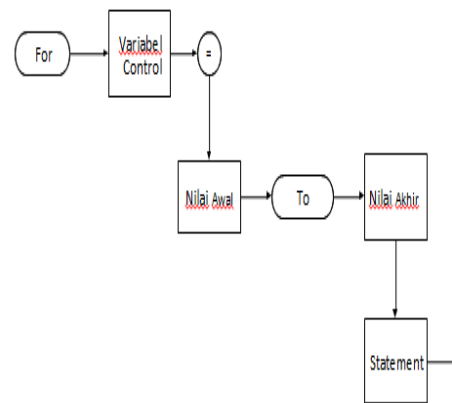
2.15 Perulangan

Salah satu proses yang hampir selalu ada di dalam pemrograman

adalah proses berulang (looping). Proses berulang adalah suatu proses dimana komputer akan mengeksekusi satu atau lebih statemen berulang kali sesuai kebutuhan. Perulangan dengan statemen For digunakan untuk mengulang statemen atau satu blok statemen berulang kali, sejumlah yang ditentukan. Perulangan dengan statemen For dapat berbentuk perulangan :

- Positif
- Tersarang

Struktur perulangan FOR



Gambar 0.10 Diagram sintak statement For

Perulangan Positif adalah perulangan dengan penghitung (counter) dari kecil ke besar atau dengan kata lain pertambahannya positif. Bentuk Umum :

For Variable Kontrol = Nilai Awal To Nilai Akhir

Contoh script perulangan positif :

```
Private Sub Command_Click()
    Dim Entry, I, Msg, Pesan
    'Deklarasi Variable
    Msg = "Tekan OK untuk memasukan 20 Data."
    MsgBox Msg , , "Tekan Ok Bosss..." ' Tampilkan pesan.
    For I = 1 To 20 ' Nilai antara 1 sampai 20.
        Entry = "Data Ke " & I '
        Buat masukan.
```

```
List1.AddItem Entry '
Masukan kedalam ListBox.
Next I
End Sub
```

Perulangan Tersarang adalah perulangan yang berada di dalam perulangan yang lainnya. Perulangan yang lebih dalam akan diproses terlebih dahulu sampai habis, kemudian perulangan yang lebih luar baru akan bertambah, mengerjakan perulangan yang lebih dalam lagi mulai dari nilai awalnya dan seterusnya.

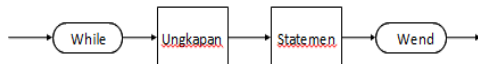
Contoh script perulangan tersarang :

```
Private Sub Command1_Click()
    Dim I, j As Integer
    For i = 1 To 3
        For j = 1 To 4
            List1.AddItem I
            & Space (30) & j
        Next j
    Next i
End Sub
```

2.16 Perulangan While-Wend

Pernyataan While..Wend digunakan untuk mengeksekusi serangkaian perintah selama kondisi yang disyaratkan bernilai true.

Diagram sintak perulangan While,,Wend sebagai berikut :



Gambar 0.11 Diagram Sintak Statemen While..Wend

Bentuk umum:

```
While Ungkapan
    Statement
```

Wend

Statement While-Wend digunakan untuk melakukan proses perulangan suatu statement atau blok statement terus menerus selama kondisi ungkapan logika pada While masih bernilai benar.

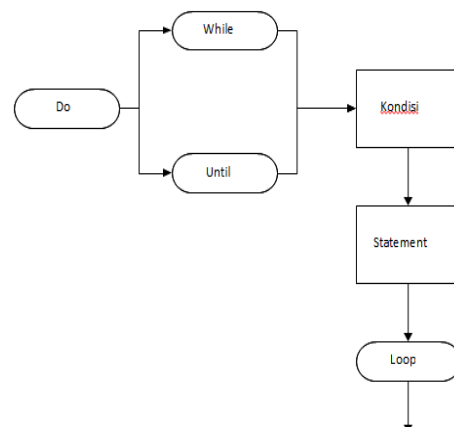
Contoh script:

```
Private Sub Command_Click()
    Dim I, total , jumlah As Integer
    Dim rata As Single
    i = 0
    total = 0
    rata = 0.0
    while i < 10
        total = total + i
        jumlah = jumlah + 1
        i = i + 1
        Label5.Caption = total
        Label6.Caption = jumlah
        Rata = total / jumlah
        Label7.Caption = rata
    Wend
End Sub
```

2.17 Perulangan Do-Loop

Selain pernyataan For..Next untuk melakukan perulangan, Visual Basic juga menyediakan pernyataan Do..Loop.pernyataan ini digunakan untuk mengulangi suatu blok statement selama kondisi yang di syaratkan bernilai true.

Diagram sintak Do-Loop adalah sebagai berikut:



Gambar 0.12 Diagram Sintak Do..Loop

Bentuk umum :

```
Do [{ While|Until } Kondisi]
    [Statement]
[exit do]
[Statement]
```

Loop

Keterangan:

- Kondisi merupakan ekspresi numeric atau ekspresi string yang bernilai true atau False. Bila bernilai Nol dianggap sama dengan False.
- Statement merupakan sebuah atau beberapa baris perintah yang terus menerus dieksekusi selama kondisi bernilai true.

