

# Generator Jadwal Mengajar Dosen Dengan Perkecualian Memakai Teori Himpunan

Katon Wijana<sup>1</sup>, Yetli Oslan<sup>2</sup>, Harianto Kristanto<sup>3</sup>

Program Studi Sistem Informasi, Universitas Kristen Duta Wacana  
Jl. Dr. Wahidin Sudirohusodo 5 - 25, Yogyakarta

<sup>1</sup>katony@staff.ukdw.ac.id

<sup>2</sup>yetli@staff.ukdw.ac.id

<sup>3</sup>harianto@staff.ukdw.ac.id

*Abstract— Lecturers have the task of "tri darma perguruan tinggi", namely teaching, dedication and research, therefore lecturers' activities are not solely teaching in class but also carry out other activities related to this matter. In this case the head of the department, as a compiler of the class schedule, only knows the course schedules that are taught by the lecturer concerned and does not know other activities carried out by the lecturer. In this case the author wants to use set theory to create a schedule generator that can exclude certain schedule lecturers on certain schedules.*

*The set theory can classify which parts are subsets or non-subsets, can also identify intersection of several sets and can combine one set with another set. The activity schedule basically can be categorized as a set that can be operated using these theories, so it is possible to identify which schedules are colliding (intersecting) and which schedules are still available (not the subsets) for certain lecturers.*

*By implementing set theory on the lecturer teaching schedule database plus the exceptions, it is expected that the lecturer teaching schedule will be obtained in accordance with the teaching schedule and the schedule of each lecturer's activities..*

**Intisari—** Dosen mempunyai tugas tri darma perguruan tinggi, yaitu pengajaran, pengabdian dan penelitian, oleh sebab itu kegiatan dosen tidak semata-mata mengajar di kelas melainkan juga melakukan kegiatan-kegiatan lain sehubungan dengan hal tersebut. Dalam hal ini ketua program studi, sebagai penyusun jadwal kuliah, hanya mengetahui mata-kuliah-matakuliah yang diampu oleh dosen yang bersangkutan dan tidak mengetahui kegiatan lain yang dilakukan oleh dosen tersebut. Pada kasus inilah penulis ingin memakai teori himpunan untuk membuat generator jadwal yang bisa menge-cualikan dosen tertentu pada jadwal tertentu.

Teori Himpunan dapat mengelompokkan mana yang merupakan himpunan bagiannya atau bukan himpunan bagiannya, dapat juga mengidentifikasi irisan dari beberapa himpunan serta dapat menggabungkan antara satu himpunan dengan himpunan yang lain. Jadwal kegiatan pada dasarnya dapat dikategorikan sebagai himpunan yang dapat di-operasikan memakai teori-teori tersebut, sehingga sangat memungkinkan untuk mengidentifikasi jadwal mana yang bertabrakan (beririsan) dan jadwal mana yang masih kosong (bukan himpunan bagiannya) untuk dosen tertentu.

Dengan mengimplementasikan teori himpunan pada basis data jadwal mengajar dosen ditambah dengan pengecualiannya, diharapkan akan diperoleh hasil jadwal mengajar dosen yang sesuai dengan jadwal pengajaran maupun dengan jadwal kegiatan dosen masing-masing.

**Kata Kunci—** Jadwal mengajar, generator jadwal, pengecualian, teori himpunan

## I. PENDAHULUAN

### 1.1 Latar Belakang

Pembuatan jadwal kuliah untuk persiapan registrasi harus memperhatikan batasan dari beberapa entitas yang masing-masing mempunyai keterbatasan untuk dibuat jadwalnya, antara lain : matakuliah, grup, semester, ruang kuliah dan dosen. Mengenai generator jadwal kuliah sudah dilakukan penelitiannya oleh peneliti beberapa waktu yang lalu, namun pada waktu itu belum memperhatikan konstrain perkecualian, yaitu jadwal kegiatan lain dari dosen yang bersangkutan. Pada kenyataannya dosen belum tentu dapat memenuhi permintaan jadwal yang dibuat dengan generator terdahulu karena dosen mempunyai kegiatan-kegiatan lain selain tugas pengajaran, yaitu penelitian dan pengabdian kepada masyarakat sesuai dengan tri dharma perguruan tinggi.

Untuk mengatasi masalah jadwal yang tumpang tindih dengan jadwal pribadi dosen diperlukan suatu metode untuk mengecualikan jadwal-jadwal tertentu yang merupakan jadwal pribadi tersebut, sehingga jadwal yang dihasilkan benar-benar dapat diterapkan secara nyata.

Teori himpunan merupakan model yang dapat digunakan untuk membantu memecahkan masalah penjadwalan mengajar dosen dengan pengecualian ini, dengan memandang tiap entitas yang terkait mempunyai anggota jadwal-jadwal yang dapat direlasikan dengan operasi-operasi himpunan.

Untuk dapat mewujudkan jadwal dosen tersebut, penulis menggunakan himpunan semua jadwal yang mungkin atau yang dalam teori himpunan dikenal dengan istilah universal (U), dan semua entitas yang terkait harus mempunyai himpunan jadwal yang merupakan himpunan

bagian (interseksi) dari himpunan jadwal universal tersebut.

Pada penelitian ini, dipilih studi kasus pembuatan jadwal mengajar dosen dengan pengecualian di program studi Sistem Informasi Universitas Kristen Duta Wacana Yogyakarta, dengan berfokus pada pokok bahasan yaitu untuk membuat jadwal mengajar dosen yang memperhatikan pengecualian, yaitu jadwal-jadwal pribadi dosen.

Langkah-langkah yang dilakukan oleh penulis yaitu dengan merancang database sehingga didapat skema diagram, yang teknik perancangannya tidak dibahas dalam tulisan ini, mengingat pokok bahasan pada tulisan ini adalah penerapan teori himpunan untuk mengatasi masalah pembuatan jadwal mengajar dosen dengan pengecualian. Kemudian implementasi teori Himpunan untuk membuat jadwal mengajar bagi dosen.

### 1.2 Perumusan Masalah

Berdasarkan uraian tersebut di atas, peneliti akan memanfaatkan sebagian dari program yang telah dibuat pada penelitian sebelumnya, yaitu penelitian dengan judul "Generator Jadwal Kuliah dengan Teknik Pencarian dan Penyaringan" yang juga merupakan hasil penelitian oleh peneliti, kemudian menambahkan pengecualian untuk jadwal-jadwal pribadi dosen dan mengoperasikannya dengan teori himpunan sehingga diperoleh jadwal mengajar dosen yang sudah memperhatikan pengecualian tersebut.

### 1.3 Batasan Masalah

Penelitian ini hanya akan membahas tentang penyelesaian masalah pembuatan jadwal mengajar dosen dengan pengecualian di program studi Sistem Informasi Universitas Kristen Duta Wacana Yogyakarta. Dalam penelitian ini akan diambil data jadwal kuliah di program studi Sistem Informasi UKDW.

