

### IMPLEMENTASI METODE *BEST FIRST FIT* DALAM PENENTUAN PENEMPATAN RUANGAN KELAS (STUDI KASUS: INSTITUT BISNIS DAN INFORMATIKA (IBI) KOSGORO 1957)

#### PENULIS

<sup>1)</sup>Prita Vernanda Virly, <sup>2)</sup>Boy Firmansyah

#### ABSTRAK

Kesesuaian ruang kelas dengan jumlah mahasiswa yang belajar merupakan hal yang sangat penting, agar dapat memberikan kenyamanan saat berlangsungnya kegiatan belajar. Penentuan kelas biasanya dilakukan secara manual dengan cara menyamakan jumlah mahasiswa di satu mata kuliah dengan daya tampung setiap kelas. Hal tersebut membutuhkan waktu yang lama dan terkadang terjadi kesalahan dalam penentuan kelas, maka dari itu diperlukan sistem untuk membantu pekerjaan tersebut dan mengurangi kesalahan. Sistem ini akan secara otomatis menentukan kelas yang sesuai setelah diketahui jumlah mahasiswa di setiap mata kuliah. Dalam pembangunan sistem ini perlu dilakukan analisa prosedur terlebih dahulu sehingga dapat menentukan kebutuhan sistem baik *database* dan juga antarmuka. Sistem berbasis desktop dengan pemrograman VB yang terintegrasi oleh *database*.

#### Kata Kunci

*Visual Basic, Database, Pencarian Ruang, Best First Fit*

#### AFILIASI

Prodi, Fakultas  
Nama Institusi  
Alamat Institusi

<sup>1)-2)</sup>Teknik Informatika, Fakultas Ilmu Komputer

<sup>1)-2)</sup>Institut Bisnis dan Informatika (IBI) Kosgoro 1957

<sup>1)-2)</sup>Jl. M. Kahfi II No. 33, Jagakarsa, Jakarta Selatan, DKI Jakarta

#### KORESPONDENSI

Penulis  
Email

Boy Firmansyah  
boy@ibi-k57.ac.id

#### LICENSE



This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).

## I. PENDAHULUAN

Penelitian dilakukan seputar pemanfaatan teknologi dalam melakukan pengolahan data maupun membantu beberapa pekerjaan yang berhubungan dengan data. Masalah ditemukan pada belum adanya program yang dapat secara otomatis menentukan ruangan kelas untuk kegiatan perkuliahan. Penentuan ruangan kelas menjadi hal yang cukup penting supaya kegiatan perkuliahan dapat berjalan dengan baik.

Selama ini penentuan ruangan kelas masih dilakukan secara manual, yang terkadang terjadi kesalahan dalam pemilihan ruangan kelas. Kesalahan tersebut seputar kelas yang ditentukan tidak sesuai dengan jumlah mahasiswa sehingga kekurangan kursi dan ruangan terlalu sempit. Dan pemilihan ruangan kelas membutuhkan waktu yang lama bila dilakukan secara manual dan juga terkadang ruangan dipilih dua kali.

Penggunaan alat bantu berupa aplikasi dalam membantu pekerjaan yang biasanya dilakukan secara manual dapat memberikan banyak keuntungan baik dari segi waktu maupun hasilnya. Pemanfaatan teknologi di berbagai bidang membuat pekerjaan tersebut dapat selesai dengan cepat dan mengurangi beberapa kesalahan yang dapat terjadi. Maka diperlukan adanya program yang dapat membantu dan memberikan hasil yang sesuai. Program dibangun menggunakan metode *Best First Fit* agar dapat menentukan kelas mana yang sesuai.

## II. METODE PENELITIAN

**Metode *Best First Fit*** adalah manajer memori untuk mencari (men- *scan*) list segmen hingga menemukan tempat yang besar, tempat tersebut dibagi menjadi dua bagian untuk proses dan ruang kosong (Nurhayati dan Zulmi 2014). Metode ini hanya mengelola parameter tunggal, alurnya melakukan penempatan data pada *array* dengan bobot *index* yang sama (Nurhasan, Suyono, dan Muslim 2014).

Metode ini mengalokasikan dengan melakukan pencarian ke seluruh ruangan yang ada untuk menentukan ruang mana yang cukup untuk proses. Pada penelitian ini *Best First Fit* digunakan untuk menentukan penempatan ruangan kelas untuk setiap perkuliahan, berkaitan dengan kuota kelas yang akan disesuaikan dengan jumlah mahasiswa di masing-masing perkuliahan. Alur dalam penentuan ruangan kelas adalah dengan melakukan penempatan kelas dengan jumlah kuota sama dengan jumlah mahasiswa di perkuliahan tersebut, tetapi bila ruangan dengan kuota yang sama sudah terisi terlebih dahulu maka akan ditempatkan di kelas dengan kuota sedikit lebih besar.

Metode ini digunakan untuk menentukan penempatan ruangan kelas, tujuannya agar mengurangi kesalahan pemilihan kelas yang terlalu kecil untuk perkuliahan. Sehingga tidak lagi terjadi kelas terlalu kecil ataupun terlalu besar.

## III. HASIL DAN PEMBAHASAN

### Pemrograman Desktop

Pemrograman komputer adalah merupakan interaktif penulisan dan penyuntingan kode sumber agar dapat membentuk sebuah program (Saragih 2018). Proses pemrograman meliputi pengujian, analisis, perbaikan kesalahan, pengoptimasian terhadap algoritma, normalisasi kode dan jika program dikerjakan oleh tim maka perlu melakukan pengkoordinasian antar *programmer*.

Menurut (Mardainis dan Andesa 2017) pemrograman desktop adalah program aplikasi yang dapat berjalan tanpa adanya jaringan internet, biasanya digunakan untuk membuat program yang akan dioperasikan tanpa memerlukan jaringan internet.

Pemrograman desktop digunakan untuk membangun aplikasi Windows berbasis GUI (*Graphical User Interface*) yang dijalankan tanpa memerlukan *web browser* maupun internet. Untuk menggunakan program aplikasi desktop harus terlebih dahulu melakukan konfigurasi dan instalasi, yang terkadang memiliki spesifikasi dari sistem operasi komputer dan terutama untuk penggunaan secara *multi-user*.

Dalam membangun atau membuat program aplikasi pemrograman desktop menggunakan bahasa pemrograman untuk memberikan perintah ke komputer agar melakukan sesuai dengan yang kita pikirkan.

Bahasa pemrograman yang digunakan dalam pemrograman desktop antara lain, *Visual Basic*, C, C++, Cobol, dan bahasa lainnya.

### Visual Basic 2010 Express

*Visual Basic* adalah bahasa pemrograman komputer, sering disebut juga sebagai sarana (*tool*) untuk membangun program aplikasi berbasis *windows* (Wiliani dan Zambi 2017).

*Visual Basic* merupakan bahasa pemrograman *object (object oriented programming)*. *Visual Basic* menggunakan bahasa Inggris untuk setiap kode perintah sehingga mudah untuk dimengerti dan memudahkan untuk melakukan pengembangan terhadap program aplikasi berbasis desktop atau Windows karena bersifat visual dan dapat digerakkan oleh *mouse (mouse driven)*. Kemampuan dari *Visual Basic 2010 Express* antara lain :

- 1) Dapat digunakan untuk membuat atau membangun program aplikasi berbasis *desktop* atau Windows.
- 2) Dapat melakukan pengujian dengan perintah *debugging* untuk menguji program berjalan atau tidak.
- 3) Digunakan untuk membuat objek pembantu seperti : *file Help*, kontrol *active X*, aplikasi internet dan lainnya.
- 4) Menghasilkan program EXE yang dapat langsung dijalankan (*executable*).

### Database

Basis data (*Database*) merupakan satu kesatuan dari sekumpulan informasi yang disusun dan disimpan secara sistematis di dalam perangkat keras (*hardware*) agar dapat diolah oleh perangkat lunak (*software*) (Swara Yoga dan Pebriadi 2016). Data yang menjadi satu kesatuan di dalam *database* dapat diolah untuk menghasilkan informasi yang berguna.

Menurut Anisah dan Mayasari, 2016 *Database* adalah sekumpulan *file* yang saling berelasi, ditunjukkan dengan kunci pada tiap *file* yang ada. Satu kumpulan data atau *file* yang dipakai dalam satu lingkup merupakan satu *database*.

*Database* tersimpan secara terintegrasi dan menjadi sumber daya perusahaan yang dapat digunakan bersama-sama. *Database* berbeda dengan sistem *file* yang menyimpan data terpisah (Khotijah 2016).

Sekumpulan data yang harus diproses atau diolah dikumpulkan ke dalam satu wadah dan data-data tersebut saling berintegrasi, yang disebut sebagai *database*. Dalam menyimpan data-data tersebut dan mengelolanya harus menggunakan alat bantu berupa aplikasi seperti *Microsoft Access*, *MySQL Workbench*.

### Microsoft Access

Microsoft Access adalah sebuah sistem manajemen *database* atau *Database Management System* (DBMS). Berfungsi untuk menyimpan berbagai data, mengatur dan mengelolanya agar dapat mudah bila digunakan atau dibutuhkan (Oktarina dan Kuswantoro 2011).

Setiap komponen-komponen dalam Microsoft Access berfungsi untuk melakukan pengaturan dan pengelolaan suatu data di dalam *database*. Data-data disimpan ke dalam suatu *table* yang tersusun dari beberapa *field* yang berbeda, kumpulan dari setiap *field* di dalam satu baris disebut *record*. Setiap data yang tersimpan di dalam satu *database* memiliki karakteristik yang membedakan satu data dengan data yang lainnya yaitu *primary key*, merupakan *field* bersifat unik sebagai kunci utama untuk identitas dari suatu data.

### UML (Unified Modeling Language)

UML merupakan salah satu alat bantu di bidang pengembangan sistem yang berorientasi objek, karena menyediakan bahasa pemodelan visual (Syafei 2016). Pemodelan yang berfokus pada pendefinisian model sistem informasi dinamis dan struktur statis.

Menurut (Suendri 2018) *Unified Modeling Language* (UML) adalah bahasa yang berdasarkan gambar atau grafik untuk memvisualisasi, menspesifikasi, membangun dan pendokumentasian sistem pengembangan *software* berbasis *object*. UML memberikan standar untuk penulisan sebuah sistem *blue print*.