Contoh script Do-Loop:

```
Private Sub Command_Click()
```

```
Const harga = 700
```

```
Dim i As Integer
```

```
Dim hargatotal As Single
```

```
i = 1
```

```
Do
```

```
hargatotal = i * harga
```

```
List1.AddItem i & Space (30) &  
hargatotal
```

```
i = i + 1
```

```
Loop While i <= 10
```

```
End Sub
```

2.18 MySQL

Sebuah perangkat lunak sistem manajemen basis data SQL (Bahasa Inggris: Database Management System) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi diseluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah komersial untuk kasus-kasus dimana penggunaanya tidak cocok dengan penggunaan GPL [14].

Tidak sama dengan proyek-proyek seperti Apache, dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana

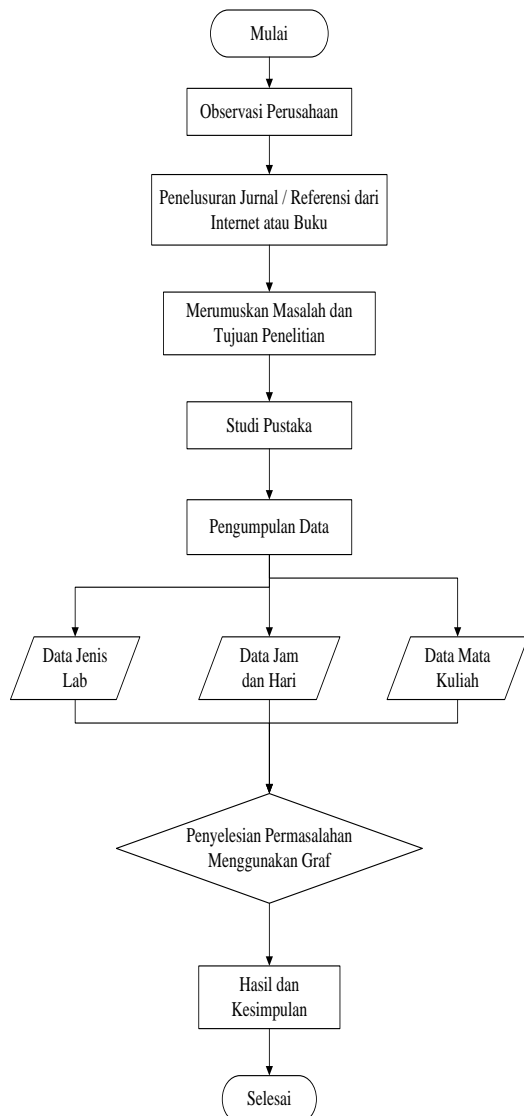
memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius.

MySQL adalah sebuah implementasi dari sistem manajemen basisdata relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya; SQL (Structured Query Language). SQL adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Kehandalan suatu sistem basisdata (DBMS) dapat diketahui dari cara kerja pengoptimasi-nya dalam melakukan proses perintah-perintah SQL yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya. Sebagai peladen basis data, MySQL mendukung operasi basisdata transaksional maupun operasi basisdata non-transaksional. Pada modus operasi non-transaksional, MySQL dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak peladen basisdata kompetitor lainnya.

3. METODE PENELITIAN

Pada sub-bab ini akan menggambarkan tahap-tahap penelitian ini yang akan digambarkan dalam bentuk diagram yaitu sebagai berikut :



Gambar 3.1 Bagan Alir Penelitian

4. HASIL DAN PEMBAHASAN Gambaran Umum Penjadwalan

Penjadwalan merupakan hal yang utama di setiap tempat untuk menuntut ilmu pendidikan terutama di kampus UDINUS Semarang bagaikan Penjadwalan Laboratorium Praktikum Fakultas Ilmu Komputer. Dalam penjadwalan mata kuliah praktikum di perlukan beberapa data yang menunjang diantaranya data jenis laboratorium, data mata pelajaran serta hari dan jam. Oleh karena itu sebagian data tersebut akan di jelaskan di bawah ini :

Deskripsi Laboratorium Komputer FIK

Laboratorium komputer merupakan tempat yang sangat wajib di bidang ilmu komputer, Karena di dalam laboratorium inilah semua mata perkuliahan yang berbaur dengan praktikum komputer akan di adakan di laboratorium tersebut. Berikut ini data-data jenis laboratorium :

Tabel 4.1 Jenis dan fungsi jurusan Lab

Lab	Jenis dan Fungsi
Lab A	Lab Animasi
Lab B	Lab Jaringan
Lab C	Lab FEB
Lab D	Lab Program
Lab E	Lab Animasi
Lab G	Lab Program
Lab H/I	Lab Dasar Pemrograman
Lab J	Lab FEB
Lab K	Lab Program
Lab HW	Lab Jaringan

Deskripsi Hari dan Jadwal Praktikum

Pada tahap ini, di dalam Kampus UDINUS menggunakan 5 hari kerja yaitu hari Senin, Selasa, Rabu, Kamis, dan Jumat. Di dalam satu hari tersebut terdapat dua sifit pelajaran, yaitu sifit pagi dan sifit malam. Perkuliahan di UDINUS dari pukul 07.00 pagi sampai dengan pukul 21.50 malam. Berikut ini rincian waktu pelajaran perkuliahan praktikum di Laboratorium Ilmu Komputer UDINUS :