### 1.4 Kontribusi Penelitian

Penelitian ini akan memberikan kontribusi pada Universitas Kristen Duta Wacana Yogyakarta, khususnya untuk meningkatkan sistem informasi akademik berbasis komputer agar proses registrasi dapat berjalan lancar pada umumnya dan secara khusus dapat membantu ketua program studi dalam menyusun jadwal kuliah di program studi Sistem Informasi Universitas Kristen Duta Wacana Yogyakarta.

## II. REFERENSI

### 2.1 Perancangan Basis Data

Perancangan basis data merupakan aktivitas yang berfokus pada desain basis data yang akan digunakan untuk menyimpan dan mengatur data pemakai. Sebuah basis data yang memenuhi kebutuhan semua pemakai bukan hanya diperlukan tetapi strukturnya harus dirancang dengan hati-hati.

Data merupakan salah satu aset yang paling berharga, data konsumen, pemasok, karyawan, pemesanan dan transaksi semuanya merupakan bagian yang sangat vital untuk keberadaan sebuah perusahaan. Dengan desain struktur data yang baik, kita akan dapat menggunakan perintah Query SQL sederhana untuk mendapatkan informasi yang dibutuhkan. [1]

### 2.2 Structure Query Language (SQL)

Perintah SELECT menentukan kolom-kolom data yang akan ditampilkan, kriteria data yang diinginkan, dan urutan data yang dihasilkan. Selanjutnya perintah SELECT dapat dikelompokkan maupun digabungkan dengan tabel lain. Komponen perintah SELECT adalah seperti yang dijelaskan pada Tabel 1. di bawah ini:

Tabel I

Komponen Perintah SQL SELECT

Komponen	Keterangan
SELECT	Menentukan kolom-kolom yang dimaksud
FROM	Menentukan nama tabel yang dimaksud
WHERE	Menentukan kriteria data yang dimaksud
GROUP BY	Untuk mengelompokkan baris data
HAVING	Untuk menentukan kriteria query agregasi
ORDER BY	Untuk mengurutkan baris berdasarkan kolom

Perintah INSERT digunakan untuk menambah baris data pada tabel. Pada perintah INSERT ditentukan tabel mana yang akan ditambah barisnya, kolom apa saja yang hendak diberi nilai dan dari mana sumber data diperoleh. Komponen perintah INSERT adalah seperti yang dijelaskan pada Tabel 2. di bawah ini:

TABEL II.

Komponen Perintah SQL INSERT

Komponen	Keterangan
INSERT INTO	Menentukan tabel yang akan ditambah baris
daftar kolom	Menentukan nama-nama kolom yang akan diberi nilai
WHERE	Menentukan kriteria data yang dimaksud
VALUES	Menentukan nilai yang hendak diberikan
SELECT	Menentukan tabel sumber data yang hendak ditambahkan.

Perintah DELETE memungkinkan kita untuk menghapus baris data. Komponen perintah DELETE adalah seperti yang dijelaskan pada Tabel 3. di bawah ini: [2]

TABEL III

Komponen Perintah SQL DELETE

Komponen	Keterangan
DELETE FROM	Menentukan tabel yang akan dihapus
WHERE	Menentukan kriteria data yang dimaksud

### 2.3 Stored Procedure

Pada SQL Server, prosedur merupakan program yang tersimpan dalam database server yang dapat diberikan parameternya. Berbeda dengan fungsi (function), prosedur tidak mengembalikan nilai, Namun, prosedur dapat mengembalikan status keberhasilan/ kegagalan ke prosedur yang memanggilnya.

Sebagai contoh misalnya menggunakan tabel **Authors** di dalam database **Library** dapat ditulis sebagai berikut :

```
CREATE PROCEDURE GetName
(
    @input_id INT = NULL, --Input
    parameter, id of the person, NULL
    default
    @name VARCHAR(128) = NULL --Input
    parameter, name of the person, NULL
    default
)
AS
BEGIN
    SELECT Name + ' is from ' + Country
    FROM Authors
    WHERE Id = @input_id OR Name =
    @name
END
GO
```

Untuk menjalankan prosedur tersebut di atas perlu dilakukan eksekusi dengan beberapa cara penulisan perintah. Pertama, EXECUTE atau EXEC seperti berikut :

```
EXECUTE GetName @id = 1
EXEC Getname @name = 'Ernest Hemingway'
```

Selain cara tersebut di atas, perintah EXEC dapat dihilangkan dan tidak harus menentukan parameter apa yang dilewati, karena semua parameter dimasukkan seperti berikut :[7]

```
GetName NULL, 'Ernest Hemingway'
```

### 2.4 Interaksi Manusia dan Komputer

Dalam buku *System Analysis and Design* yang ditulis oleh Kendall & Kendall mengatakan bahwa ineteraksi dengan pemakai yang minimal merupakan hal yang sangat penting dalam perancangan antarmuka grafis.

Penekanan tombol keyboard merupakan bagian yang paling membuat jalannya sistem komputer menjadi lambat, oleh karena itu ada 8 petunjuk yang harus diperhatikan, yaitu :

1. Yang dimasukkan cukup kodenya saja, jangan semua teks yang ditampilkan pada layar.
2. Hanya memasukkan data yang belum tersimpan pada file atau tabel.
3. Sediakan karakter pengatur format data yang dibutuhkan tanpa pemakai menuliskannya.
4. Gunakan nilai default untuk field pada layar masukan.
5. Mendesain suatu penyaringan data yang mirip dengan data yang dikehendaki oleh pemakai.
6. Sediakan ketikan (*keystrokes*) yang dapat memunculkan menu pilihan.
7. Gunakan *Radio button* dan *Drop down list* untuk mengatur tampilan menu yang sesuai.
8. Atur posisi cursor agar berada pada tempat yang perlu untuk diakses berikutnya. [3]

### 2.4 Pemrograman C#

.NET Framework berisi sekumpulan *library* (pustaka) berupa kelas yang diatur dan dikelompokkan dalam bentuk hirarki *namespace*. *Class Library* dalam .NET Framework dibagi ke dalam dua kelompok : *Base Class Library* (BCL) dan *Framework Class Library* (FCL).

*Framework Class Library* (FCL) merupakan *superset* dari BCL dan mengacu ke seluruh *library* kelas yang terdapt di dalam .NET Class Library. FCL berisi sekumpulan kelas untuk pembuatan program yang melibatkan Windows Form (WinForm), ADO.NET, ASP.NET, *Language Integrated Query* (LINQ), *Windows Presentation Foundation* (WPF), *Windows Communication Foundation* (WCF) dll. [4]

Antarmuka *grafis* memungkinkan pemakai berinteraksi dengan program secara visual. Sebuah antarmuka grafis didukung oleh kontrol-kontrol antarmuka grafis yang membuat pemakai dapat menghemat waktu karena tidak perlu untuk menghafalkan perintah-perintah apa yang harus diberikan melainkan tinggal memilih menu-menu dalam bentuk grafis. [5].