*Unified Modeling Language* (UML) merupakan bahasa yang menjadi standar di industri untuk visualisasi, merancang dan mendokumentasi sistem piranti lunak (Dharwiyanti dan Wahono Satria 2003).

UML merupakan bahasa atau bisa disebut juga sebagai *tool / model* yang digunakan untuk merancang sistem yang berorientasi *object* secara visual. *Unified Modeling Language* sebagai standar pemodelan yang dapat digunakan untuk menjelaskan dan menggambarkan pengembangan dari sebuah sistem.

*Unified Modeling Language (UML)* memiliki beberapa bentuk diagram yang digunakan untuk memvisualisasikan sebuah sistem, antara lain:

### **Use Case Diagram**

*Use case* diagram dibuat untuk memvisualisasikan hubungan yang terjadi antara aktor dengan *use case*. Diagram ini juga mempresentasikan kegunaan sistem dari sudut pandang *user* (Syafei 2016).

*Use case* diagram menggambarkan *use cases* dan *actor* beserta hubungannya. Diagram ini menjelaskan apa yang dilakukan oleh sistem, berguna untuk menentukan model dari sistem. Diagram ini memudahkan komunikasi dengan pengguna dan menjelaskan fasilitas yang disediakan oleh sistem.

### **Pengujian Black Box**

Pengujian *black box* menurut Jaya, 2018 adalah teknik pengujian perangkat lunak yang berfokus pada spesifikasi fungsional perangkat lunak. Pengujian *black box* dilakukan dari sudut pandang pengguna perangkat lunak.

Menurut Hidayat dan Muttaqin, 2018 *black box testing* merupakan pengujian dimana tester dapat mendefinisikan kumpulan kondisi *input* dan melakukan pengetesan spesifikasi fungsional program.

*Black box testing* pengujian perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program untuk mengetahui kesesuaian masukan dan keluaran pada program. (Cholifah, Yulianingsih, dan Sagita 2018).

### **Analisis Sistem**

Analisis sistem merupakan proses pengidentifikasian untuk dapat mengevaluasi masalah dan kebutuhan dalam penelitian yang dilakukan. Sehingga akan dapat menghasilkan usulan-usulan atau bahkan perkembangan dari sistem yang berjalan.

Penganalisan dilakukan dengan cara membagi sistem yang ada ke dalam komponen-komponen seperti teknik penguraian, untuk memudahkan dalam mengidentifikasinya. Beberapa hal yang akan dianalisis pada tahapan ini meliputi analisis sistem atau prosedur yang sedang berjalan, analisis kebutuhan sistem, kebutuhan pengguna, kebutuhan fungsional, *unified modelling language (UML)*.

### **Analisis Prosedur Berjalan**

Prosedur yang berjalan untuk menentukan penempatan ruangan dilakukan secara manual tanpa bantuan sistem. Dengan mencari secara satu per satu ruangan mana yang memiliki daya tampung yang cukup untuk setiap jumlah orang.

Data yang digunakan adalah data jadwal dengan berapa jumlah orang di setiap pembelajaran dan data ruangan yang tersedia. Data tersebut digunakan untuk menentukan ruangan mana yang sesuai.

### **Analisis Permasalahan**

Analisis permasalahan bertujuan untuk mengevaluasi kekurangan dari prosedur yang sedang berjalan sehingga dapat dilakukannya perubahan atau perkembangan. Pada prosedur yang sedang berjalan belum adanya sistem khusus untuk menentukan ruangan untuk setiap pembelajaran sehingga membutuhkan waktu yang relatif lama untuk mengerjakannya. Masalah yang lain seperti salah menentukan ruangan dan adanya perbedaan informasi ruangan.

### **Analisis Kebutuhan Pengguna**

Analisis pengguna menjelaskan siapa saja yang akan berhubungan atau menggunakan sistem ini. Setiap pengguna memiliki spesifikasinya masing-masing dengan hak aksesnya terhadap sistem. Adapun pengguna pada sistem ini dibedakan menjadi 2 yaitu:

- 1) **Pengelola**  
Pengelola memiliki akses untuk mengelola data ruangan dan data jadwal. Pengelolaan data mencakup meng-*input* data baru, menyunting isi data dan menghapus data yang sudah ada di dalam *database*.
- 2) **Admin**  
Admin hanya dapat melakukan pengoperasian sistem untuk menentukan ruangan yang cocok dan menyimpan data tersebut.

### Kebutuhan Fungsional

Sistem ini memiliki kebutuhan fungsional yang ditentukan berdasarkan proses di dalam sistem. Kebutuhan fungsional sistem sebagai berikut:

- 1) Mampu menentukan ruangan yang sesuai kebutuhan.
- 2) Mampu meminimalisir kesalahan dan mempersingkat waktu pengerjaan.

### Kebutuhan Sistem

Kebutuhan ini menspesifikasikan kebutuhan dari sistem untuk pembangunan sampai sistem diimplementasikan. Analisis kebutuhan ini menjelaskan kebutuhan perangkat keras dan juga perangkat lunak agar sistem ini dapat dibangun dan berjalan sesuai dengan prosedur dan perancangan.

- 1) Kebutuhan *Hardware*

Laptop dengan spesifikasi:

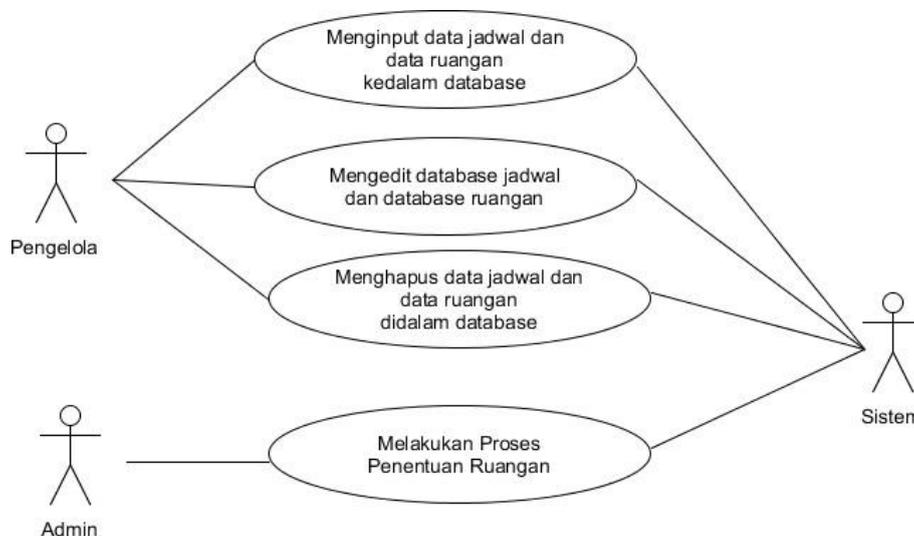
  - OS Windows 10 64Bit
  - RAM 2GB
  - Prosesor Intel Celeron CPU N2840
- 2) Kebutuhan *Software*
  - Visual Basic 2010 Express
  - Microsoft Office

### Unified Modelling Language (UML)

*Unified modelling language* digunakan untuk menggambarkan dan menjelaskan tahapan dari aktivitas pada sistem yang akan dibangun. Aktivitas pada sistem ini digambarkan dengan 3 jenis diagram yaitu, *use case diagram*, *class diagram*, *activity diagram*, *sequence diagram*. Adapun bentuk dari diagram-diagram tersebut sebagai berikut.

### Use Case Diagram

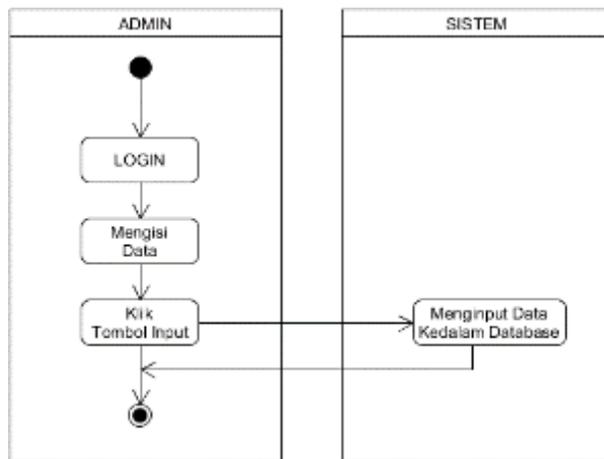
Gambar 1 berikut merupakan *use case diagram* untuk menunjukkan interaksi antara pengelola dan admin ke sistem.



Gambar 1. *Use Case Diagram*

### Activity Diagram Meng-input Data Jadwal dan Data Ruangan

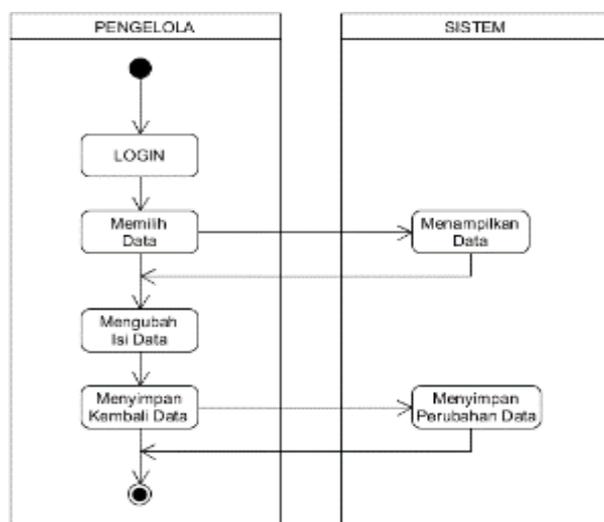
Gambar 2 berikut merupakan gambaran aktivitas yang dilakukan *user* untuk meng-*input* data baru ke dalam *database*.