Tabel 4.2 Jam Praktikum

Sift	Jam
Pagi	07.00 - 08.40
	08.40 - 10.20
	10.20 - 12.00
	12.30 - 14.10
Sore	14.10 - 16.20
	16.20 - 18.00

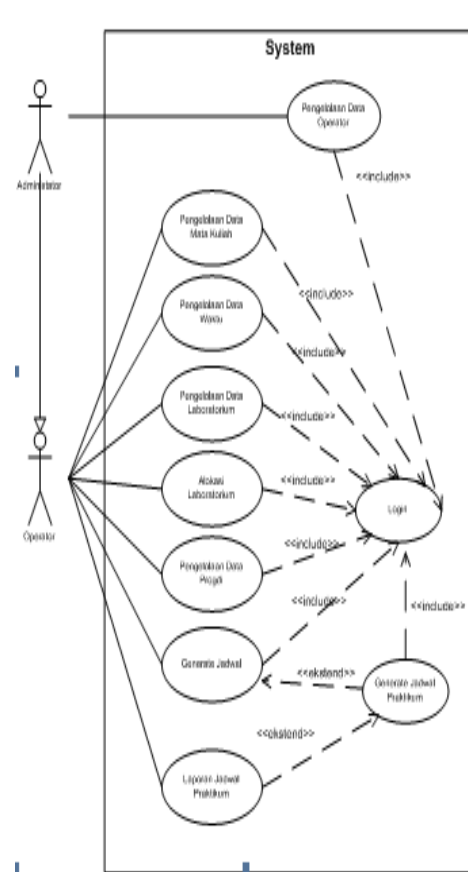
	18.30 - 20.10
	20.10 - 21.50

Alur Diagram

Pada permasalahan bagaimana mengatur jadwal yang baik agar tidak terjadi tabrakan jadwal ada beberapa langkah dalam penyelesaian agar menjadi jadwal yang baik. Dalam langkah menyusun jadwal yang baik terdapat beberapa proses yang diantaranya di gambarkan dalam beberapa diagram. Diagram tersebut berfungsi untuk melihat alur langkah proses dan bagaimana suatu data tersebut dapat bekerja sangat baik.

Use case diagram

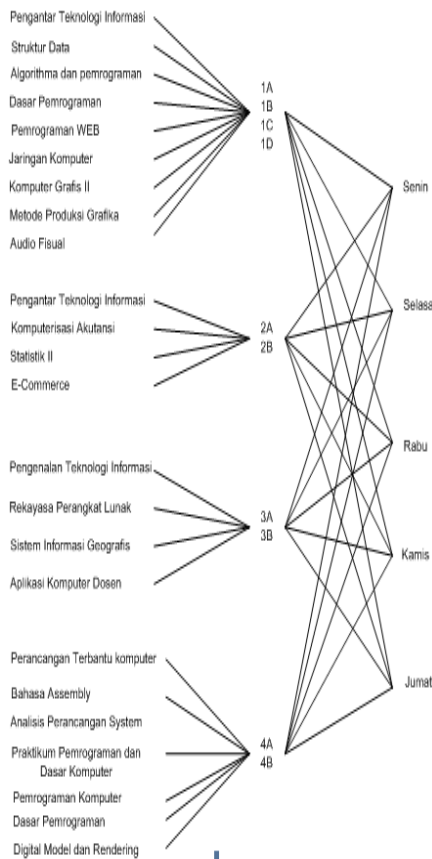
Di dalam pembuatan use case diagram pengelolaan jadwal praktikum laboratorium di UDINUS maka membutuhkan suatu alur langkah proses yaitu membutuhkan suatu proses alur sistem data. Data yang dimiliki adalah : (1) Login, (2) Pengelolaan Data Mata Kuliah Praktikum, (3) Pengelolaan Data Waktu, (4) Pengelolaan Data Laboratorium, (5) Alokasi Laboratorium, (6) Pengelolaan Data Progdi, (7) Laporan Jadwal Praktikum



Gambar 4.1 Use Case Diagram

Graf Penjadwalan Ruang Praktikum

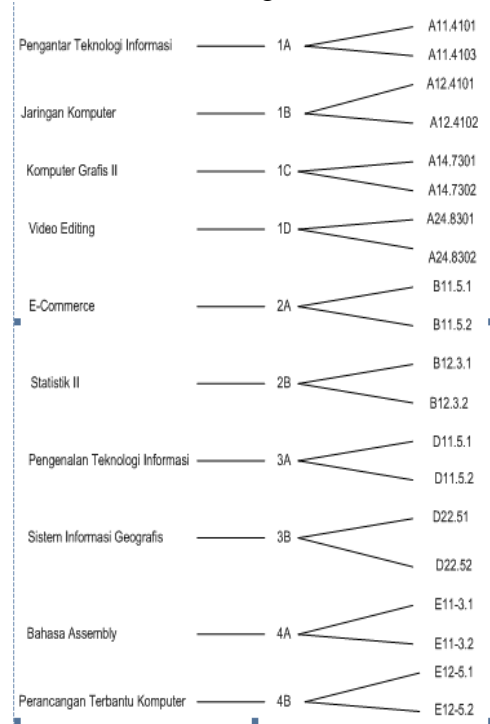
Terbatasnya ruang praktikum laboratorium Fakultas Ilmu Komputer membuat suatu penghalang kendala tersendiri guna menentukan dalam suatu penyusunan jadwal praktikum pada tiap-tiap Progdi. Maka sangat di perlukan penyusunan jadwal praktikum agar tidak terjadi tabrakan pada setiap alokasi Lab dengan jadwal praktikum pada tiap-tiap progdi. Dalam pembuatan teori implementasi graf sebelumnya mempresentasikan komponen-komponen dalam penjadwalan graf[2]. Adapun graf yang menggambarkan banyaknya mata kuliah praktikum, banyaknya Kelompok dan banyaknya hari sebagai berikut :



