

**LAPORAN PROYEK AKHIR
STRUKTUR DATA NON LINEAR
“Pendaftaran Misa”**

Dosen Pengampu :
Kartono Pinaryanto, S.T.,M.Cs.



Disusun Oleh : Kelompok 1

Bernardus Gandhang Panji Wardhani	165314074
Maria Ribka Restu Sukma Ningsih	205314044
Ario Tua Purba	205314094
Ong Gabriel Riverine Susanto	205314111

**PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS SANATA DHARMA
YOGYAKARTA
2022**

A. Latar Belakang

Covid-19 muncul pada tahun 2019 dan masuk ke Indonesia pertama kali pada tanggal 2 maret 2020. Penyebarannya sangat cepat karena gejala umum virus ini tidak ada bedanya dengan flu sehingga orang tidak tahu bahwa ia sudah terpapar virus tersebut. Hal ini menyebabkan berbagai bidang terdampak dan aktivitas yang berpotensi menyebabkan kerumunan dihentikan. Mulai dari sekolah sampai ibadah pun dilakukan secara daring. Di satu sisi kita terlindungi dari virus ini tetapi di sisi lain kita juga merasa tidak nyaman karena kita sudah terbiasa melakukan kegiatan-kegiatan tersebut secara luring.

Banyak dari kita yang merasakan ketika mengikuti pembelajaran atau misa online kita menjadi tidak fokus sama sekali, kita bisa saja sambil bermain HP atau bahkan sambil makan. Disini yang menjadi perhatian kami adalah ibadah yang seharusnya dilakukan dengan serius dan penuh penghayatan tetapi karena *online* malah disepelekan. Adanya orang-orang yang hadir secara fisik dan berdoa bersama kita mendorong kita untuk merasa malu jika melakukan hal lain seperti bermain HP atau makan di dalam gereja sehingga tidak bisa kita pungkiri bahwa kita membutuhkan orang lain untuk membantu kita mengikuti misa dengan sungguh-sungguh. Oleh karena itu kami mempunyai ide untuk membuat aplikasi pendaftaran misa yang memungkinkan umat terkhususnya umat katolik dapat mengikuti misa secara luring. Memang ujung-ujungnya akan tetap melibatkan orang banyak tetapi kuota misa akan dibatasi lewat aplikasi ini sehingga dari pihak gereja bisa mengatur jarak antar tempat duduk tiap umat agar umat tetap menerapkan protokol kesehatan.

B. Tujuan

Untuk memudahkan semua umat agar bisa ikut andil dalam melaksanakan ibadah secara langsung di Gereja dan tetap mengikuti protokol kesehatan dari pemerintah dengan cara mendaftar melalui aplikasi.

C. Manfaat

Manfaat dari program ini adalah untuk memudahkan umat dalam mendaftar ibadah di gereja melalui aplikasi dan juga memudahkan dalam administrasi untuk proses Penjadwalan misa.

D. Capture Program dan Kode

Berikut kami lampirkan potongan program yang berisi method-method penting yang kami gunakan dalam mendukung project.

1. Tree

```

13 public class Tree<E> {
14     private TreeNode<E> root;
15     String tampunginorder = "";
16
17     public Tree(TreeNode<E> root) {
18         this.root = root;
19     }
20
21     public Tree() {
22     }
23
24     public void insert(E data) {
25         root = insertHelper(data, root);
26     }
27
28     public TreeNode<E> insertHelper(E data, TreeNode<E> localRoot) {
29         if (localRoot == null) {
30             localRoot = new TreeNode(data);
31             return localRoot;
32         }
33         if (((Comparable) data).compareTo(localRoot.getData()) < 0 || ((Comparable) data).compareTo(localRoot.getData()) == 0) {
34             localRoot.setLeftNode(insertHelper(data, localRoot.getLeftNode()));
35             localRoot.getLeftNode().setParent(localRoot);
36         } else {
37
29     public TreeNode<E> insertHelper(E data, TreeNode<E> localRoot) {
30         if (localRoot == null) {
31             localRoot = new TreeNode(data);
32             return localRoot;
33         }
34         if (((Comparable) data).compareTo(localRoot.getData()) < 0 || ((Comparable) data).compareTo(localRoot.getData()) == 0) {
35             localRoot.setLeftNode(insertHelper(data, localRoot.getLeftNode()));
36             localRoot.getLeftNode().setParent(localRoot);
37         } else {
38             localRoot.setRightNode(insertHelper(data, localRoot.getRightNode()));
39             localRoot.getRightNode().setParent(localRoot);
40         }
41
42         return localRoot;
43     }
44
46     public TreeNode<E> getCurrent(E data) {
47         TreeNode bantu = root;
48         while (bantu != null) {
49             if (((Comparable) data).compareTo(bantu.getData()) == 0) {
50                 return bantu;
51             } else {
52                 if (((Comparable) data).compareTo(bantu.getData()) < 0) {
53                     bantu = bantu.getLeftNode();
54                 } else {
55                     bantu = bantu.getRightNode();
56                 }
57             }
58         }
59
60         return null;
61     }