Gambar 2 Activity Diagram Meng-input Data Jadwal Dan Data Ruang

### Mengedit Data Jadwal dan Data Ruang

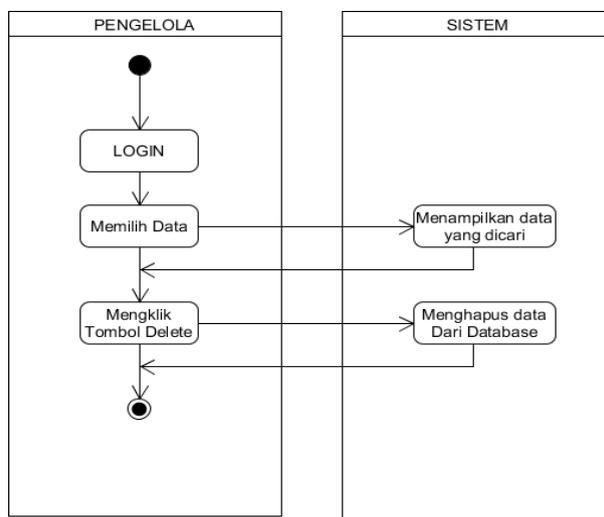
Gambar 3 berikut menggambarkan aktivitas penyuntingan isi data yang tersimpan di dalam *database*. Pengguna hanya dapat menyunting beberapa isi data



Gambar 3. Activity Diagram Mengedit Data Jadwal Dan Data Ruang

### Menghapus Data Jadwal dan Data Ruang

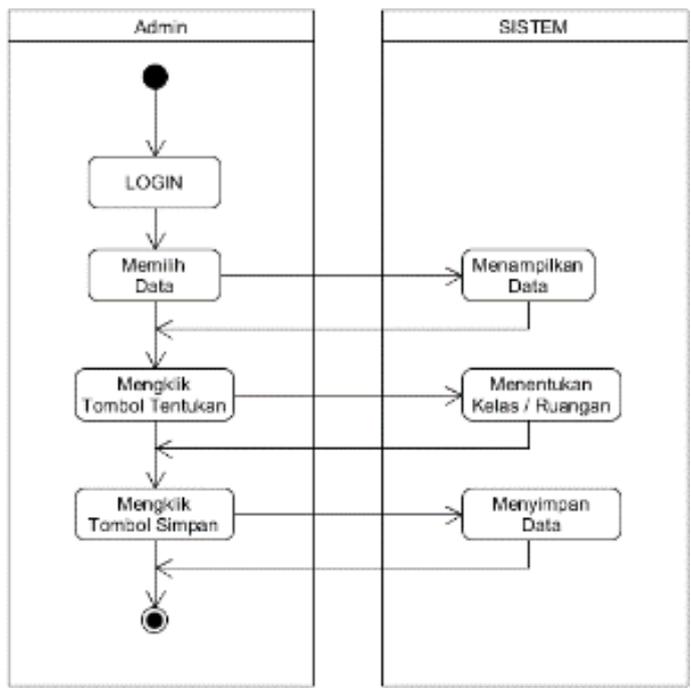
Gambar 4 berikut merupakan gambar alur kerja saat pengguna menghapus salah satu data yang ada di dalam *database*.



Gambar 4. Activity Diagram Menghapus Data Jadwal Dan Data Ruang

**Penentuan Ruang**

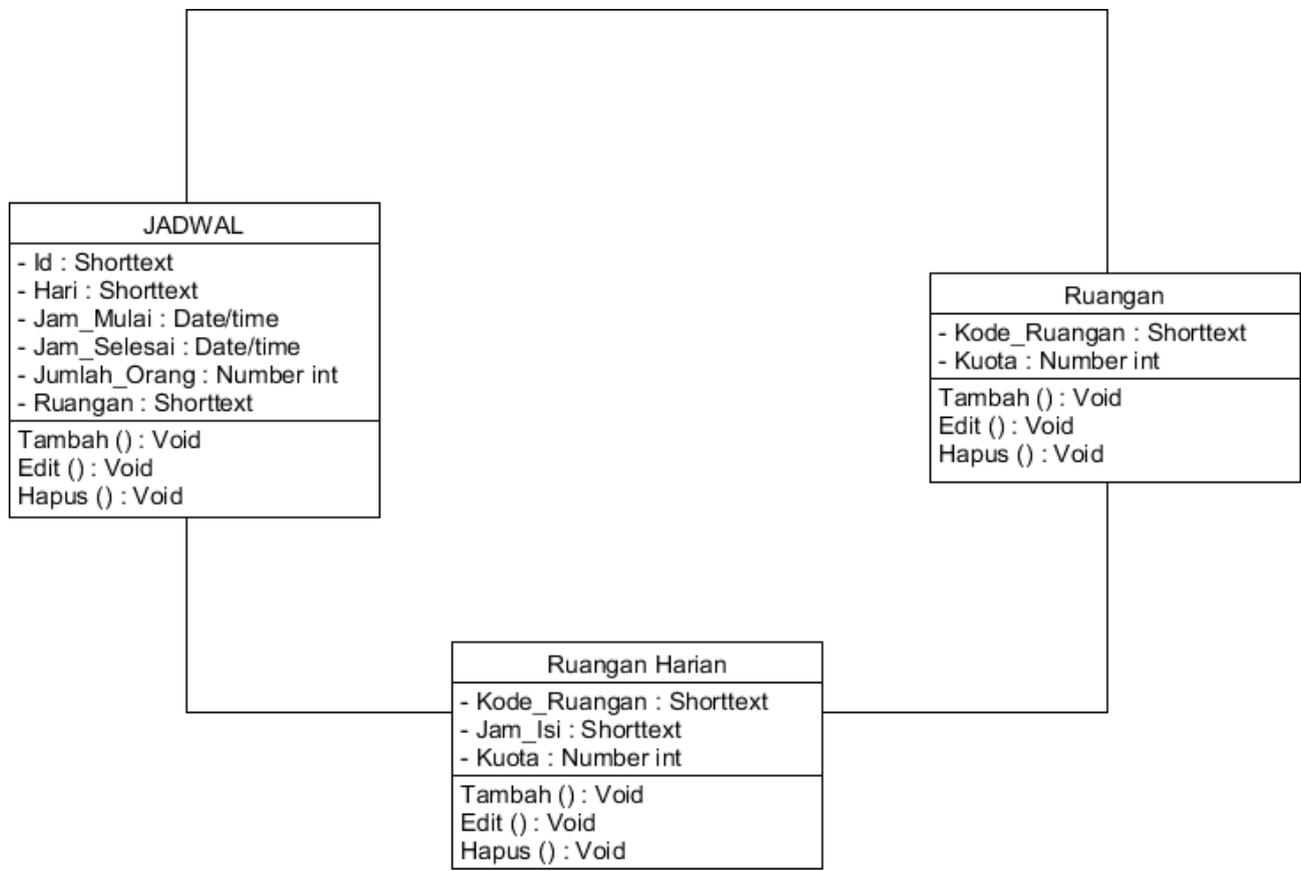
Gambar 5 berikut menunjukkan aktivitas untuk menentukan ruangan.



**Gambar 5. Activity Diagram Penentuan Ruang**

**Class Diagram**

*Class diagram* menjelaskan hubungan antar tabel atau *class* yang ada di dalam sistem, *class digram* pada sistem ini ditunjukkan pada Gambar 6



**Gambar 6. Class Diagram**

## Perancangan

Perancangan berfungsi untuk memberikan gambaran sederhana dari sistem yang sedang dibangun. Perancangan sebaiknya dilakukan sebelum pembangunan sebuah sistem agar menjadi acuan atau pedoman seperti apa sistem tersebut akan dibuat.

Perancangan dalam pembangunan sebuah sistem biasanya meliputi perancangan struktur tabel atau bisa disebut juga struktur data yang akan berhubungan dengan sistem. Selain itu ada juga perancangan *interface* atau antar muka dari sistem tersebut, perancangan ini sangat penting karena menjadi gambaran dan acuan bagaimana tata letak pada antar muka sistem.

## Perancangan Struktur Tabel

Pada *database* terdiri dari beberapa tabel, beberapa tabel memiliki struktur tabel yang sama. Sehingga berikut ini merupakan struktur tabel yang digunakan pada tabel-tabel yang ada pada *database*.

**Tabel 1. Struktur Tabel Ruangannya**

NAMA	TYPE	NULL	KETERANGAN
Ruangan	Short Text(5)	Not Null	Primary Key
Jam Isi	Short Text(10)	Not Null	
Kuota	Number(Int)	Not Null	

**Tabel 2. Struktur Tabel Jadwal**

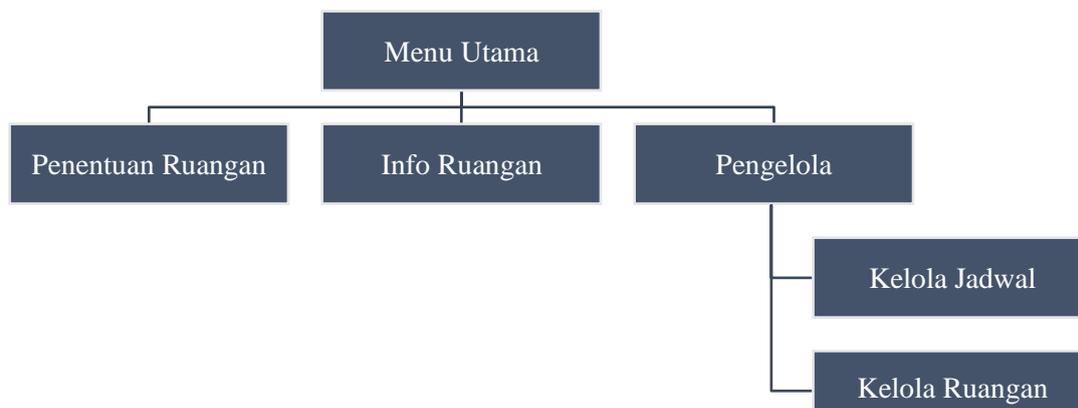
NAMA	TYPE	NULL	KETERANGAN
Id	Short Text(3)	Not Null	Primary Key
Matakuliah	ShortText(		
Hari	Varchar (6)	Not Null	
Jam Mulai	Time	Not Null	
Jam Selesai	Time	Not Null	
Jumlah Orang	Number(Int)	Not Null	
Ruangan	Short Text(5)	Null	

**Tabel 3. Struktur Tabel Ruangan**

NAMA	TYPE	NULL	KETERANGAN
Ruangan	Short Text(5)	Not Null	Primary Key
Kuota	Number(Int)	Not Null	

## Perancangan Struktur Menu

Perancangan struktur menu digunakan untuk menjelaskan fungsi setiap menu yang ada pada sistem. Struktur menu yang digunakan pada sistem atau aplikasi ini sebagai berikut:



**Gambar 8. Struktur Menu**

Struktur menu utama terdiri dari 3 sub menu dan salah satu sub menu memiliki sub menu. Sub menu penentuan ruangan berfungsi untuk menampilkan *form* penentu ruangan. Sub menu info ruangan menampilkan info ruangan setiap hari. Sub menu pengelola memiliki 2 sub menu yaitu kelola jadwal dan kelola ruangan, sub menu kelola jadwal dan kelola ruangan akan menampilkan *form login* pengelola.