### 2.5 Teori Himpunan<sup>1</sup>

Suatu kumpulan srigala, sekelompok anggur, atau sekawanan pinguin dapat dikategorikan sebagai Himpunan. Konsep matematis dari suatu himpunan dapat digunakan sebagai dasar untuk segala macam matematika. Konsep utama dari teori himpunan adalah keanggotaan, jika  $x$  anggota  $A$  ( $x$  adalah elemen dari  $A$ ,  $x$  terkandung di dalam  $A$ ), maka akan dituliskan  $x \in A$ . [6]

Himpunan dilambangkan dengan huruf kapital dan elemennya dilambangkan dengan huruf kecil (atau angka). Satu himpunan dapat berisi apa saja dan elemen-elemen dari suatu himpunan tidak perlu berupa objek

<sup>1</sup>[https://www3.nd.edu/~dgalvin1/10120/10120\\_S16/Topic02\\_6p2\\_Galvin.pdf](https://www3.nd.edu/~dgalvin1/10120/10120_S16/Topic02_6p2_Galvin.pdf)

yang sama. Dalam notasi matematika suatu himpunan dapat didefinisikan dengan mendaftar elemen-elemennya di antara kurung kurawal:

$$A = \{1, 2, 5, 8\}$$

$$B = \{x \mid x \text{ in } N, 10 \leq x \leq 200\}$$

**Himpunan Bagian (Subsets)**

Suatu himpunan P dikatakan merupakan himpunan bagian dari himpunan Q jika setiap elemen dari P juga merupakan elemen Q. Ini tidak terkecuali kemungkinan bahwa  $P = Q$ .

$$A = \{1, 2, 3, 4, 5, 6\}$$

$$B = \{1, 3, 7\}$$

$$C = \{2, 4, 6\}$$

Pada contoh di atas, maka C adalah himpunan bagian dari A, tetapi B bukan himpunan bagian dari A karena mengandung unsur 7 yang bukan anggota himpunan A.

**Operasi Himpunan**

Operasinya pada himpunan meliputi union (gabungan), interseksi (irisan), komplemen (negasi), difference (perbedaan), dan produk Cartesian.

**Gabungan (UNION)**

Gabungan dua himpunan A dan B adalah himpunan yang berisi semua elemen dalam A atau B atau keduanya. Hal ini dituliskan dengan  $A \cup B$ . Gabungan dari n himpunan,  $A_1, A_2, \dots, A_n$  adalah himpunan semua objek yang termasuk setidaknya salah satu dari himpunan:

$$A = \{2, 5\}$$

$$B = \{1, 9\}$$

$$A \cup B = \{1, 2, 5, 9\}$$

Ilustrasi diagram Venn :



Apabila ada elemen himpunan A yang sama dengan elemen himpunan B, maka elemen tersebut hanya disebutkan satu kali saja untuk mencegah duplikasi penggunaan operator SQL pada operator lain.

**Irisan (Intersection)**

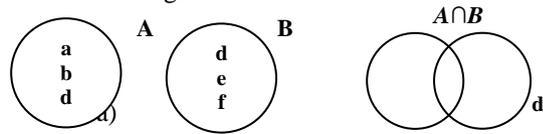
Irisan dua himpunan A dan B adalah himpunan semua elemen secara umum yang ditemukan di kedua himpunan A maupun B. It is written as  $A \cap B$ . Kasus ini ditulis sebagai  $A \cap B$ . The intersection of n sets,  $A_1, A_2, \dots, A_n$  is the set of all objects that belong to every one of the sets: Irisan dari n himpunan,  $A_1, A_2, \dots, A_n$  adalah himpunan semua objek yang menjadi milik setiap salah satu himpunan:

$$A = \{a, b, d\}$$

$$B = \{d, e, f\}$$

$$A \cap B = \{d\}$$

Ilustrasi diagram Venn:



**Komplemen**

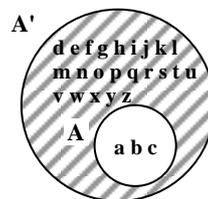
Komplemen (atau pelengkap absolut) dari himpunan A adalah himpunan NOT (A) atau  $A'$  yang terdiri dari semua unsur yang tidak dalam A. Definisi ini mensyaratkan adanya himpunan Uni-versal U:

$$A = \{a, b, c\}$$

$$U = \{a, b, c, \dots, z\}$$

$$A' = \{d, e, f, \dots, z\}$$

Ilustrasi diagram Venn:



**Selisih (Difference)**

Selisih dari himpunan A dan B didefinisikan sebagai himpunan  $A - B$ , terdiri dari semua elemen A yang tidak ada dalam B. Selisih dari A dan B tidak sama dengan selisih dari B dan A. Definisi selisih berarti bahwa A dan B tidak memiliki kesamaan apapun.

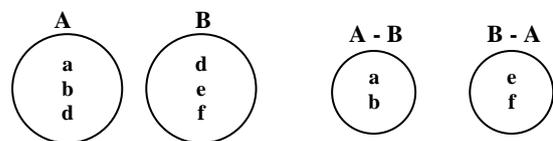
$$A = \{1, 3, 5, 7, 8\}$$

$$B = \{3, 4, 6, 8, 10\}$$

$$A - B = \{1, 5, 7\}$$

$$B - A = \{4, 6, 10\}$$

Ilustrasi diagram Venn:



**Produk Cartesian(Cartesian Product)**

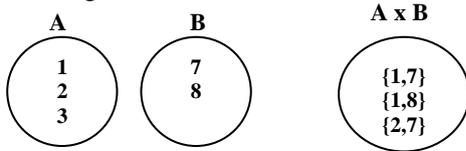
Produk Cartesian dari dua himpunan A dan B dilambangkan dengan  $A \times B$  (juga disebut produk himpunan atau produk dari A dan B) adalah satu himpunan pasangan berurutan dimana komponen pertama adalah anggota dari set pertama dan yang kedua adalah komponen anggota set kedua:

$$A = \{1, 2, 3\}$$

$$B = \{7, 8\}$$

$$A \times B = \{(1,7), (1,8), (2,7), (2,8), (3,7), (3,8)\}$$

Ilustrasi diagram Venn:



### III. METODE PENELITIAN

#### 3.1 Strategi Pencapaian Tujuan Penelitian

Penelitian ini merupakan pengembangan dari penelitian sebelumnya yang berjudul "Generator Jadwal Kuliah dengan Teknik Pencarian dan Penyaringan", tujuan penelitian kali ini adalah menambahkan batasan yaitu perkecualian bagi tiap dosen agar jadwal yang dihasilkan tidak tumpang tindih dengan jadwal pribadi dosen dengan memakai teori himpunan.

Pertama peneliti akan memodelkan tabel-tabel pada database menjadi himpunan-himpunan jadwal dan menyediakan sebuah himpunan jadwal perkecualian dosen (jadwal pribadi dosen). Pembangkit jadwal akan dihasilkan dengan cara mencoba satu persatu seluruh jadwal yang tersedia yang tidak beririsan dengan jadwal yang sudah ada maupun tidak beririsan dengan jadwal pribadi dosen.