```

```

69 public void preorderHelper(TreeNode<E> localRoot) {
70     if (localRoot != null) {
71         System.out.print(localRoot.getData() + " ");
72         preorderHelper(localRoot.getLeftNode());
73         preorderHelper(localRoot.getRightNode());
74     }
75 }
76
77 public int countNodes(TreeNode<E> r) {
78     if (r == null) {
79         return 0;
80     } else {
81         int jumlah = 1;
82         jumlah += countNodes(r.getRightNode());
83         jumlah += countNodes(r.getLeftNode());
84         return jumlah;
85     }
86 }
87
88
89 public boolean isEmpty() {
90     if (root == null) {
91         return true;
92     } else {
93         return false;
94     }
95 }
96
97 public String inorderHelper(TreeNode localRoot) {
98
99     if (localRoot != null) {
100         inorderHelper(localRoot.getLeftNode());
101         tampunginorder += " "+localRoot.getData();
102         inorderHelper(localRoot.getRightNode());
103     }
104     return tampunginorder;
105 }
106
107 public String inorderTraversal() {
108     String inorder = inorderHelper(root);
109     return inorder;
110 }
111
112 public boolean delete(E data) {
113     TreeNode<E> node = getCurrent(data);
114     boolean cek = false;
115     if (node != null) {
116         if (isLeaf(node.getData())) {
117             cek = delete0(node);
118         } else if (node.getRightNode() == null || node.getLeftNode() == null) {
119             cek = delete1(node);
120         } else {
121             cek = delete2(node);
122         }
123     }
124     return cek;
125 }

```

```

149 public boolean delete0(TreeNode<E> data) {
150     TreeNode parent = data.getParent();
151     if (data == root) {
152         root = null;
153         return true;
154     } else {
155         if (parent.getRightNode() == data) {
156             parent.setRightNode(null);
157             return true;
158         } else {
159             parent.setLeftNode(null);
160             return true;
161         }
162     }
163 }
164
166 public boolean delete1(TreeNode<E> data) {
167     TreeNode parent = data.getParent();
168     if (data == root) {
169         if (root.getLeftNode() == null) {
170             root = root.getRightNode();
171         } else {
172             root = root.getLeftNode();
173         }
174         return true;
175     } else {
176         if (parent.getLeftNode() == data) {
177             if (data.getLeftNode() != null) {
178                 parent.setLeftNode(data.getLeftNode());
179                 data.getLeftNode().setParent(parent);
180                 setNull(data);
181                 return true;
182             } else {
183                 parent.setLeftNode(data.getRightNode());
184                 data.getRightNode().setParent(parent);
185                 setNull(data);
186                 return true;
187             }
188         } else {
189             if (data.getLeftNode() != null) {
190                 parent.setRightNode(data.getLeftNode());
191                 data.getLeftNode().setParent(parent);
192                 setNull(data);
193                 return true;
194             } else {
195                 parent.setRightNode(data.getRightNode());
196                 data.getRightNode().setParent(parent);
197                 setNull(data);
198                 return true;
199             }
200         }
201     }
202 }
203
204