### Implementasi Sistem

Tahapan ini merupakan proses pembuatan atau pembangunan sistem atau aplikasi, dengan bantuan perangkat lunak dan perangkat keras yang sesuai dengan analisis kebutuhan. Pembuatan sistem ini dilakukan pada perangkat lunak *Visual Basic 2010 Express* dengan menggunakan bahasa *basic* dan *Microsoft Access* sebagai pengolah *database*.

Kemudian setelah tahapan pengkodean (*coding*) dilakukan pengujian terhadap kode program tersebut. Pengujian *statement* dari kode dilakukan juga selama proses pengkodean berlangsung agar mengetahui apakah fungsi kode tersebut berjalan dengan semestinya.

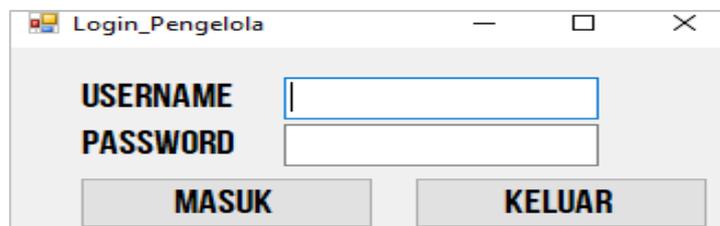
### Tampilan Awal (*Form Home*)



Gambar 9. *Form Home*

Saat aplikasi atau sistem dijalankan tampilan atau *form* yang akan muncul pertama kali adalah *form home*. Tampilan dari *form home* dapat dilihat pada Gambar 9, pada *form* ini menggunakan *ToolStrip* untuk menampilkan menu pada *form*. Setiap menu tersebut memiliki fungsi yang berbeda dan *coding* yang berbeda.

### *Form Login* Pengelola

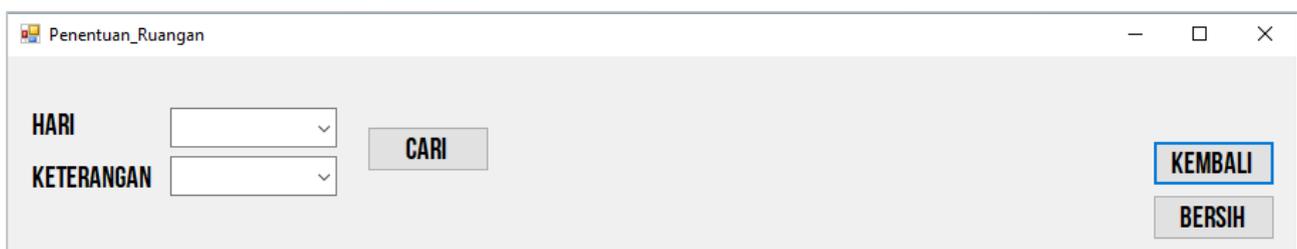


Gambar 10. *Form Logging* Pengelola

*Form* ini akan muncul saat meng-klik menu pengelola pada *form home*, setelah berhasil *login* maka akan menampilkan antara *form* kelola jadwal atau kelola ruangan. *Form* ini bertujuan agar tidak semua *user* dapat mengakses beberapa *form* kelola jadwal dan kelola ruangan.

*Form login* menggunakan fungsi percabangan untuk menentukan apakah *username* dan *password* tersebut untuk menampilkan *form* kelola jadwal atau *form* kelola ruangan, jika *username* dan *password* salah maka tidak dapat masuk ke sistem.

### *Form Penentuan* Ruang



Gambar 11. *Form Penentuan* Ruang

Gambar 11 merupakan tampilan yang muncul saat mengakses *form* penentuan ruangan. *Form* ini memiliki fungsi untuk melakukan proses menentukan ruangan untuk setiap jadwal, dengan menarik data jadwal yang diinginkan.

*Form* ini akan menarik data pada tabel jadwal dalam *database* untuk ditampilkan ke dalam *DataGridView* di *form*. Data yang ditarik sesuai dengan keinginan seperti dapat dilihat pada Kotak Kode 1 yang memberikan perintah menampilkan beberapa data yang ada di dalam tabel yang memiliki hari dan keterangan yang sama dengan *text* pada *combobox* hari dan keterangan yang dipilih *user*.

```
da = New OleDbDataAdapter("SELECT ID, Jam_Mulai, Jam_Selesai, Jumlah_Orang, Ruangan FROM
Jadwal WHERE Hari = '" & cbhari.Text & "' And Ruangan = '" & cbket.Text & "'",
connection)
ds = New DataSet
da.Fill(ds, "Jadwal")
dgvten.DataSource = ds.Tables("Jadwal")
```

### Kode 1. Coding Untuk Menarik Data

Setelah *user* memilih syarat pada *combobox* dan meng-klik *button* cari, maka tampilan *form* akan seperti gambar 12.

The screenshot shows a Windows application window titled 'Penentuan\_Ruangan'. It contains two dropdown menus: 'HARI' with 'Senin' selected and 'KETERANGAN' with 'Pagi' selected. There are three buttons: 'CARI' (highlighted with a blue border), 'KEMBALI', and 'BERSIH'. Below the buttons is a 'TENTUKAN' button. At the bottom, a *DataGridView* table displays search results:

ID	Jam_Mulai	Jam_Selesai	Jumlah_Orang	Ruangan
M01	30/12/1899 8:00	30/12/1899 10:00	35	
M02	30/12/1899 8:00	30/12/1899 11:00	40	
M03	30/12/1899 8:27	30/12/1899 10:27	32	
*				

Gambar 12. *Form* Penentuan Ruangan Setelah Menarik Data

Data yang ditarik dari *database* ditampilkan menggunakan *DataGridView*, tidak hanya menarik data tetapi *button* cari juga berfungsi untuk menampilkan *button* tentukan untuk memberikan perintah mencari kelas yang sesuai dengan jadwal yang ada pada *DataGridView* tersebut.

Pada *button* tentukan memberikan perintah untuk menentukan ruangan mana yang sesuai dengan mengambil data ruangan pada tabel ruangan. *Button* tentukan menggunakan fungsi *looping* dan juga fungsi percabangan IF-ELSEIF-ELSE.

Fungsi *looping* digunakan untuk mengecek ke dalam seluruh baris pada *DataGridView* ruangan mana yang sudah digunakan agar tidak ada ruangan yang sama di dalam *DataGridView* tersebut. Sedangkan fungsi percabangan untuk menentukan tabel ruangan mana yang digunakan untuk proses penentuan ruangan. Tabel yang digunakan sesuai dengan hari yang dipilih pada *combobox* sebelumnya, jika hari Senin maka tabel yang digunakan adalah tabel ruangan hari Senin.

Ketika *button* tentukan diklik maka dengan otomatis menampilkan kode ruangan pada kolom ruangan yang adapada *DataGridView*, dan juga menampilkan *button* simpan. Tampilan *form* setelah *button* tentukan diklik ditunjukkan pada Gambar 13.

ID	Jam_Mulai	Jam_Selesai	Jumlah_Orang	Ruangan
M01	30/12/1899 8:00	30/12/1899 10:00	35	B202
M02	30/12/1899 8:00	30/12/1899 11:00	40	B204
M03	30/12/1899 8:27	30/12/1899 10:27	32	B205
*				

**Gambar 13. Form Penentuan Ruang Setelah Klik Tentukan**

Button simpan memberikan perintah untuk menyimpan data yang ada di dalam *DataGridView* ke *database*. Button simpan juga menggunakan fungsi *looping* dan percabangan, hampir sama dengan kegunaan fungsi *looping* dan percabangan di *button* tentukan. Pada *button* simpan fungsi *looping* digunakan untuk melakukan perulangan untuk menyimpan setiap baris pada *DataGridView* ke *database*.

Fungsi percabangan yang digunakan yaitu IF-ELSEIF- ELSE, untuk melakukan perubahan. Misalnya data ruangan yang digunakan merupakan data pada tabel ruangan Senin maka saat data tersebut disimpan secara otomatis sistem akan memberikan perintah *update* juga untuk mengubah *field* jam pada tabel ruangan. *Coding* yang digunakan pada *button* simpan ini bukan *insert* tetapi *update* karna mengubah *field* ruangan saja pada tabel jadwal.

Jika data berhasil tersimpan maka akan muncul *messagebox* sebagai pemberitahuan bahwa data sudah berhasil tersimpan. Dan setelah itu *DataGridView* akan menampilkan data gabungan dari tabel ruangan dan tabel jadwal ditunjukkan pada Gambar 14.

**Gambar 14. Form Penentuan Ruang Setelah Klik Simpan**

The screenshot shows a web application window titled "Penentuan\_Ruangan". It features two dropdown menus: "HARI" set to "Senin" and "KETERANGAN" set to "Pagi". There are buttons for "CARI", "KEMBALI", "BERSIH", and "TENTUKAN". Below these is a table with the following data:

ID	Jam_Mulai	Jam_Selesai	Jumlah_Orang	Ruangan	Hari	Kuota
M01	30/12/1899 8:00	30/12/1899 10:00	35	B202	Senin	35
M02	30/12/1899 8:00	30/12/1899 11:00	40	B204	Senin	40
M03	30/12/1899 8:27	30/12/1899 10:27	32	B205	Senin	38
*						

At the bottom of the window is a "SIMPAN" button.