Himpunan jadwal semesta adalah sebuah himpunan yang anggotanya adalah seluruh jadwal yang mungkin, yaitu hari Senin - Jumat dan setiap hari mulai sesi 1 hingga sesi 4, kemudian himpunan semesta ini diinterseksi dengan jadwal-jadwal yang sudah terisi yaitu jadwal pemakaian ruang, jadwal semester dan jadwal pribadi dosen. Dengan hasil perolehan operasi interseksi ini maka hasil yang tidak berinterseksi (tidak tumpang tindih) dipastikan menjadi jadwal yang akan dihasilkan.

#### 4.2 Waktu dan tempat penelitian

Penelitian dilakukan pada bulan Maret –Juli 2019 di Laboratorium komputer Universitas Kristen Duta Wacana. Pengambilan data dan uji validitas prosedur dan algoritma dilakukan memakai data prodi Sistem Informasi UKDW.

#### 4.3 Peralatan dan Bahan

Peralatan yang diperlukan dalam penelitian ini antara lain adalah seperangkat komputer lengkap yang dijadikan server sementara untuk kepentingan penelitian sehingga data tetap berada di kampus UKDW. Laptop untuk digunakan masing-masing peneliti dengan koneksi ke jaringan internet. Memori sekunder (*harddisk* dan *flashdisk*) yang memadai dengan fungsi utama sebagai media *backup*.

Bahan lain berupa data diperoleh dari unit PUSPINDIKA, Biro Akademik, dan Administrasi Prodi SI Universitas Kristen Duta Wacana .

## IV. ANALISIS DAN IMPLEMENTASI

### 4.1 Perancangan Basis Data

Penelitian ini merupakan kelanjutan dari penelitian sebelumnya yang berjudul "Generator Jadwal Kuliah dengan Teknik Pencarian dan Penyaringan" oleh karena itu peneliti juga memakai desain database yang sudah ada dengan ditambahkan tabel yang digunakan untuk menyimpan jadwal dosen diluar jadwal kuliah yang pada penelitian ini disebut **TabelKecuali**.

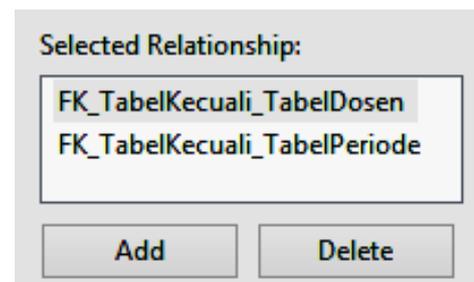
**TabelKecuali** tersebut membutuhkan kolom **periode** untuk mengetahui jadwal perkecualian tersebut berlaku pada periode semester dan tahun ajaran berapa karena jadwal yang akan dibuat adalah jadwal pada periode tertentu. Untuk mengenali jadwal perkecualian tersebut milik dosen yang mana maka tabel ini juga membutuhkan kolom **nik**. Kemudian untuk menentukan relasi antara jadwal perkecualian dengan jadwal yang dibuat dibutuhkan kolom-kolom : **hari**, **jamMulai** dan **jamSelesai** serta **keterangan** kegiatan.

Struktur tabel yang dibuat menyesuaikan dengan struktur tabel yang berelasi pada basis data yang sudah ada seperti Gambar 1 di bawah ini:

Column Name	Data Type	Allow Nulls
kecuali_id	int	<input type="checkbox"/>
periode	int	<input checked="" type="checkbox"/>
nik	char(8)	<input checked="" type="checkbox"/>
hari	varchar(6)	<input checked="" type="checkbox"/>
jamMulai	char(6)	<input checked="" type="checkbox"/>
jamSelesai	char(6)	<input checked="" type="checkbox"/>
keterangan	varchar(250)	<input checked="" type="checkbox"/>

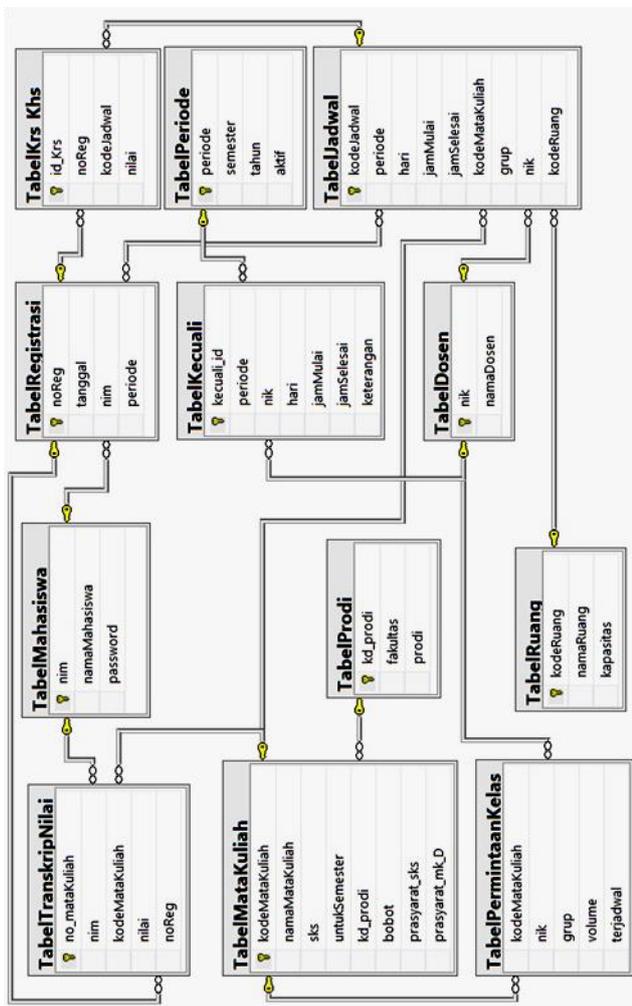
Gambar 1 Struktur TabelKecuali

Untuk menjaga integrasi data kolom periode dan kolom nik dan periode diatur sebagai kunci tamu (*foreign key*), yang masing-masing berhubungan dengan TabelDosen dan TabelPeriode, sehingga dibuat konstrain Foreign Key FK\_TabelKecuali\_TabelDosen dan FK\_TabelKecuali\_TabelPeriode seperti Gambar 2 di bawah ini:



Gambar 2 Konstrain Kunci Tamu pada TabelKecuali

Skema diagram database tersebut digambarkan seperti Gambar 3 di bawah ini :



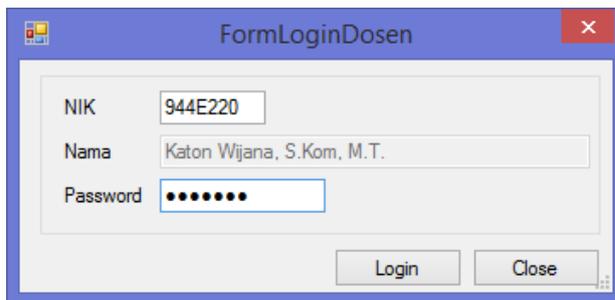
Gambar 3 Skema Diagram Basis Data Generator Jadwal Dengan Perkecualian

#### 4.2 Antarmuka Grafis

Untuk keperluan penelitian ini diperlukan beberapa antarmuka grafis untuk memasukkan data dan melakukan proses yang dibutuhkan untuk membuat jadwal secara otomatis dengan perkecualian.