Gambar 4.2 Graf matakuliah, kelompok, dan waktu

Dari gambar graf di atas terlihat bahwa di dalam setiap pembelajaran praktikum pasti ada matakuliah praktikum, laboratorium praktikum, serta jadwal. Di dalam gambar graf tersebut terlihat juga bahwa di dalam setiap mata kuliah praktikum harus memiliki alokasi ruang laboratorium yang di sediakan untuk di alokasikan pada mata kuliah tersebut. Di dalam setiap mtakuliah serta alokasi ruang praktikum laboratorium komputer untuk hari yang di gunakan yaitu dari hari senin sampai dengan hari jumat. Di dalam kelompok matakuliah, penawaran program studi yang menggunakan laboratorium sebagai praktikum yaitu Fakultas Ilmu Komputer yang terdiri dari Teknik Informatika-S1, Sistem Informasi-S1, Desain Komunikasi Visual, Broadcast, dan Ilmu Komunikasi. Selain itu fakultas lain

yang menggunakan di antaranya Fakultas Ekonomi Bisnis, Fakultas Kesehatan, dan Teknik Industri. Dalam gambar bagan di atas, program studi diatas digambarkan dalam bentuk 1A,1B,1C,1D untuk fakultas Ilmu Komputer, sedangkan untuk Fakultas Ekonomi Bisnis yaitu 2A, dan 2B, untuk Fakultas Kesehatan yang terdiri dari Rekam Medis dan Kesehatan Masyarakat yaitu dengan 3A dan 3B, untuk Fakultas Teknik yaitu diantaranya Teknik Industri dan Teknik Elektro dengan 4A dan 4B.



Gambar 4.3 Graf matakuliah, Progdi, dan Kelompok

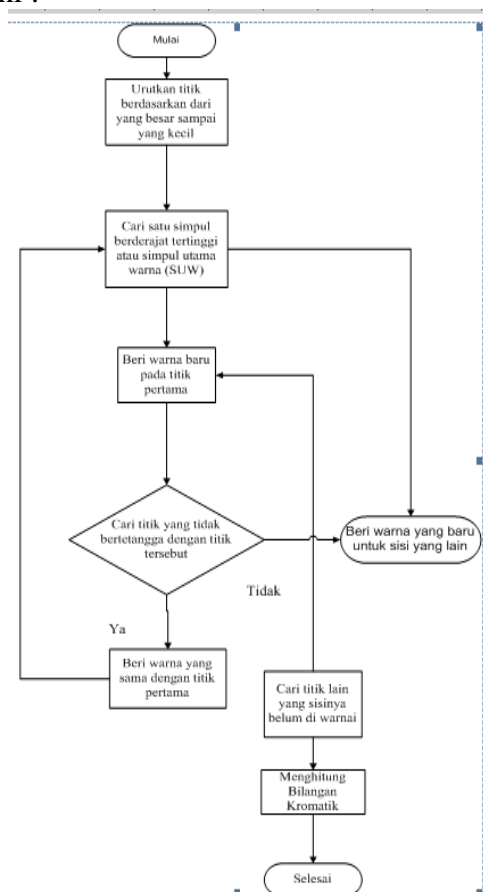
Pada gambar bagan diatas merupakan graf dari perkuliahan praktikum pada setiap progdi. Setiap progdi pasti tidak hanya mempunyai satu kelompok. Pasti lebih dari 1 kelompok pada setiap mata kuliah yang bersangkutan.

Pewarnaan Graf

Di dalam teori pewarnaan graf, sebelum membuat pewarnaan graf terlebih dahulu membuat graf yang

menggambarkan hubungan antara matakuliah praktikum, kelompok, dan ruang laboratorium perharinya.

Pewarnaan yang digunakan pada graf penjadwalan ini adalah pewarnaan sisi. Pewarnaan sisi adalah pewarnaan pada setiap bagian sisi-sisi sehingga setiap sisi-sisi tidak mempunyai warna yang sama[2]. Dari warna sisi-sisi tersebut di cari bilangan kromatiknya. Dalam pewarnaan sisi akan di lihat dari bilangan kromatik dari pewarnaan graf tersebut. Langkah algorithma pengerjaan pewarnaan sisi graf. Algorithma pewarnaan graf di gambarkan di dalam flowchart di bawah ini :