Gambar 15. Form Penentuan Ruang setelah Klik OK Pada Pemberitahuan

Menarik data dari tabel ruangan yaitu daya tampung dari ruangan dan mengambil seluruh data dari tabel jadwal yaitu ID, Jam\_Mulai, Jam\_Selesai, Jumlah\_Orang, Hari dan Ruangan, dapat dilihat pada Gambar 15. Untuk coding lengkap pada form penentuan ruangan dapat dilihat pada lampiran.

### Form Info Ruang

The screenshot shows a web application window titled "Info Ruang Kosong". It displays room availability for each day of the week (SENIN, SELASA, RABU, KAMIS, JUMAT, SABTU). Each day has a table with columns: Kode\_Ruangan, Jam\_Isi, and Kuota. A "KEMBALI" button is visible on the right side.

Day	Kode_Ruangan	Jam_Isi	Kuota
SENIN	B201		30
	B202	10:00:00	35
	B204	11:00:00	40
	B205	10:27:00	38
	a201		35
	A202		40
SELASA	B201		30
	B202		35
	B204		40
	B205		38
	a201		35
	A202		40
RABU	B201		30
	B202		35
	B204		40
	B205		38
	a201		35
	A202		40
KAMIS	B201		30
	B202		35
	B204		40
	B205		38
	a201		35
	A202		40
JUMAT	B201		30
	B202		35
	B204		40
	B205		38
	a201		35
	A202		40
SABTU	B201		30
	B202		35
	B204		40
	B205		38
	a201		35
	A202		40

Gambar 16. Form Info Ruang

Pada Gambar 16 merupakan tampilan saat form info ruangan dibuka, form ini menunjukkan informasi kelas setiap hari seputar jam kelas selesai digunakan dan juga daya tampung setiap kelas. Form ini menarik semua data tabel ruangan dan menampilkannya menggunakan data grid view. Contoh kode untuk menarik data dan menampilkan data ke DataGridView pada form info ruangan ditunjukkan pada kotak kode 2 dan untuk kode yang lengkap ditunjukkan pada lampiran.

```

Private Sub Info_Ruangan_Kosong_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    Call conn()
    da = New OleDbDataAdapter("SELECT * FROM RuanganSenin", connection)
    ds = New DataSet da.Fill(ds, "RuanganSenin")
    dgvsenin.DataSource = ds.Tables("RuanganSenin")

```

### Kode 2. Coding Menarik Dan Menampilkan Data Form Info Ruangan

#### Form Kelola Jadwal

ID	Matakuliah	SKS	Hari	Jam_Mulai	Jam_Selesai	Durasi	Jumlah_Orang	Ruangan
M01	RPL	4	Senin	30/12/1899 15:00	30/12/1899 18:40	220 Menit	30	Ruangan

Gambar 17. Form Kelola Jadwal

Gambar 17 adalah tampilan awal dari form kelola jadwal saat muncul pertama kali, untuk mengakses form ini user harus terlebih dahulu login. Pada form ini user dapat melakukan pengelolaan data di dalam tabel jadwal seperti *insert*, *select*, *update* dan *delete*.

Dan pada form ini juga memiliki fungsi pemberian kode otomatis untuk jadwal supaya tidak ada kode yang sama dan mengurangi *redudansi*. Adapun contoh kode untuk menyimpan data ke dalam database ditunjukkan pada Kotak Kode 3, sedangkan coding pada form kelola jadwal secara keseluruhan akan ditunjukkan pada lampiran. Tampilan form kelola jadwal saat berhasil menyimpan data ditunjukkan pada Gambar 18.

```

Dim str As String = "INSERT INTO Jadwal VALUES ('" & txtid.Text & "', '" & txtmatakuliah.Text & "', '" & SKS & "', '" & cbhari.Text & "', '" & dtpmulai.Text & "', '" & dtpakhir.Text & "', '" & txtdurasi.Text & "', '" & txtjmlorg.Text & "', '" & txtket.Text & "')"
cmd = New OleDbCommand(str, connection)
cmd.ExecuteNonQuery()

da = New OleDbDataAdapter("SELECT * FROM Jadwal", connection)
ds = New DataSet
da.Fill(ds, "Jadwal")
dgvjadwal.DataSource = ds.Tables("Jadwal")
MessageBox.Show("DATA BERHASIL TERSIMPAN")

```

### Kode 3. Coding Simpan Pada Form Kelola Jadwal

KELOLA JADWAL

MENU ▾

ID: M02

MATAKULIAH: Visual Basic

HARI: Senin ▾

JAM MULAI: 16:09

SKS:  2  3  4

JUMLAH ORANG: 30

RUANGAN:

SIMPAN

UBAH

HAPUS

KEMBALI

BERSIH

DATA BERHASIL TERSIMPAN

OK

ID	Matakuliah	SKS	Hari	Durasi	Jumlah_Orang	Ruangan
M01	RPL	4	Senin	30/12/1899 15:00 - 30/12/1899 18:40	220 Menit	30
M02	Visual Basic	3	Senin	30/12/1899 16:09 - 30/12/1899 18:54	165 Menit	30

Gambar 18. Form Kelola Jadwal Setelah Berhasil Disimpan

Button hapus. Yang berfungsi untuk menghapus data yang ada di dalam *database*, data yang dihapus sesuai dengan data yang dipilih oleh *user*. Proses menghapus data ditunjukkan Gambar 19, jika data berhasil dihapus maka akan muncul pemberitahuan.

KELOLA JADWAL

MENU ▾

ID: M02

MATAKULIAH: Visual Basic

HARI: Senin ▾

JAM MULAI: 16:09

SKS:  2  3  4

JUMLAH ORANG: 30

RUANGAN:

SIMPAN

UBAH

HAPUS

KEMBALI

BERSIH

DATA BERHASIL TERHAPUS

OK

ID	Matakuliah	SKS	Hari	Durasi	Jumlah_Orang	Ruangan
M01	RPL	4	Senin	30/12/1899 15:00 - 30/12/1899 18:40	220 Menit	30
M02	Visual Basic	4	Senin	30/12/1899 16:09 - 30/12/1899 19:49	220 Menit	30

Gambar 19. Form Kelola Jadwal Setelah Berhasil Menghapus Data

Fungsi *update* dalam *form* ini memberikan perintah mengubah isi *field* dari data yang dipilih semua *field* dapat diubah kecuali *id* dan ruangan. Sama seperti proses *insert* dan *delete*, proses *update* juga memberikan pemberitahuan jika data berhasil ter-*update*, ditunjukkan pada Gambar 20.

The screenshot shows the 'KELOLA JADWAL' application window. On the left, there is a 'MENU' dropdown. The main form contains several input fields: 'ID' (M02), 'MATAKULIAH' (Visual Basic), 'HARI' (Senin), 'JAM MULAI' (16:09), 'SKS' (radio buttons for 2, 3, and 4, with 4 selected), 'JUMLAH ORANG' (30), and 'RUANGAN'. To the right of the form are buttons: SIMPAN, UBAH (highlighted in blue), HAPUS, KEMBALI, and BERSIH. A small dialog box titled 'DATA BERHASIL TERUBAH' with an 'OK' button is overlaid on the form. Below the form is a table with columns: ID, Matakuliah, SKS, Hari, Durasi, Jumlah\_Orang, and Ruangan. The table contains two rows: M01 (RPL, 4 SKS, Senin, 30/12/1899 15:00 - 30/12/1899 18:40, 220 Menit, 30) and M02 (Visual Basic, 3 SKS, Senin, 30/12/1899 16:09 - 30/12/1899 18:54, 165 Menit, 30). The M02 row is highlighted in blue.

Gambar 20. *Form* Kelola Jadwal Setelah Berhasil Meng-*Update* Data

### *Form* Kelola Ruang

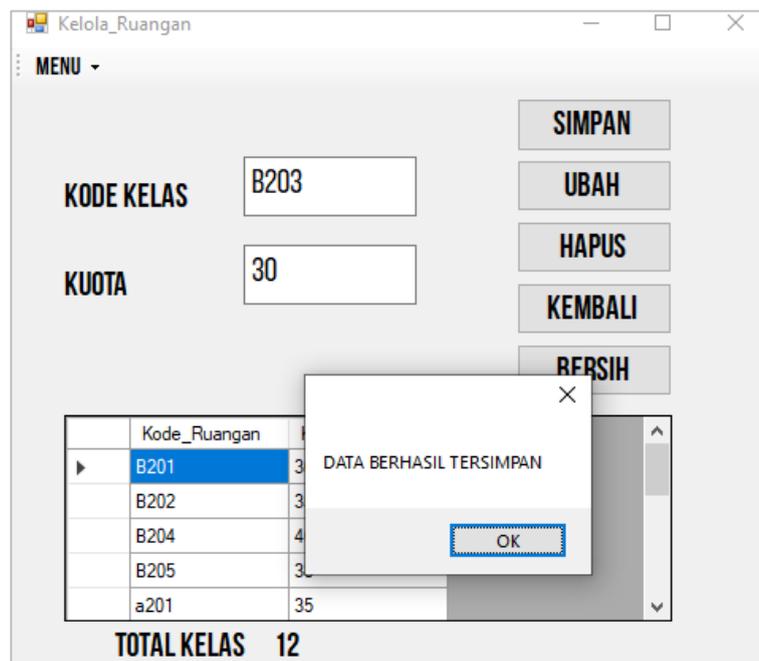
The screenshot shows the 'Kelola\_Ruangan' application window. It has a 'MENU' dropdown. The form contains two input fields: 'KODE KELAS' and 'KUOTA'. To the right of the form are buttons: SIMPAN, UBAH, HAPUS, and KEMBALI. Below the form is a table with columns: Kode\_Ruangan and Kuota. The table contains five rows: B201 (30), B202 (35), B203 (30), B204 (40), and B205 (38). The B201 row is highlighted in blue. Below the table, it says 'TOTAL KELAS 13'.