##### 4.2.1 Antarmuka Grafis Jadwal Perkecualian

Sebelum jadwal kuliah dibuat, setiap dosen harus terlebih dahulu memasukkan jadwal kegiatannya, supaya ketika wakil dekan bidang akademik menyusun jadwal mengecualikan jadwal-jadwal tersebut. Jadwal ini diisi oleh masing-masing dosen yang akan mengajar pada periode tersebut, oleh karena itu sebelum mengisi Jadwal Perkecualian ini perlu dilakukan otentikasi memakai NIK dan password seperti Gambar 4 di bawah ini.



Gambar 4 Antarmuka grafis untuk Login jadwal

Setelah berhasil Login, maka dosen yang bersangkutan dapat mengisi jadwal-jadwal probadi agar nanti dikecualikan pada saat dibuat jadwal secara otomatis, seperti Gambar 5 berikut ini:



Gambar 5 Antarmuka grafis untuk mengisi jadwal perkecualian dosen

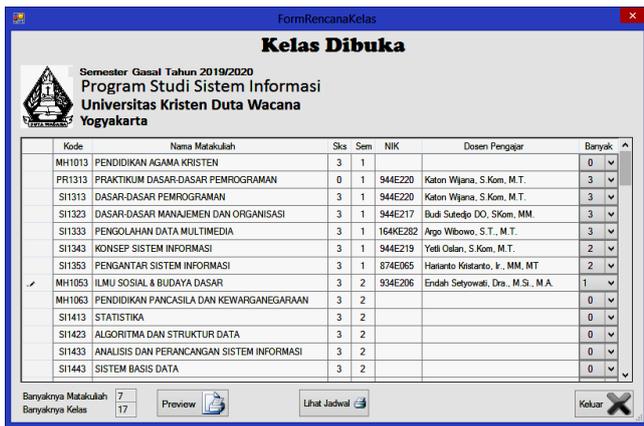
Demikian seterusnya dosen-dosen harus mengisi jadwal-jadwal yang harus dikecualikan, sehingga diperoleh data pada Tabel kecuai, misalnya seperti berikut:

	kecuai_id	periode	nik	hari	jamMulai	jamSelesai	keterangan
1	1	9	944E220	Jumat	07:30	10:00	Seminar Nasional
2	3	9	944E220	Rabu	07:30	10:00	Doa Pagi
3	4	9	944E219	Sabtu	07:30	19:00	Di rumah saja
4	5	9	944E220	Selasa	10:30	13:00	Rapat Mingguan
5	6	9	001DL016	Rabu	10:30	13:00	Kontrol dokter pribadi di Rumah Sakti
6	7	9	001DL016	Jumat	11:30	15:00	Ibadah hari Jumat
7	8	9	944E219	Selasa	07:30	10:00	Rapat Rutin Unit InQA Universitas

Gambar 6 Data jadwal perkecualian dosen pada TabelKecuali

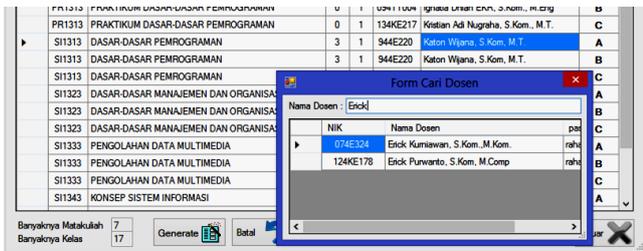
##### 4.2.2 Antarmuka Grafis Generator Jadwal Dengan Perkecualian

Untuk membuat jadwal secara otomatis, tiap-tiap program studi mempunyai daftar mata kuliah masing-masing, pada penelitian ini digunakan daftar mata kuliah untuk program studi Sistem Informasi. Antarmuka grafis pertama kali akan menampilkan semua mata kuliah yang diselenggarakan oleh program studi Sistem Informasi dan mata kuliah umum (mata kuliah bersama seluruh prodi). Gambar 7. adalah antarmuka grafis yang menampilkan seluruh mata kuliah yang ada tersebut



Gambar 7 Menentukan kelas yang akan ditawarkan

Melalui antarmuka grafis ini ketua program studi menentukan nama-nama dosen utama yang akan mengajar mata kuliah tersebut dengan cara melakan double-klik pada sel yang akan diisi, setelah itu menentukan banyaknya kelas yang akan dibuka dengan cara memilih pada Combo box yang tersedia. Sebagai contoh seperti tampak pada Gambar 7 di atas, mata kuliah Dasar-Dasar Pemrograman akan diampu oleh Katon Wijana, S.Kom., M.T. dan akan dibuka sebanyak 3 kelas. Setelah semua mata kuliah yang akan ditawarkan sudah ditentukan nama dosen pengajar utamanya dan banyaknya kelas yang akan dibuka, dengan menekan tombol Preview maka akan ditampilkan grup-grup sesuai dengan banyaknya kelas yang sudah ditentukan. Pada antarmuka grafis ini pada mulanya nama-nama dosen untuk mata kuliah tertentu pada semua grup sama, namun dapat diganti dengan cara melakukan double-klik pada sel yang akan diganti nama dosennya seperti Gambar 8 di bawah ini:



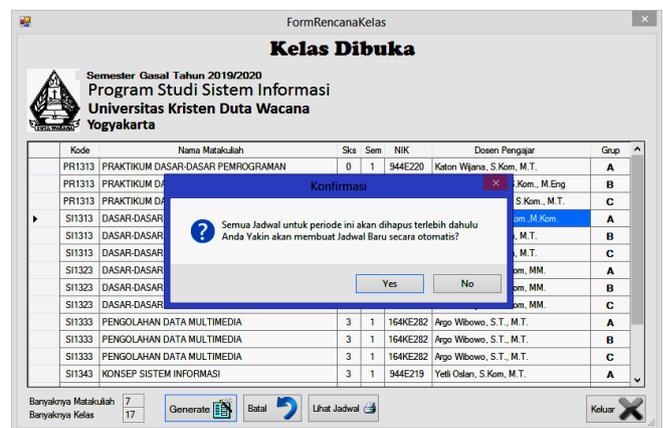
Gambar 4.8 Mengganti nama dosen pengajar

Setelah ditentukan nama-nama dosen pengajar, maka akan diperoleh daftar kelas beserta grupnya seperti tampak pada Gambar 9, kemudian jadwal otomatis siap untuk dibuat dengan cara menekan tombol Generate.