```

```

212 public boolean isLeaf(E data) {
213     TreeNode node = getCurrent(data);
214     if (node != null) {
215         if (node.getRightNode() == null && node.getLeftNode() == null) {
216             return true;
217         } else {
218             return false;
219         }
220     } else {
221         System.out.println(data + " tidak ditemukan");
222         return false;
223     }
224 }
225
226 public TreeNode<E> getPredecessor(TreeNode<E> node) {
227     if (getCurrent(node.getData()) != null) {
228         TreeNode bantu = node.getLeftNode();
229         while (bantu.getRightNode() != null) {
230             bantu = bantu.getRightNode();
231         }
232         return bantu;
233     } else {
234         return null;
235     }
236 }
237
238 public boolean delete2(TreeNode<E> hapus) {
239     TreeNode parentPre = getPredecessor(hapus).getParent();
240     TreeNode parentHapus = hapus.getParent();
241     TreeNode predecessor = getPredecessor(hapus);
242     TreeNode kiri, kanan;
243
244     if (parentPre != hapus) {
245         parentPre.setRightNode(null);
246         if (parentHapus == null) {
247
248             this.root = predecessor;
249             predecessor.setParent(null);
250             predecessor.setRightNode(hapus.getRightNode());
251             predecessor.setLeftNode(hapus.getLeftNode());
252             predecessor.getRightNode().setParent(predecessor);
253             predecessor.getLeftNode().setParent(predecessor);
254             return true;
255         }
256     }
257     if (parentHapus.getLeftNode() == hapus) {
258         parentHapus.setLeftNode(predecessor);
259         kiri = parentHapus.getLeftNode();
260         kiri.setParent(parentHapus);
261         kiri.setRightNode(hapus.getRightNode());
262         kiri.getRightNode().setParent(kiri);

```

```

263         kiri.setLeftNode(hapus.getLeftNode());
264         kiri.getLeftNode().setParent(kiri);
265         return true;
266     } else {
267         parentHapus.setRightNode(predecessor);
268         kanan = parentHapus.getRightNode();
269         kanan.setParent(parentHapus);
270         kanan.setLeftNode(hapus.getLeftNode());
271         kanan.getLeftNode().setParent(kanan);
272         kanan.setRightNode(hapus.getRightNode());
273         kanan.getRightNode().setParent(kanan);
274         return true;
275     }
276 } else {
277     if (parentHapus.getLeftNode() == hapus) {
278         parentHapus.setLeftNode(predecessor);
279         kiri = parentHapus.getLeftNode();
280         kiri.setParent(parentHapus);
281         return true;
282     } else {
283         parentHapus.setRightNode(predecessor);
284         kanan = parentHapus.getRightNode();
285         kanan.setParent(parentHapus);
286         return true;
287     }

```

2. TreeNode

```

19 public class TreeNode<E> {
20     private E Data;
21     private TreeNode<E> leftNode;
22     private TreeNode<E> rightNode;
23     private TreeNode<E> parent;
24
25     public TreeNode(E Data) {
26         this.Data = Data;
27     }
28
29     public E getData() {
30         return (E)Data;
31     }
32
33     public TreeNode<E> getParent() {
34         return parent;
35     }
36

```

3. Umat

```

14     public class Umat implements Comparable<Object>{
15
16     private String nama ;
17     private String alamat;
18     private String id;
19
20     public Umat() {
21     }
22
23     public Umat(String nama, String alamat, String id) {
24         this.nama = nama;
25         this.alamat = alamat;
26         this.id = id;
27     }
28
29
30     public Umat(String id) {
31         this.id = id;
32     }
33
34
35     @Override
36     public String toString() {
37         return nama + "\t\t" + alamat + "\t" + id + "\n";
38     }
39
40
41     @Override
42     public int compareTo(Object o) {
43         return id.compareTo(((Umat)o).getId());
44     }
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66

```

4. Jadwal

```

13     public class Jadwal {
14         private String hari;
15         private int kuota;
16         private String jam;
17         private String tanggal;
18         private String imam;
19         private Tree<Umat> umat;
20
21
22     public Jadwal(int kuota, String jam, String tanggal,String hari) {
23         this.kuota = kuota;
24         this.jam = jam;
25         this.tanggal = tanggal;
26         this.hari = hari;
27         umat = new Tree<Umat>();
28     }
29
30     public Jadwal() {
31         this(0,null,null,null);
32     }
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93

```


E. Penjelasan Umum

Program pendaftaran misa ini dibagi menjadi 2 tampilan, yaitu tampilan untuk admin dan tampilan untuk umat yang ingin mendaftar misa.

- Menu admin dapat diakses jika kita menginput username “Admin” dan password “2022”.

Terdapat beberapa menu yang bisa diakses admin yaitu :

- Home

Halaman ini berisi panduan penggunaan menu-menu yang bisa diakses admin, menu ini otomatis tampil ketika admin pertama kali login sehingga langsung bisa terbaca oleh admin.