Gambar 21. *Form* Kelola Ruang

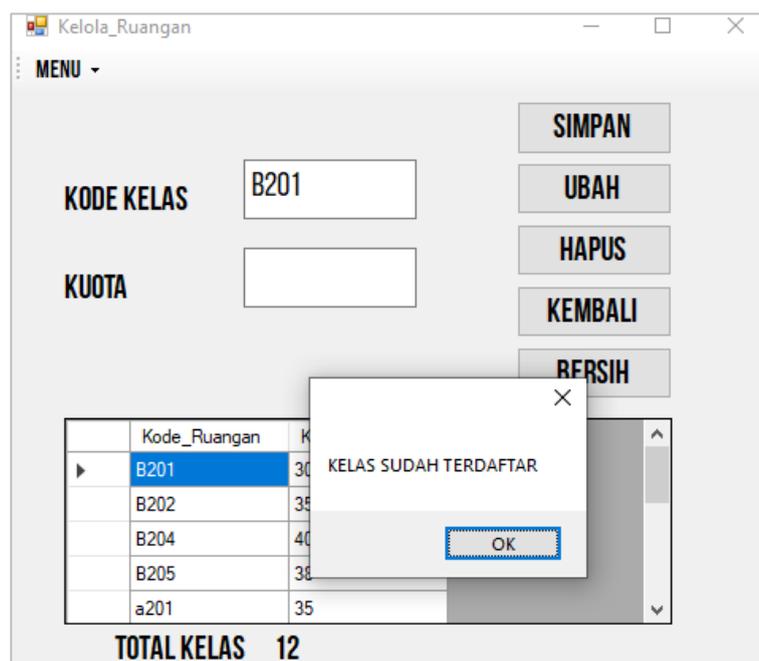
Gambar 21 menunjukkan tampilan *form* kelola ruangan saat pertama kali dibuka, *form* kelola ruangan memiliki proses yang hampir sama dengan fungsi pada *form* kelola jadwal. Perbedaan dari kedua *form* tersebut adalah pada tabel yang digunakan.

Pada *form* kelola ruangan menarik dan mengelola data pada tabel ruangan, dengan menggunakan fungsi *insert*, *select*, *update* dan *delete*. Agar *user* mengetahui apakah data berhasil ter-*insert* ke dalam *database coding* di atas juga memerintahkan untuk memunculkan pemberitahuan bahwa data berhasil di-*insert*. Tampilan *form* kelola ruangan saat berhasil melakukan proses *insert* data ditunjukkan pada Gambar 22.

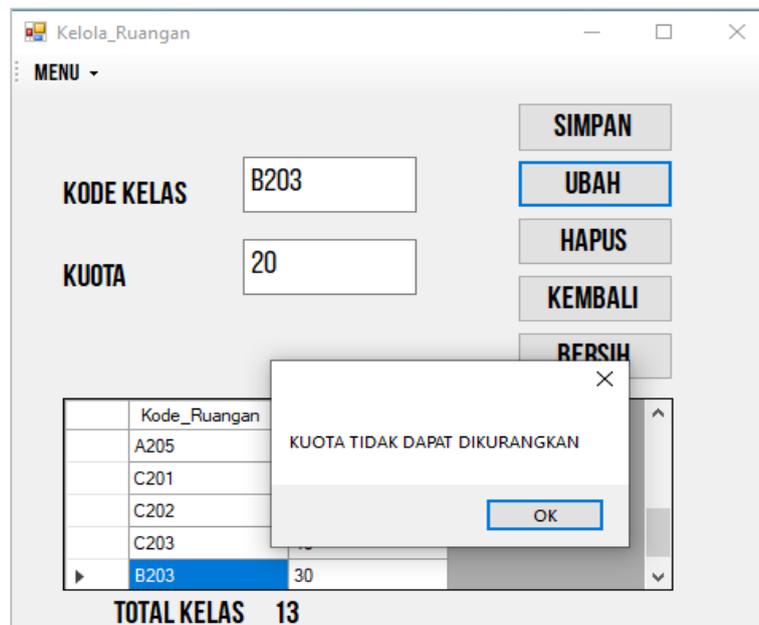
Saat *user* akan memasukkan data kode kelas jika kode kelas sudah terdaftar sebelumnya, maka secara otomatis sistem akan memunculkan pemberitahuan bahwa kelas tersebut sudah terdaftar sebelum *user* melakukan *insert* kembali, ditunjukkan pada Gambar 23.



Gambar 22. *Form* Kelola Ruangan Berhasil Meng-*Insert* Data



Gambar 23. *Form* Kelola Ruangan Saat Muncul Pemberitahuan Kode Ruangan

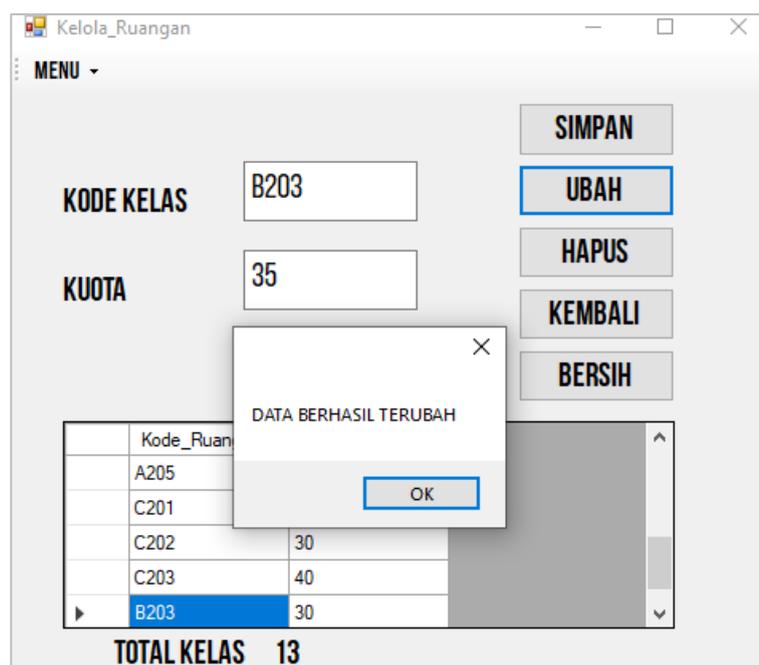


**Gambar 24.** Form Kelola Ruang Saat Muncul Pemberitahuan Kuota

Data ruangan hanya dapat dilakukan perubahan pada isi kolom daya tampung atau kuota kelas, tetapi daya tampung yang baru harus lebih besar dari daya tampung sebelumnya. Jika *user* memasukkan angka pada *textbox* kuota lebih kecil dari pada jumlah daya tampung sebelumnya, sistem akan memberikan pemberitahuan kuota tidak dapat dikurangkan dapat dilihat pada Gambar 24. Dan juga ada batas maksimal untuk kuota kelas, jika jumlah kuota yang dimasukkan lebih besar dari kuota maksimal maka sistem akan memunculkan pemberitahuan. Namun saat data berhasil diubah atau *ter-update* maka sistem juga memberikan pemberitahuan data berhasil *ter-update*, ditunjukkan pada Gambar 25. Contoh *coding update* pada *form* kelola ruangan ditunjukkan pada Kotak Kode 4 dan untuk *coding* lengkap akan ditunjukkan pada lampiran.

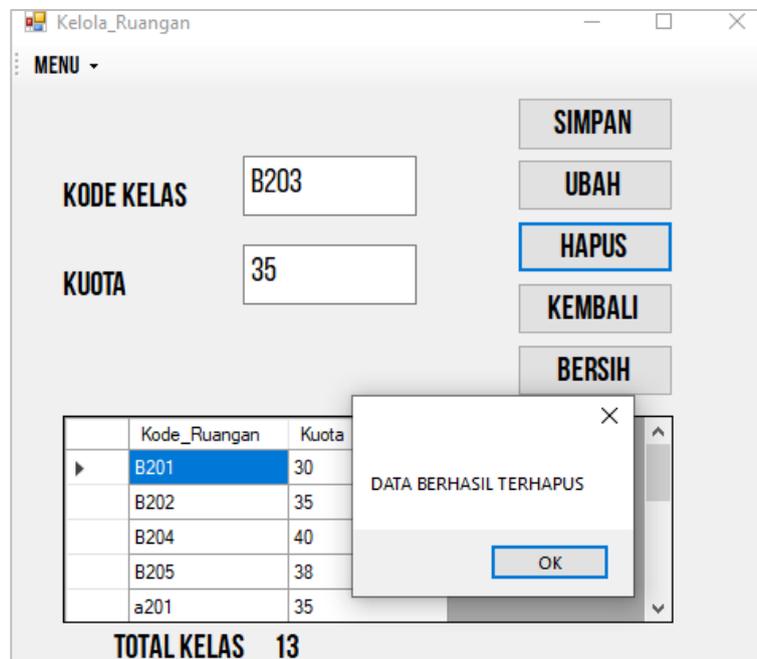
```
Dim update As String = "UPDATE Ruang Kuota = " & txtkuota.Text & " WHERE
Kode_Ruangan = " & txtkdcls.Text & ""
cmd = New OleDbCommand(update, connection)
cmd.ExecuteNonQuery()
```

**Kode 4.** Coding Update Data Pada Form Kelas Ruang



**Gambar 25.** Form Kelola Ruang Berhasil Mengubah Data

Fungsi *button* hapus di *form* kelola ruangan adalah menghapus data ruangan yang ada di dalam *database*. Data yang dihapus adalah data yang dipilih oleh *user* dengan meng-klik data yang akan dihapus pada *DataGridView* kemudian data akan secara otomatis mengisi *textbox* yang ada di *form* kelola ruangan. Pengkodean yang digunakan untuk memberi perintah menghapus data di dalam *database* ditunjukkan pada Kotak Kode 5 dan tampilan *form* ketika berhasil menghapus data dapat dilihat pada Gambar 26.