Gambar 4.9 Daftar kelas beserta grup yang akan dibuat jadwalnya

Untuk menghindari jadwal ganda, maka apabila dilakukan pembuatan Jadwal secara otomatis (Generate), maka semua jadwal kuliah pada periode tersebut akan dihapus terlebih dahulu, program aplikasi akan melakukan konfirmasi melalui jendela pesan seperti Gambar 10 di bawah ini:



Gambar 10 Konfirmasi penghapusan jadwal yang sudah dibuat sebelumnya

Jika dipilih tombol Yes, maka proses pembuatan jadwal secara otomatis dengan perkecualian akan dibuat, apabila proses sudah selesai maka akan ditampilkan pesan bahwa proses sudah selesai seperti Gambar 11 berikut ini:



Gambar 11 Konfirmasi bahwa jadwal dengan perkecualian sudah selesai dibuat

Untuk melihat jadwal yang dihasilkan dapat dilakukan dengan cara menekan tombol Lihat Jadwal, sehingga

akan ditampilkan Jadwal yang sudah jadi seperti Gambar 12 di bawah ini:

Hari	Mulai	Selesai	Nama Mata Kuliah	SSKS	Nama Dosen	Ruang	Grup
Senin	07.30	10.00	PRAKTIKUM DASAR-DASAR PEMROGRAMAN	0	Katon Wijana, S.Kom. M.T.	B.3.1	A
Selasa	07.30	10.00	PRAKTIKUM DASAR-DASAR PEMROGRAMAN	0	Katon Wijana, S.Kom. M.T.	B.3.1	B
Kamis	07.30	10.00	PRAKTIKUM DASAR-DASAR PEMROGRAMAN	0	Katon Wijana, S.Kom. M.T.	B.3.1	C
Senin	10.30	13.00	DASAR-DASAR PEMROGRAMAN	3	Katon Wijana, S.Kom. M.T.	B.3.1	A
Rabu	10.30	13.00	DASAR-DASAR PEMROGRAMAN	3	Katon Wijana, S.Kom. M.T.	B.3.1	B
Kamis	10.30	13.00	DASAR-DASAR PEMROGRAMAN	3	Katon Wijana, S.Kom. M.T.	B.3.1	C
Senin	07.30	10.00	DASAR-DASAR MANAJEMEN DAN ORGANISASI	3	Budi Seteja DO, S.Kom. MM.	B.3.2	A
Senin	07.30	10.00	DASAR-DASAR MANAJEMEN DAN ORGANISASI	3	Budi Seteja DO, S.Kom. MM.	B.3.2	B
Rabu	07.30	10.00	DASAR-DASAR MANAJEMEN DAN ORGANISASI	3	Budi Seteja DO, S.Kom. MM.	B.3.1	C
Rabu	07.30	10.00	PENGOLAHAN DATA MULTIMEDIA	3	Arjo Wibowo, S.T. M.T.	B.3.2	A
Kamis	07.30	10.00	PENGOLAHAN DATA MULTIMEDIA	3	Arjo Wibowo, S.T. M.T.	B.3.2	B
Senin	07.30	10.00	PENGOLAHAN DATA MULTIMEDIA	3	Arjo Wibowo, S.T. M.T.	B.3.3	C
Kamis	07.30	10.00	KONSEP SISTEM INFORMASI	3	Yetli Oslan, S.Kom. M.T.	B.3.3	A
Rabu	07.30	10.00	KONSEP SISTEM INFORMASI	3	Yetli Oslan, S.Kom. M.T.	B.3.3	B
Jumat	07.30	10.00	PENGANTAR SISTEM INFORMASI	3	Harianto Kristanto, i. MM. MT	B.3.1	A
Senin	10.30	13.00	PENGANTAR SISTEM INFORMASI	3	Harianto Kristanto, i. MM. MT	B.3.2	B

Gambar 4.12 Hasil Generator Jadwal Dengan Perkecualian

### 4.3 Implementasi Teori Himpunan

Untuk mengimplementasikan teori himpunan pada penelitian ini menggunakan relasi interseksi untuk mendeteksi adanya tabrakan pada suatu jadwal. Ada 4 hal yang harus diperiksa untuk mencegah adanya jadwal yang tumpang tindih yang berakibat kuliah tidak dapat dilaksanakan, yaitu :

#### a. Semester dan Grup

Karena banyaknya mahasiswa peserta kuliah, maka penyelenggaraan kuliah untuk tiap mata kuliah dibuat beberapa grup, sehingga untuk mata kuliah yang disediakan untuk semester tertentu, untuk grup yang sama satu dengan yang lain tidak boleh tumpang tindih. Sebagai contoh misalnya mata kuliah Dasar-Dasar Pemrograman untuk semester 1 dan Pengantar Sistem Informasi yang juga untuk mahasiswa semester 1, maka jadwal kelas Dasar-Dasar Pemrograman Grup A tidak boleh tumpang tindih dengan kelas Pengantar Sistem Informasi Grup A karena kedua mata kuliah tersebut seharusnya diambil oleh mahasiswa yang sama.

#### b. Jadwal mengajar Dosen

Pada waktu jadwal dibuat, apabila seorang dosen sudah dijadwalkan pada hari dan jam tertentu, maka jadwal yang lainnya tidak boleh tumpang tindih dengan jadwal mengajar dosen yang sudah ada.

#### c. Jadwal pemakaian ruang kuliah

Sama seperti jadwal mengajar dosen, apabila pada hari dan jam tertentu suatu ruang sedang digunakan, maka untuk jadwal yang baru tidak boleh menempati ruang tersebut.

#### d. Jadwal yang harus dikecualikan

Apabila seorang dosen mempunyai jadwal pribadi pada hari dan jam tertentu, maka pada hari dan jam tersebut dosen yang bersangkutan tidak dapat dijadwalkan. Pada prinsipnya ama seperti point b tetapi yang diperiksa bukan dari jadwal yang telah dibuat melainkan dari tabel perkecualian yang sudah diisi oleh masing-masing dosen.

#### 4.3.1 Prosedur Mendeteksi Irisan Pada Jadwal

Untuk mendeteksi irisan pada jadwal yang sudah dibuat, point a, b dan c dapat disatukan dalam sebuah query SELECT, kemudian dapat dihitung berapa banyak baris yang bertabrakan berdasarkan kriteria tersebut. Berikut ini adalah prosedur yang dibuat:

```
USE [SistemInformasiAkademik2019]
GO
/***** Object: StoredProcedure
[dbo].[interseksi]
Script Date: 10/18/2019 14:11:19 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author       : Katon Wijana, S.Kom., M.T.
-- Create date  : 17 Oktober 2019
-- Description  : Mendeteksi Irisan Jadwal
-- =====
```

```
CREATE PROCEDURE [dbo].[interseksi] (@periode
int, @semester int, @hari char(6), @grup
char(1), @mulai char(5), @selesai char(5), @nik
char(8), @kodeRuang int)
AS
BEGIN
(SELECT COUNT(kodeJadwal) ada FROM View_Jadwal
WHERE
((periode=@periode AND untukSemester=@semester
AND
hari=@hari AND grup=@grup) OR
(periode=@periode AND
hari=@hari AND nik=@nik) OR (periode=@periode
AND
hari=@hari AND kodeRuang=@kodeRuang)) AND
((jamMulai>=@mulai AND jamMulai<@selesai) OR
(jamSelesai>@mulai AND
jamSelesai<=@selesai)));
END
```