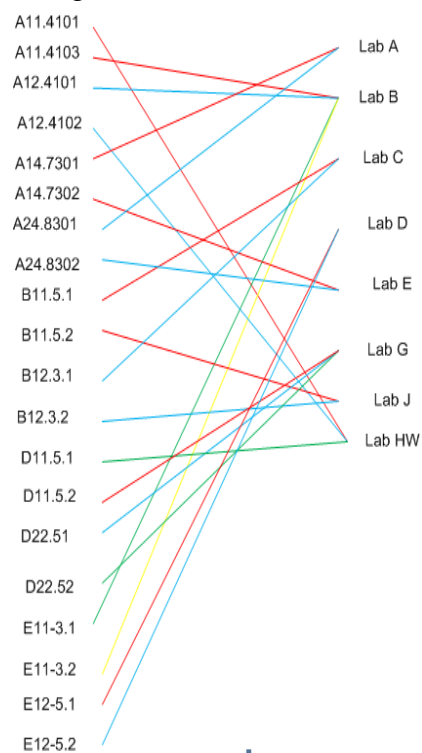


Gambar 4.4 Flowchart Algorithma pewarnaan sisi

Dari sebuah graf yang di buat dengan pewarnaan sisi maka dapat di ketahui dua hal. Yang pertama apabila pewarnaan sisi berwarna sama maka

jadwal dapat dikerjakan dengan waktu yang bersamaan. Sedangkan apabila pewarnaan sisi dengan warna yang berbeda maka jadwal tidak dapat di kerjakan dengan waktu yang bersama. Kemudian yang kedua banyaknya warna merupakan penggambaran jumlah mata kuliah praktikum pada tiap harinya.

Hal yang perlu di buat dalam pewarnaan graf yaitu waktu perkuliahan yang terjadi di dalam ruangan laboratorium. Jadi pewarnaan graf yang akan di buat yaitu pewarnaan graf antara kelompok dan ruang laboratorium. Berikut ini adalah pewarnaan graf antara kelompok dan ruang:



Gambar 4.5 Pewarnaan Graf antara kelompok dan Ruang Laboratorium

Berdasarkan gambar pewarnaan graf di atas dapat dilihat graf bisa di warnai dengan minimal 4 warna sehingga bilangan kromatiknya $X(G) = 4$. Warna yang berbeda menyatakan bahwa pelaksanaan waktu praktikum di laboratorium berbeda-beda. jadi pada

hari senin bisa di jadikan 2-4 jadwal pada tiap ruang praktikum laboratorium

Desain tabel

Di dalam implementasi sebuah perangkat lunak pasti ada suatu rancangan design databases. Di dalam implementasi pada permasalahan penjadwalan sebelum membuat aplikasi pendukung, yang harus di buat terlebih dahulu adalah konsep rancangan database. Di dalam aplikasi penjadwalan komponen-komponen database harus di susun dengan langkah proses normalisasi.

Proses normalisasi bertujuan untuk mengurangi ketidak normalan rancangan table yang redundansi atau memiliki struktur atau nilai ganda[11]. Normalisasi dapat di kerjakan sampai dengan langkah bentuk normal ketiga. Berikut ini adalah bentuk desain table database pada implementasi aplikasi pendukung penjadwalan.

Penjadwalan
kdJadwal
kdRuang
nmRuang
kdSesi
nmSesi
stsKelas
kdMatkul
nmMatkul
semester
Hari
kdkelas
kdProgdi
nmKelas
stsKelas
semesterke

Gambar 4.6 Desain Database

Bentuk desain table di atas merupakan bentuk tabel yang tidak normalisasi, karena atribut-atribut di atas banyak perulangan atau redundansi,

sehingga kata lain bentuk desain tabel diatas yaitu bentuk tidak normal (UnNormalize). Pada bentuk tidak normal, kita masih melihat record yang tidak lengkap karena datangnya atribut dengan apa adanya.apabila dokumen memiliki bentuk lebih kompleks maka kita akan sulit membentuk record. Untuk menjadikan tabel yang sempurna, atribut-atribut desain tabel di atas perlu di pecah lagi dijadikan ke bentuk tabel normal. Didalam langkah normalisasi database, bisa di buat dalam bentuk normal kesatu (1NF), bentuk normal kedua (2NF), dan yang terakhir bentuk normal ketiga (3NF).

Bentuk Normal kesatu

Pada bentuk normal kesatu merupakan uraian dan pemecahan dari bentuk tidak normal. Pada tahapan normal kesatu, atribut-atribut yang tepat dan memiliki nilai atomic membentuk suatu tabel yang flat yang disebut flat file. Pada desain tabel penjadwalan perkuliahan praktikum, berikut ini merupakan bentuk dari desain tabel bentuk normal kesatu:

Penjadwalan
kdJadwal*
kdRuang*
nmRuang
kdSesi*
nmSesi
kdMatkul*
nmMatkul
kdProgdi*
Hari
kdKelas*
nmKelas
stsKelas
Semesterke