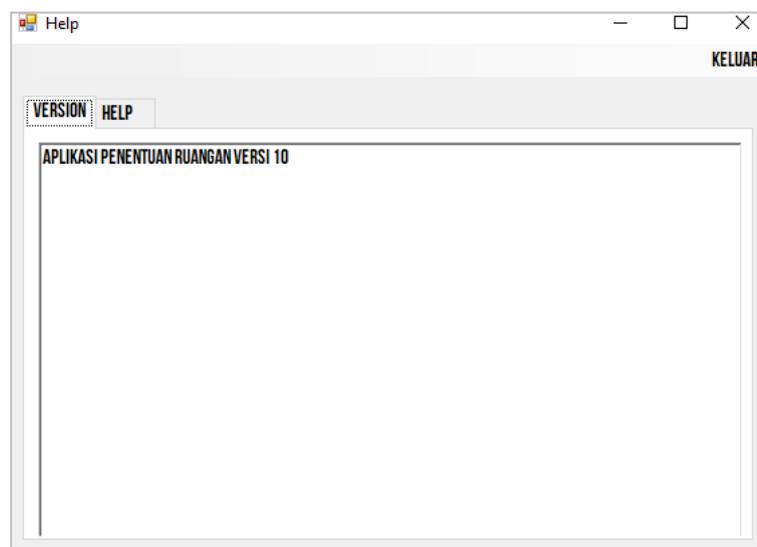
Gambar 26. *Form* Kelola Ruangan Berhasil Menghapus Data

```
Dim str As String = "DELETE FROM Ruangan WHERE Kode_Ruangan = '" & txtkdcls.Text & "'"
cmd = New OleDbCommand(str, connection)
cmd.ExecuteNonQuery()
```

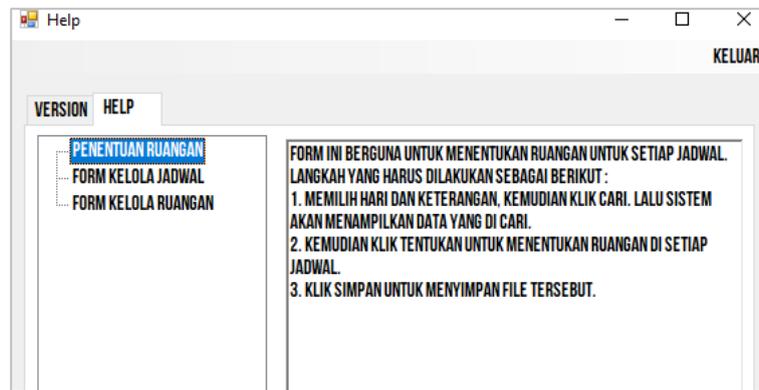
Kode 5. Coding Menghapus Data Pada *Form* Kelola Ruangan

### *Form Help*

*Form* ini berguna untuk menunjukkan versi dari aplikasi seperti yang ditunjukkan pada Gambar 27 dan beberapa langkah penggunaan *form* yang ditunjukkan pada Gambar 28.



Gambar 27. *Form Help* Versi Aplikasi



Gambar 28. Form Help Petunjuk Aplikasi

### Database

Sub bab ini menunjukkan koneksi sistem dengan *database* yang digunakan, perangkat lunak yang digunakan untuk *database* pada sistem ini adalah Microsoft Access. *Database* disimpan dengan format (.mdb) atau *database* access versi 2002 – 2007, sehingga fungsi yang digunakan pada pengkodean sistem adalah OleDb.

Untuk menghubungkan sistem dengan *database* yang dibuat, menggunakan *module* yang berisi *coding* koneksi *database*. Sehingga pada saat pengkodean *form* yang ingin terkoneksi dengan *database*, *user* hanya perlu memasukkan kode untuk memanggil fungsi pada *module* koneksi yang ada dalam sub conn.

### Struktur Tabel Database Tabel Data Jadwal

Field Name	Data Type
ID	Short Text
Matakuliah	Short Text
SKS	Number
Hari	Short Text
Jam_Mulai	Date/Time
Jam_Selesai	Date/Time
Durasi	Short Text
Jumlah_Orang	Number
Ruangan	Short Text
Keterangan	Short Text

Gambar 29. Struktur Tabel Jadwal

Gambar 29 menunjukkan *field* apa saja yang ada di dalam tabel jadwal dan apa tipe data yang digunakan di dalam tabel jadwal. Data yang ada di dalam tabel jadwal diolah di *form* kelola jadwal, setiap *user* melakukan proses *input* di *form* kelola jadwal maka akan secara otomatis data tersebut tersimpan ke dalam tabel jadwal. Sama dengan halnya *user* melakukan proses *delete* data pada *form* kelola jadwal maka data tersebut juga akan dihapus dari dalam tabel jadwal. Karena *form* kelola jadwal dan tabel jadwal saling terkoneksi. Data yang ada pada tabel jadwal ditunjukkan pada Gambar 30.

ID	Matakuliah	SKS	Hari	Jam_Mulai	Jam_Selesai	Durasi	Jumlah_Ora	Ruangan	Keterangan
M01	RPL	4	Senin	15:00	18:40	220 Menit	30		Siang

Gambar 30. Data pada tabel jadwal

### Tabel Data Ruang Keseluruhan

Field Name	Data Type
Kode_Ruangan	Short Text
Kuota	Number

Gambar 31. Struktur Tabel Ruang Keseluruhan

Struktur data pada tabel ruangan dapat dilihat pada Gambar 31 yang menunjukkan *field* yang ada di dalam tabel dan tipe data yang digunakan setiap *field*. Tabel ini terkoneksi dengan *form* kelola jadwal sehingga setiap proses yang dilakukan pada *form* akan secara otomatis terjadi di data yang ada di dalam tabel ruangan ini. Sehingga proses pengolahan data dilakukan melalui *form* kelola jadwal, Gambar 32 menunjukkan isi dari tabel ruangan yang berisi kode dari ruangan dan kuota atau daya tampung ruangan.

Kode_Ruangan	Kuota
a201	35
A202	40
a203	40
A204	35
A205	38
B201	30
B202	35
B204	40
B205	38
C201	25
C202	30
C203	40

**Gambar 32. Data Pada Tabel Ruangan Keseluruhan**

#### Tabel Data Ruangan Harian

Field Name	Data Type
Kode_Ruangan	Short Text
Jam_Isi	Short Text
Kuota	Number

**Gambar 33. Struktur Tabel Ruangan Harian**

Tabel ruangan harian berbeda dengan tabel ruangan keseluruhan, tabel ruangan harian merupakan data ruangan setiap harinya. Tabel ini berbeda isi datanya dengan tabel ruangan harian yang lain dan juga tabel ruangan keseluruhan. Tabel ruangan harian dibuat menurut hari misalnya ruangan Senin, ruangan Selasa, ruangan Rabu, ruangan Kamis, ruangan Jumat dan ruangan Sabtu. Semua tabel itu memiliki struktur yang sama tetapi menyimpan data yang berbeda, setiap data yang ingin disimpan ke tabel harian ini ditentukan oleh hari pada data tersebut. Tabel ini digunakan untuk melihat ruangan mana yang kosong dan kapan ruangan selesai digunakan pada hari tersebut. Isi data ruangan harian ditunjukkan pada Gambar 34.

Kode_Ruangan	Jam_Isi	Kuota
a201		35
A202		40
a203		40
A204		35
A205		38
B201		30
B202	10:00:00	35
B204	11:00:00	40
B205	10:27:00	38
C201		25
C202		30
C203		40

**Gambar 34. Data Pada Tabel Ruangan Harian**

### Black Box Testing

Pada tahap ini dilakukan pengujian terhadap sistem atau aplikasi yang dibuat, dengan tujuan untuk mengetahui sistem atau aplikasi yang dibuat sudah berjalan sesuai dengan prosedur dan tujuan sistem atau aplikasi tersebut dirancang.

### Pengujian Fungsional

**Tabel 4. Skenario Pengujian Sistem**

UJI FITUR	DETAIL PENGUJIAN	JENIS PENGUJIAN
<i>Form Login</i> Pengelola	Isi <i>form login</i>	<i>Black Box</i>
<i>Form</i> Kelola Jadwal	1. Menambahkan data Jadwal 2. Mengubah data Jadwal	<i>Black Box</i>
<i>Form</i> Kelola Ruang	1. Menambahkan data ruangan 2. Mengubah data ruangan 3. Menghapus data ruangan	<i>Black Box</i>
<i>Form</i> Info Ruang	Menampilkan data ruangan harian	<i>Black Box</i>
<i>Form</i> Penentuan Ruang	1. Menampilkan data jadwal 2. Menentukan ruangan untuk setiap jadwal	<i>Black Box</i>

### Kasus Dan Hasil Pengujian

**Tabel 5. Pengujian *Login* Pengelola**

KASUS DAN HASIL UJI DATA BENAR			
Data Masukan	Yang Diharapkan	Hasil	Kesimpulan
Mengisi Data <i>Login</i> <i>Username</i> : KelolaJadwal <i>Password</i> : 12345	Jika <i>username</i> dan <i>password</i> benar, maka <i>user</i> akan masuk ke dalam <i>form</i> kelola jadwal	Data <i>login valid</i> , sistem menampilkan <i>form</i> kelola jadwal	Diterima
Mengisi Data <i>Login</i> <i>Username</i> : KelolaRuang <i>Password</i> : 12345	Jika <i>username</i> dan <i>password</i> benar, maka <i>user</i> akan masuk ke dalam <i>form</i> kelola ruangan	Data <i>login valid</i> , sistem menampilkan <i>form</i> kelola ruangan	Diterima

**KASUS DAN HASIL UJI DATA SALAH**

Data Masukan	Yang Diharapkan	Hasil	Kesimpulan
<i>Username</i> dan <i>password</i> salah atau tidak diisi	Menampilkan pemberitahuan "GAGAL LOGIN"	Menampilkan pemberitahuan gagal	Diterima

**Tabel 6. Pengujian *Form* Kelola Jadwal**

KASUS DAN HASIL UJI DATA BENAR			
Skenario Pengujian	Yang Diharapkan	Hasil	Keterangan
Mengisi data jadwal pada <i>form</i> kelola jadwal dengan lengkap	Jika data sudah lengkap dan tepat, maka sistem akan memproses data untuk disimpan dan menampilkan pemberitahuan "Data Berhasil Tersimpan"	Data yang dimasukkan lengkap, sistem memproses data untuk disimpan dan menampilkan pemberitahuan data tersimpan	Berhasil
Memilih data pada <i>datagridview</i> kemudian mengganti isi data yang ingin diubah	Jika data sudah dipilih dan diubah dengan benar, maka sistem akan memproses data untuk menyimpan perubahan data dan menampilkan pemberitahuan "Data Berhasil Terubah"	Jika sudah memilih data dan mengubah isi data dengan benar, sistem akan memproses data untuk menyimpan perubahan data dan menampilkan pemberitahuan data terubah	Berhasil