Prosedur yang bernama interseksi ini memakai 8 parameter, yaitu periode, semester, hari, grup, jam mulai, jam selesai, nik dosen dan kode ruang. Parameter-parameter tersebut akan digunakan untuk mendeteksi 3 macam irisan :

- Untuk mendeteksi irisan mengenai semester dan grup digunakan kriteria :  
(periode=@periode AND untukSemester=@semester AND hari=@hari AND grup=@grup)
- Untuk mendeteksi irisan jadwal mengajar dosen digunakan kriteria :  
(periode=@periode AND hari=@hari AND nik=@nik)
- Untuk mendeteksi irisan jadwal pemakaian ruang digunakan kriteria :  
(periode=@periode AND hari=@hari AND kodeRuang=@kodeRuang)

Ketiga kriteria tersebut digabungkan dengan jam mulai dan jam selesai jadwal, apabila ada salah satu waktu yang berada diantara jam mulai dan jam selesai maka dapat dipastikan jadwal tumpang tindih (bertabrakan). Kriteria jam tersebut adalah seperti berikut:

```
(jamMulai>=@mulai AND jamMulai<@selesai)
OR
```

```
(jamSelesai>@mulai AND jamSelesai<=
@selesai))
```

#### 4.3.2 Prosedur Mendeteksi Irisan Pada Jadwal Yang Dikecualikan

Untuk mendeteksi jadwal yang dikecualikan dilakukan dengan cara yang sama seperti prosedur irisan pada sub bab 4.3.1, hanya saja banyaknya parameter hanya 5, yaitu : periode, hari, jam mulai, jam selesai dan nik dosen. Prosedur ini diberinama interseksiKecuali, seperti di bawah ini:

```
USE [SistemInformasiAkademik2019]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author       : Katon Wijana, S.Kom., M.T.
-- Create date  : 18 Oktober 2019
-- Description  : Mendeteksi Irisan Kecuali
-- =====
ALTER PROCEDURE
[dbo].[interseksiKecuali] (@periode int, @hari
char(6), @mulai char(5), @selesai char(5), @nik
char(8))
AS
BEGIN
    (SELECT COUNT(kecuali_id) ada FROM
TabelKecuali WHERE
(periode=@periode AND hari=@hari AND nik=@nik)
AND
((jamMulai>=@mulai AND jamMulai<@selesai) OR
(jamSelesai>@mulai AND
jamSelesai<=@selesai));
END
```

Prinsip kerjanya sama seperti prosedur interseksi, hanya saja tabel yang diselidiki interseksinya adalah tabel yang bernama TabelKecuali.

#### 4.3.3 Kelas Himpunan

Untuk mempermudah penggunaan prosedur pada antarmuka grafis, maka dibuat kelas Himpunan yang mempunyai metode-metode untuk memanggil prosedur-prosedur tersebut. Berikut ini adalah kelas Himpunan yang dimaksud:

```
public class Himpunan{
    public static bool interseksi(int periode, int
semester, string hari, string grup, string
mulai,
string selesai, string nik, int kodeRuang){
    SqlConnectionStringBuilder strCon = new
    SqlConnectionStringBuilder();
    strCon.DataSource = ".\\SQLSERVER";
    strCon.InitialCatalog =
    "SistemInformasiAkademik2019";
    strCon.IntegratedSecurity = true;
    SqlConnection con = new
    SqlConnection(strCon.ToString());
    con.Open();
    string stringSQL = "EXEC interseksi {0}, {1},
    {2}, {3}, {4}, {5}, {6}, {7}";
    stringSQL = String.Format(stringSQL, periode,
semester, hari, grup, mulai, selesai, nik,
kodeRuang);
```

```
SqlCommand cmd = new SqlCommand(stringSQL,
con);
SqlDataReader dr = cmd.ExecuteReader();
int hasil = 0;
if (dr.HasRows){
    dr.Read();
    hasil =
    int.Parse(dr.GetValue(dr.GetOrdinal(
"ada")).ToString());
}
con.Close();
return (hasil>0);
}

public static bool interseksiKecuali(int periode,
string hari, string mulai, string selesai,
string nik){
    SqlConnectionStringBuilder strCon =
    new SqlConnectionStringBuilder();
    strCon.DataSource = ".\\SQLSERVER";
    strCon.InitialCatalog =
    "SistemInformasiAkademik2019";
    strCon.IntegratedSecurity = true;
    SqlConnection con = new
    SqlConnection(strCon.ToString());
    con.Open();
    string stringSQL = "EXEC interseksiKecuali {0},
    {1}, {2}, {3}, {4}";
    stringSQL = String.Format(stringSQL, periode,
hari, mulai, selesai, nik);
    SqlCommand cmd = new SqlCommand(stringSQL,
con);
    SqlDataReader dr = cmd.ExecuteReader();
    int hasil = 0;
    if (dr.HasRows){
        dr.Read();
        hasil =
        int.Parse(dr.GetValue(dr.GetOrdinal(
"ada")).ToString());
    }
    con.Close();
    return (hasil > 0);
}
}
```

#### 5.3.4 Cara Kerja Generator Jadwal Dengan Perkecualian

Program aplikasi Generator Jadwal Mengajar Dosen Dengan Perkecualian memakai metode Himpunan ini akan membuat seluruh jadwal yang akan ditawarkan dibuat jadwalnya, oleh karena itu untuk menghindari duplikasi jadwal (jadwal yang sudah ada dibuat lagi) maka pertama kali semua jadwal pada periode tersebut akan dihapus terlebih dahulu, kemudian baru di-Generate jadwal secara otomatis seperti tampak pada program tombol btnGenerate di bawah ini:

```
private void btnGenerate_Click(object sender,
EventArgs e)
{
    DialogResult jwb = MessageBox.Show("Semua Jadwal
    untuk periode ini akan dihapus terlebih dahulu"
+
    "\nAnda Yakin akan membuat Jadwal Baru secara
    otomatis?", "Konfirmasi",
    MessageBoxButtons.YesNo, MessageBoxIcon.Question);
    if (jwb == DialogResult.Yes)
    {
        jadwal.eksekusiSQL("DELETE FROM TabelJadwal
        WHERE periode=" +
```