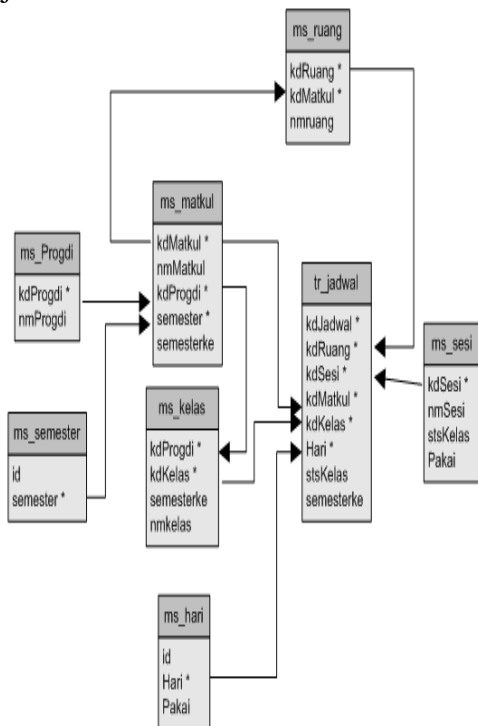
Gambar 4.7 Bentuk Normal Kesatu

Simbol bintang pada masing-masing field kdJadwal, kdRuang, kdSesi, kdMatkul, kdProgdi, kdKelas

menunjukkan bahwa field tersebut merupakan kunci kandidat. Dari bentuk normal kesatu tersebut, masih terdapat kelemahan dan kekurangan yaitu masih terdapat field yang masih redundansi, serta di dalam aplikasi masih kesulitan untuk membentuk record. Apabila bentuk normal kesatu digunakan sebagai transaksi di suatu aplikasi, belum bisa berjalan sesuai harapan. Oleh karena itu harus dilanjutkan ke langkah selanjutnya yaitu ke langkah bentuk normalisasi kedua.

Bentuk Normal Kedua

Pada tahapan langkah ini, semua atribut yang non kunci tidak ada yg tergantung dengan atribut kunci. Pada langkah ini, komponen dari bentuk normal kesatu di pecah menjadi beberapa tabel yang di dalamnya tidak mengandung perulangan data. Berikut ini adalah bentuk dari tabel normal kedua pada permasalahan perencanaan jadwal:

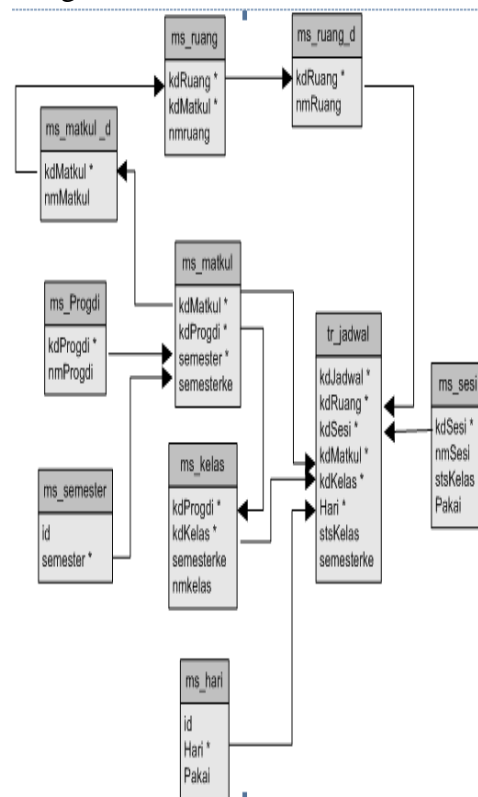


Gambar 4.8 Bentuk Normal Kedua

Di dalam bentuk normal kedua, sudah tidak ada redundansi data tetapi bila pada tahapan normal kedua di implementasikan pada aplikasi pendukung penjadwalan, akan terjadi tumpuknya data pada tabel ms_ruang, dan ms_matkul. Oleh karena itu pada tahapan selanjutnya kedua tabel tersebut di pecah menjadi ke bentuk normal ketiga.

Bentuk Normal Ketiga

Didalam tahapan bentuk normal ketiga meliki syarat, yaitu setiap tabel tidak boleh memiliki atribut yang tergantung transitif, tetapi harus bergantung penuh pada kunci utamanya. Berikut ini adalah bentuk dari normal ketiga:



Gambar 4.9 Bentuk Normal Ketiga

Pada bentuk normal ketiga sudah tidak terjadi proses data rangkap bila sudah di implementasikan pada aplikasi pendukung. Bentuk normal ketiga adalah bentuk desain tabel yang

paling sempurna dan yang paling rinci agar tidak terjadi suatu tumpukan data yg akan mengakibatkan redundansi data. Pada tabel detail yang di buat didalam bentuk normalisasi menjelaskan data spesifik dari sebuah transaksi tersebut, bila didalam proses tersebut dijadikan satu maka akan terjadi suatu proses pencetakan data yang prosesnya berulang. Dengan demikian desain yang dibuat akan menghasilkan data yang maksimal dan tidak mengalami perulangan data yang tidak perlu.

Desain Input Output

Didalam permasalahan pengelolaan jadwal Lab Komputer dibutuhkan tools guna pelengkap dalam penyelesaian permasalahan tersebut. Pada permasalahan ini, digunakan tools dengan pemrograman visual basic yang menggambarkan bagaimana pengelolaan jadwal lab praktikum di UDINUS. Berikut ini adalah desain dari form utama:



Gambar 4.10 Desain Form Utama

Pada tampilan design menu utama terdapat beberapa tahapan dalam proses implementasi sistem. Diantaranya

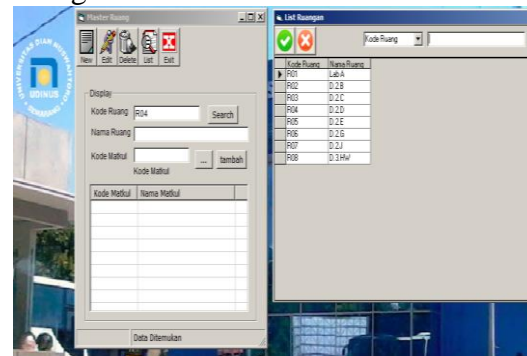
terdapat desain input, desain proses, lalu desain output.