KASUS DAN HASIL UJI DATA SALAH			
Data Masukan	Yang Diharapkan	Hasil	Kesimpulan
Tidak mengisi data dengan lengkap	Sistem akan menampilkan pemberitahuan "Data Belum Lengkap" dan tidak akan memproses data	Sistem menampilkan pemberitahuan data belum lengkap dan tidak memproses data	Berhasil

Tabel 7. Pengujian *Form* Kelola Ruang

KASUS DAN HASIL UJI DATA BENAR			
Skenario Pengujian	Yang Diharapkan	Hasil	Keterangan
Mengisi data ruangan pada <i>form</i> kelola ruangan dengan lengkap	Jika data sudah lengkap dan tepat, maka sistem akan memproses data untuk disimpan dan menampilkan pemberitahuan “Data Berhasil Tersimpan”	Data yang dimasukkan lengkap, sistem memproses data untuk disimpan dan menampilkan pemberitahuan data tersimpan	Berhasil
Memilih data pada <i>datagridview</i> kemudian mengganti isi data kuota dengan jumlah yang lebih besar dari sebelumnya	Jika data sudah dipilih dan diubah dengan benar, maka sistem akan memproses data untuk menyimpan perubahan data dan menampilkan pemberitahuan “Data Berhasil Terubah”	Jika sudah memilih data dan mengubah isi data dengan benar, sistem memproses data untuk menyimpan perubahan data dan menampilkan pemberitahuan data terubah	Berhasil
Memilih data pada <i>datagridview</i>	Jika data sudah dipilih maka sistem akan memproses untuk menghapus data dan akan menampilkan pemberitahuan “Data Berhasil Terhapus”	Jika sudah memilih data, maka sistem menghapus data dan menampilkan pemberitahuan data terhapus	Berhasil

## KASUS DAN HASIL UJI DATA SALAH

Tidak mengisi data ruangan pada <i>form</i> kelola ruangan dengan lengkap	Sistem akan menampilkan pemberitahuan “Data Belum Lengkap” dan data tersebut tidak diproses	Sistem menampilkan pemberitahuan data belum lengkap dan data tidak diproses	Berhasil
Memilih data pada <i>datagridview</i> kemudian mengganti isi data kuota dengan jumlah yang lebih kecil dari sebelumnya	Sistem akan menampilkan pemberitahuan “Kuota Tidak Dapat Dikurangkan”	Sistem akan menampilkan pemberitahuan kuota tidak dapat dikurangkan dan tidak dapat diproses	Berhasil
Memasukkan isi data kuota dengan jumlah yang melebihi batas	Sistem akan menampilkan pemberitahuan “Penambahan Kuota Melebihi Batas”	Sistem akan menampilkan pemberitahuan kuota tidak dapat dikurangkan dan tidak dapat diproses	Berhasil

Tabel 8. Pengujian *Form* Info Ruang

KASUS DAN HASIL UJI DATA BENAR			
Skenario Pengujian	Yang Diharapkan	Hasil	Keterangan
Saat <i>user</i> membuka <i>form</i> info ruangan	Jika <i>form</i> info ruangan dibuka maka akan menampilkan data dari tabel ruangan harian ke dalam <i>DataGridView</i>	Form menampilkan data ruangan harian ke <i>DataGridView</i> saat <i>form</i> dibuka	Berhasil

Tabel 9. Pengujian *Form* Penentuan Ruang

KASUS DAN HASIL UJI DATA BENAR			
Skenario Pengujian	Yang Diharapkan	Hasil	Keterangan
Memilih data pada <i>combobox</i> hari dan keterangan	Jika <i>combobox</i> hari dan keterangan sudah dipilih maka sistem akan memproses dan menampilkan data yang dicari ke dalam <i>DataGridView</i>	Jika kedua <i>combobox</i> diisi maka sistem akan memproses dan menampilkan data yang sesuai ke dalam <i>DataGridView</i>	Berhasil
Setelah muncul data pada <i>DataGridView</i> , tentukan ruangan untuk setiap jadwal	Jika data sudah tampil pada <i>DataGridView</i> , maka sistem akan memproses untuk menentukan ruangan dan memasukkannya ke dalam kolom ruangan yang ada pada <i>DataGridView</i> dengan tidak ada kesamaan ruangan	Jika data sudah tampil pada <i>DataGridView</i> , sistem akan memproses untuk menentukan ruangan yang berbeda satu dengan yang lain dan memasukkannya ke dalam kolom ruangan pada <i>DataGridView</i>	Berhasil
Menyimpan data yang ada di dalam <i>DataGridView</i>	Jika sudah terisi semua kolom ruangan, sistem akan memproses data untuk	Jika kolom ruangan sudah terisi semua, maka sistem akan	Berhasil

	disimpan dan memunculkan pemberitahuan “Data Berhasil Tersimpan”	memproses data untuk disimpan dan menampilkan pemberitahuan berhasil tersimpan	
--	--	--	--

#### Kasus dan Hasil Uji Data Salah

Tidak memilih isi pada <i>combobox</i> hari dan keterangan	Sistem tidak akan menampilkan data apa pun dan data tidak dapat diproses lebih lanjut	Sistem tidak menampilkan data apa pun sehingga tidak ada data yang dapat diproses	Berhasil
--	---	---	----------

#### IV. PENUTUP

Berdasarkan dari penjelasan-penjelasan dalam setiap bab dan sub bab sebelumnya, maka dapat diambil kesimpulan untuk penelitian ini sebagai berikut:

- 1) Program atau sistem dapat menentukan kelas mana yang sesuai pada setiap jadwal sesuai dengan jumlah orang dan kuota ruangan.
- 2) Sistem menggunakan metode *best first fit* dalam menentukan ruangan yang tepat untuk setiap jadwal.

#### DAFTAR PUSTAKA

- Anisah, Anisah, and Melati Suci Mayasari. 2016. “Desain Database Sistem Informasi Penerimaan Peserta Didik Baru Pada Selective English Course.” *Simetris: Jurnal Teknik Mesin, Elektro Dan Ilmu Komputer* 7 (1): 183. <https://doi.org/10.24176/simet.v7i1.503>.
- Cholifah, Wahyu Nur, Yulianingsih Yulianingsih, and Sri Melati Sagita. 2018. “Pengujian Black Box Testing Pada Aplikasi Action & Strategy Berbasis Android Dengan Teknologi Phonegap.” *STRING (Satuan Tulisan Riset Dan Inovasi Teknologi)* 3 (2): 206. <https://doi.org/10.30998/string.v3i2.3048>.
- Dharwiyanti, Sri, and Romi Wahono Satria. 2003. “Pengantar Unified Modeling Language (UML),” 1–13.
- Hidayat, Taufik, and Mahmudin Muttaqin. 2018. “Pengujian Sistem Informasi Pendaftaran Dan Pembayaran Wisuda Online Menggunakan Black Box Testing Dengan Metode Equivalence Partitioning Dan Boundary Value Analysis.” *Jurnal Teknik Informatika UNIS JUTIS* 6 (1): 2252–5351. [www.ccsenet.org/cis](http://www.ccsenet.org/cis).
- Jaya, Tri Snadhika. 2018. “Pengujian Aplikasi Dengan Metode Blackbox Testing Boundary Value Analysis (Studi Kasus: Kantor Digital Politeknik Negeri Lampung).” *Jurnal Informatika Pengembangan IT (JPIT)* 3 (2): 45–46. <https://doi.org/10.30591/jpit.v3i1.647>.
- Khotijah, Siti. 2016. “Perancangan Database E-Learning Manajemen System Untuk Pembelajaran Pada Sekolah Menengah Pertama.” *STRING (Satuan Tulisan Riset Dan Inovasi Teknologi)* 1 (1): 65–73. <https://doi.org/10.30998/string.v1i1.970>.
- Mardainis, Mardainis, and Khusaeri Andesa. 2017. “Integrasi Pemrograman Web Pada Pemrograman Desktop Sebagai Alternatif Fasilitas Laporan Dalam Pengembangan Program Aplikasi.” *Digital Zone: Jurnal Teknologi Informasi Dan Komunikasi* 8 (2): 104–13. <https://doi.org/10.31849/digitalzone.v8i2.635>.
- Nurhasan, Usman, Hadi Suyono, and M. Muslim. 2014. “Sistem Cerdas Pencarian Lokasi Parkir Terbaik Dengan Algoritma Fuzzy-Best Fit.” *Jurnal EECCIS* 8 (1): pp.7-12.
- Nurhayati, Sri, and Ryan Zulmi. 2014. “Sistem Monitoring Distribusi Bantuan Bencana Alam Berbasis Website Menggunakan Metode Algoritma First-Fit.” *Jurnal Teknik Komputer Unikom* 3 (2): 18–22.
- Oktarina, N., and A. Kuswanto. 2011. “Pembelajaran Berbasis It Aplikasi Program Ms.Acces Untuk Meningkatkan Pemahaman Mahasiswa Pada Pokok Bahasan Inventarisasi.” *Jurnal Penelitian Pendidikan Unnes* 28 (2): 125871. <https://doi.org/10.15294/jpp.v28i2.5637>.
- Saragih, Richy R. 2018. “Pemrograman Dan Bahasa Pemrograman,” no. December. <https://www.researchgate.net/publication>.
- Suendri. 2018. “Implementasi Diagram UML (Unified Modelling Language) Pada Perancangan Sistem

Informasi Remunerasi Dosen Dengan Database Oracle (Studi Kasus: UIN Sumatera Utara Medan)”  
6341 (November): 1–9.

Swara Yoga, Ganda, and Yunes Pebriadi. 2016. “Rekayasa Perangkat Lunak Pemesanan Tiket Bioskop Berbasis Web.” *Jurnal TEKNOIF* 4 (2): 27–39.

Syafei, Henderi. 2016. “OBJECT ORIENTED MODELLING WITH UNIFIED MODELING LANGUAGE (UML),” no. June.

<https://doi.org/10.13140/RG.2.1.3464.4088>.

Wiliani, Ninuk, and Syadid Zambani. 2017. “Rancang Bangun Aplikasi Kasir Nonton Bola Breng Pada X Kasir Di Suatu Lokasi X Dengan Dengan Visual Basic 2010 Dan MySql” 6 (2): 77–83.