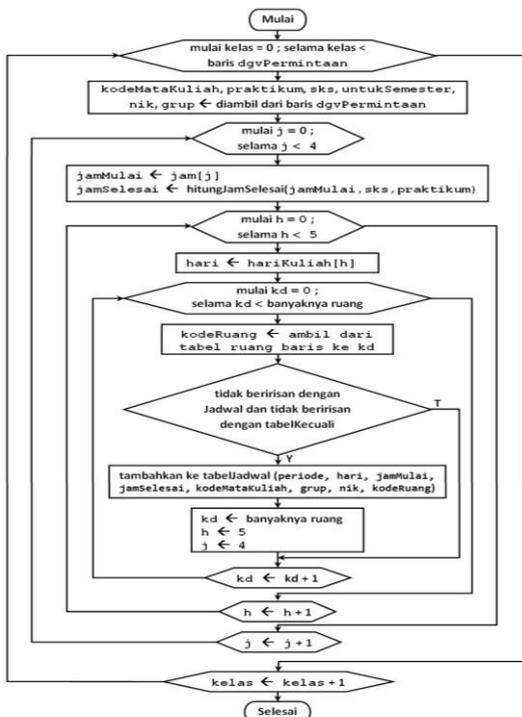
teorinya, sehingga nilai SKS untuk praktikum dihitung sebesar 0, namun pada realisasinya jam praktikum adalah 3 SKS. Untuk keperluan menentukan jam selesai suatu kelas, khususnya apabila mata kuliah tersebut adalah praktikum, yaitu yang ditandai dari 2 digit pertama dari kode mata kuliah adalah "PR" maka perhitungan jam selesai adalah 3 SKS, selain itu akan menyesuaikan nilai SKS yang ada.

Proses pencarian jadwal akan dilakukan terus selama belum mendapatkan jadwal, kecuali batasannya sudah habis (seluruh kemungkinan sudah dicoba), oleh karena itu ketika jadwal sudah didapatkan proses pencarian harus dihentikan untuk menghindari duplikasi jadwal. Untuk keperluan ini dipakai variabel sudah yang diberi nilai awal false dan jika sudah ditemukan maka sudah diberi nilai true.

Generator jadwal ini hanya digunakan untuk membuat jadwal pada periode tertentu saja, dalam hal ini periode yang dimaksud adalah periode yang sudah ditulis pada label lblPeriode, kode programnya adalah seperti berikut:

```
bool praktikum = false, sudah = false;
periode = int.Parse(lblPeriode.Text);
```

Selanjutnya akan dibuat jadwal otomatis untuk tiap kelas (grup) yang sudah ditentukan, yaitu yang ditampilkan pada DataGridView dgvPermintaan. Penjadwalan ini mengutamakan jam pagi, sehingga apabila suatu jadwal overlap maka akan dicoba dulu hari berikutnya pada jam yang paling pagi terlebih dahulu, seperti Flowchart di bawah ini:



Gambar 4.12 Flowchart Generator Jadwal Dengan Perkecualian

## V. KESIMPULAN

Berdasarkan hasil penelitian dan eksperimen data yang telah dilakukan menggunakan studi kasus jadwal kuliah prodi Sistem Informasi Universitas Kristen Duta Wacana Yogyakarta, dapat ditarik beberapa kesimpulan sebagai berikut:

1. Jadwal kuliah maupun jadwal kegiatan pribadi masing-masing dosen dapat dipandang sebagai himpunan-himpunan jadwal yang dapat dioperasikan memakai teori himpunan untuk menentukan adanya irisan (interseksi) maupun tidak adanya interseksi (overlap jadwal).
2. Dengan memakai model teori himpunan, khususnya komplemen dari irisan (interseksi) Aplikasi **Generator Jadwal Mengajar Dosen Dengan Perkecualian Memakai Teori Himpunan** dapat menghasilkan jadwal secara otomatis dengan memperhatikan jadwal pribadi masing-masing dosen.
3. Untuk dapat menghasilkan jadwal kuliah secara otomatis dengan memperhatikan jadwal pribadi dosen, sistem ini membutuhkan himpunan-himpunan jadwal yang berkaitan dengan jadwal yang akan dibuat, yaitu:
  - Himpunan daftar kelas yang akan ditawarkan, termasuk mata kuliah, grup dan nama dosen pengajar yang ditawarkan
  - Himpunan daftar jadwal pribadi dosen pada tabel **TabelKecuali**.
4. Teori himpunan yang digunakan untuk menghasilkan jadwal otomatis ini adalah operasi interseksi (irisan), sehingga untuk membuat jadwal dilakukan dengan cara memilih yang tidak beririsan dengan Jadwal yang sudah ada dan tidak beririsan dengan jadwal pribadi dosen, apabila suatu jadwal yang direncanakan beririsan maka sistem akan mencoba jadwal berikutnya secara berurutan (sequensial) sehingga dapat diperoleh jadwal yang aman dari overlap.
5. Pemeriksaan interseksi himpunan (overlap jadwal) dilakukan secara bertahap, proses pembuatan jadwal dilakukan dengan urutan prioritas iterasi sebagai berikut:
  - a. Daftar kelas yang ditawarkan
  - b. Daftar hari yang tersedia
  - c. Daftar waktu (jam mulai kuliah)
  - d. Daftar ruang yang tersedia

Sehingga jadwal yang dihasilkan pertama-tama akan mengubah pilihan ruang kuliah terlebih dahulu, jika tidak memungkinkan maka akan mengubah jam mulainya dan yang terakhir baru mengubah hari kuliahnya. Apabila ruang, waktu dan hari tidak ada yang dapat memenuhi syarat, maka kelas tersebut tidak mendapatkan jadwal.

## UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Universitas Kristen Duta Wacana Yogyakarta, khususnya

Program Studi Sistem Informasi Fakultas Teknologi Informasi atas fasilitas-fasilitas yang disediakan sehingga dapat terselesaikannya penelitian ini.

Penulis juga sangat berterima kasih kepada rekan-rekan dosen yang tidak jemu-jemu untuk membantu memberikan petunjuk dalam menyelesaikan masalah penelitian ini.

Tidak lupa tentu saja kepada segenap staf redaksi maupun tim *reviewer* yang telah memeriksa naskah penelitian saya, kiranya Tuhan yang maha pengasih menyertai dan memberkati kita semua.

#### DAFTAR PUSTAKA

- [1] Coronel, Morris and Rob, "Database systems design, implementation and management," Cengage Learning. Boston, pp. 11-13, 2013.
- [2] Mark Spenik and Orryn Sledge, "Microsoft SQL Server 2000 DBA Survival Guide," Sams Publishing. Indianapolis, pp. 559-574, 2001.
- [3] Kenneth E. Kendall and Julie E. Kendall, "System analysis and design," Prentice Hall. Upper Saddle River, New Jersey, pp. 555-556, 2006.
- [4] Raharjo Budi, "Mudah belajar C#," Penerbit Informatika. Bandung, pp. 2, 2015.
- [5] H.M. Dietel, P.J. Dietel, J.A. Listfield, T.R. Nieto, C.H. Yaeger and M. Zlatkina, "C#: A Programmer's introduction," Prentice Hall. Upper Saddle River, New Jersey, pp. 361, 2003.
- [6] Halmos Paul R., "Naive Set Theory", Springer Science+Business Media New York, pp.1, 1974
- [7] Free Programming Book, "Microsoft® SQL Server® Notes for Professionals", <https://goalkicker.com/MicrosoftSQLServerBook>, pp. 158