Desain Input

Pada komponen master input terdapat beberapa file diantaranya master ruang, master sesi, master mata kuliah, dan master kelompok. Master master tersebut berupa data inputan yang nantinya akan di proses.

1. Master Ruang

Pada master ruang merupakan penjelasan dan inputan dari jenis lab tersebut yang nantinya tiap mata kuliah akan di plotting di lab-lab tersebut. Berikut ini adalah desain dari master ruang:



Gambar 4.11 Desain Master Ruang

2. Master Sesi

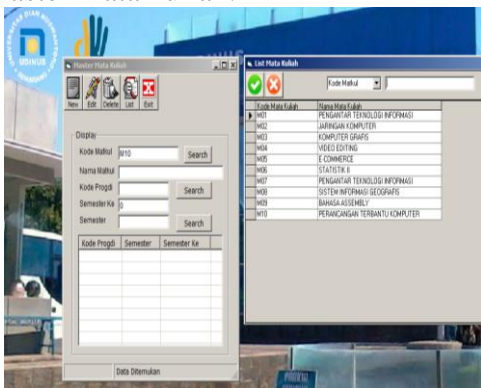
Pada komponen selanjutnya yaitu terdapat file master sesi. Pada master sesi merupakan data inputan untuk sesi yang berisi kode sesi, nama sesi, status kelas dan pilihan di pakai atau tidaknya. Pada setiap perkuliahan pasti terdapat sesi waktu perkuliahan yang menjadi kunci waktu. Berikut ini adalah desain master sesi:



Gambar 4.12 Desain Master Sesi

3. Master mata kuliah

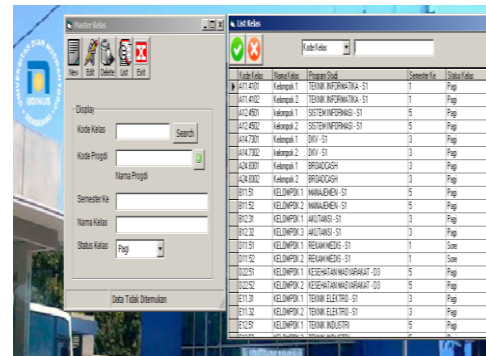
Pada master mata kuliah terdapat beberapa komponen field diantaranya terdapat kode mata kuliah, nama mata kuliah, kode progde, dan semester ke berapa. Pada master tersebut sebagai kunci yang nantinya untuk generate jadwal. Berikut ini adalah desain dari master mata kuliah:



Gambar 4.13 Desain Master Mata Kuliah

4. Master Kelompok

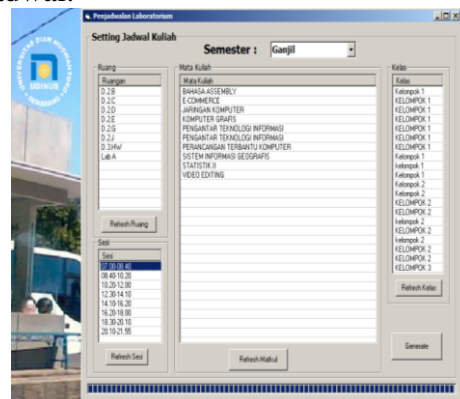
Pada master kelompok menunjukkan tentang data kelompok yang didalamnya terdapat progdi, semester ke berapa, status kelas apakah pagi atau sore. Di dalam perkuliahan di UDINUS terutama untuk praktikum pasti terdapat kelas pagi dan sore dari beberapa progdi. Berikut ini adalah desain dari master kelompok:



Gambar 4.14 Desain Master Mata Kuliah

Desain Proses

Pada proses generate jadwal. Dari semua master tersebut akan di generate menjadi sebuah jadwal praktikum yang akan di plot pada tiap ruang lab dari hari senin sampai jumat serta beberapa sesi yang di pakai. Tampilan generate jadwal ini adalah desain dari alur penjadwalan sesuai dengan jenis lab nya. Di dalam generate jadwal memploting mata kuliah, di gabungkan dan di cocokkan dengan ruang, sesi, serta kelompok yang akan membentuk jadwal praktikum. Berikut ini adalah desain dari proses generate jadwal:



Gambar 4.15 Desain Proses Generate Jadwal

Desain Output

Berikut ini adalah desain output dari sistem penjadwalan:

Hari	Jam	Nama Kuliah	Ruang
Senin	07.00-08.00	PENGANTAR TEKNOLOGI INFORMASI	012.402
	08.40-09.30	PENGANTAR TEKNOLOGI INFORMASI	012.403
	10.30-11.30	PENGANTAR TEKNOLOGI INFORMASI	012.404
	12.30-13.30	PENGANTAR TEKNOLOGI INFORMASI	012.405
	14.30-15.30	PENGANTAR TEKNOLOGI INFORMASI	012.406
Selasa	07.00-08.00	PENGANTAR TEKNOLOGI INFORMASI	012.407
	08.40-09.30	PENGANTAR TEKNOLOGI INFORMASI	012.408
	10.30-11.30	PENGANTAR TEKNOLOGI INFORMASI	012.409
	12.30-13.30	PENGANTAR TEKNOLOGI INFORMASI	012.410
	14.30-15.30	PENGANTAR TEKNOLOGI INFORMASI	012.411
Rabu	07.00-08.00	PENGANTAR TEKNOLOGI INFORMASI	012.412
	08.40-09.30	PENGANTAR TEKNOLOGI INFORMASI	012.413
	10.30-11.30	PENGANTAR TEKNOLOGI INFORMASI	012.414
	12.30-13.30	PENGANTAR TEKNOLOGI INFORMASI	012.415
	14.30-15.30	PENGANTAR TEKNOLOGI INFORMASI	012.416
Kamis	07.00-08.00	PENGANTAR TEKNOLOGI INFORMASI	012.417
	08.40-09.30	PENGANTAR TEKNOLOGI INFORMASI	012.418
	10.30-11.30	PENGANTAR TEKNOLOGI INFORMASI	012.419
	12.30-13.30	PENGANTAR TEKNOLOGI INFORMASI	012.420
	14.30-15.30	PENGANTAR TEKNOLOGI INFORMASI	012.421
Jumat	07.00-08.00	PENGANTAR TEKNOLOGI INFORMASI	012.422
	08.40-09.30	PENGANTAR TEKNOLOGI INFORMASI	012.423
	10.30-11.30	PENGANTAR TEKNOLOGI INFORMASI	012.424
	12.30-13.30	PENGANTAR TEKNOLOGI INFORMASI	012.425
	14.30-15.30	PENGANTAR TEKNOLOGI INFORMASI	012.426

Gambar 4.16 Desain Output

Di ambil sampel pada lab D.3.HW dapat di lihat bahwa untuk di pakai matakuliah Pengantar Teknologi Informasi jurusan teknik informatika – S1, dipakai juga untuk PTI sistem Informasi – S1, dan di pakai untuk FKes. Itu tadi desain output dari contoh sistem pengelolaan jadwal LAB. Dengan demikian jadwal tersusun rapi dan meminimalisir terjadi tabrakan jadwal tersebut, serta memberikan penggunaan LAB yang optimal.

5. PENUTUP

Dari hasil pembahasan implementasi teori graf penjadwalan antara banyaknya mata kuliah, banyaknya kelompok, dan banyaknya hari terlihat bahwa untuk setiap tingkatan semester memiliki jadwal perkuliahan antara hari senin sampai jumat. Dapat disimpulkan bahwa teori pewarnaan graf dapat digunakan untuk meminimalisir tabrakan atau tumpang tindih jadwal di dalam alokasi ruang praktikum di laboratorium dan memberikan penggunaan alokasi ruang laboratorium yang optimal

DAFTAR PUSTAKA

- [1]. Astuti, Setia. (Januari 2011). Penyusunan Jadwal Ujian dengan Algorithma Pewarnaan Graf Welch Powell. *Jurnal Dian Vol. 11 No 1*.
- [2]. Yahya, Nisky Imansyah, dkk. (2013). Penerapan Konsep Graf dalam Penyusunan Jadwal Perkuliahan DiJurusan Pendidikan FMIPA UNG. *Jurnal Dian Vol. 1 No.1*.
- [3]. Dewi, Findra Kartika Sari. (Januari 2010). Pembangunan Perangkat Lunak Pembangkit Jadwal Kuliah dan Ujian dengan Metode Pewarnaan Graf. *Jurnal Buana Informatika Vol.1 No.1*.
- [4]. Depdikbud. (1991). *Kamus Besar Bahasa Indonesia*. Jakarta: Balai Pustaka.
- [5]. Syadid, M. (2005). *Penjadwalan Perkuliahan Menggunakan Algoritme Genetika*. Bogor: Institut Pertanian Bogor.
- [6]. Munir, Renaldi. (2005). *Matematika Diskrit*. Bandung: Informatika.

- [7]. Lipschutz, Seymour, Marc Lipson. (2008). *Matematika Diskret Edisi 3*. Jakarta: Erlangga.
- [8]. Budiman, Hengky. *Penerapan Graph Colouring untuk merencanakan Jadwal* [online]. Tersedia: <http://www.informatika.org/~rinaldi/Matdis/2007/2008/Makalah/MakalahIF2153-0708-025.pdf>, Diakses tanggal 18 maret 2014
- [9]. Siahaan, Daniel. (2012). *Analisa Kebutuhan Dalam Rekayasa Perangkat Lunak*. Yogyakarta: Andi.
- [10]. Microsoft Support. (03 Agustus 2013). *Normalisasi* [online]. tersedia : <http://support.microsoft.com/kb/283878.id-id>. Diakses tanggal 6 Mei 2014.
- [11]. Frieyade. (2010). *Mudah Belajar Pemrograman Database MySQL dengan Microsoft Visual Basic 6.0*. Yogyakarta: Andi.
- [12]. Suryana, Taryana. (2009). *Visual Basic*. Yogyakarta: Graha Ilmu.
- [13]. Mckelvy, Mike. (2000). *Special Edition Using Visual Basic. QUE*.
- [14]. Kadir, Abdul. (2008). *Tuntutan Praktis : Belajar Database Menggunakan SQL*. Yogyakarta: Andi.