- Cari data Umat

Menu ini digunakan oleh admin untuk mencari data umat yang sudah terdaftar pada jadwal misa tertentu. Untuk mencari data umat kita menggunakan id yang sudah dimiliki oleh umat yang diperoleh ketika mendaftar. Maksud dari pencarian ini adalah untuk validasi bahwa umat yang boleh masuk hanya yang sudah mendaftar saja karena bisa saja terjadi penambahan kuota yang tidak terdeteksi.

- Lihat Data Umat

Menu ini digunakan admin untuk melihat daftar umat yang sudah mendaftar pada misa tertentu. Menu ini akan menampilkan nama, alamat, dan juga kode/id yang sudah diterima umat ketika mendaftarkan dirinya pada misa tertentu. Jika jadwal belum ada maka keterangannya akan menampilkan “Belum ada data”.

- Tambah Jadwal Misa

Menu ini digunakan admin untuk menambahkan jadwal misa yang akan dilaksanakan. Inputan yang akan dilakukan admin adalah berupa hari, tanggal, waktu misa, kuota misa, serta pilihan imam yang tersedia. Disini kami juga menerapkan apa yang telah dipelajari di matakuliah IMK yaitu bagaimana cara meminimalisir kesalahan input dari user jadi untuk inputan tanggal, hari dan imam kami gunakan combo box, lalu untuk inputan yang diketikan langsung oleh user kami cantumkan format penulisan dan diberi warna merah agar menarik perhatian user. Ada sedikit variasi untuk memberikan arahan pada user yaitu memberikan keterangan pada kotak input.

- Hapus Jadwal Misa

Menu ini digunakan admin untuk menghapus jadwal misa yang sudah ada atau tersedia pada program. Cara kerjanya adalah admin bisa memilih jadwal mana yang akan dihapus dengan menginput nomor jadwal sesuai list

sehingga jika admin ingin menghapus salah satu jadwal admin dapat menggunakan menu ini.

- Kosongkan Jadwal Misa

Menu ini digunakan oleh admin untuk mengosongkan seluruh jadwal misa yang sudah ada atau tersedia.

- Logout

Tombol ini digunakan untuk keluar dari tampilan menu admin dan kembali ke tampilan awal. Saat kita mengklik tombol tersebut maka akan muncul dialog konfirmasi seperti dibawah ini untuk mengantisipasi jika sewaktu-waktu user salah mengklik tombol tersebut.

- Menu umat dapat diakses jika jadwal misa sudah tersedia, sedangkan jika jadwal misa tidak tersedia maka program akan menampilkan pesan untuk mengecek di lain waktu. Ketika diakses, menu umat ini akan menampilkan list jadwal misa, kuota yang dapat ditampung, serta Imam yang akan memimpin jalannya misa.

Pada menu umat terdapat 2 fitur yang ditawarkan yaitu :

- Daftar Misa

Menu ini digunakan umat untuk mendaftarkan dirinya pada misa yang tersedia. Ketika umat mengakses menu Daftar Misa maka program secara otomatis akan meminta umat untuk menginputkan nama, alamat, serta memilih jadwal misa yang tersedia. Jika pendaftaran berhasil maka umat akan mendapatkan kode pendaftaran, tetapi jika kuota sudah penuh maka pendaftaran gagal.

- Pembatalan Misa

Menu ini digunakan ketika umat ingin membatalkan misa yang sudah dipilih. Untuk membatalkan misa umat diminta untuk memasukkan kode misa yang diperoleh pada saat pendaftaran, kemudian akan dikonfirmasi lagi dengan cara menampilkan nama umat tersebut untuk memastikan bahwa umat tidak salah menghapus data. Jika data sudah benar maka akan dikonfirmasi sekali lagi apakah umat benar-benar ingin menghapus data atau tidak, jika ya maka data terhapus tetapi jika tidak maka data umat tidak jadi dihapus.

F. Penjelasan Khusus Terkait Materi SDNL (Struktur Data Non Linear)

Materi Struktur Data Non Linear yang kami gunakan pada program kami ini adalah materi pencarian pohon biner atau *Binary Search Tree* (BST).

Kami mengimplementasikan:

1. Method insert BST

Method ini kami gunakan ketika ingin menambahkan data umat yang mendaftar misa dan akan disimpan pada pohon biner yang akan dibentuk, data yang membangun.

2. Method Delete BST

Method ini kami gunakan ketika ingin menghapus data umat yang sudah tersimpan ke dalam pohon biner atau yang sudah mendaftar pada salah satu jadwal misa.

3. Method Inorder Traversal

Method ini kami gunakan ketika ingin menampilkan isi dari pohon biner atau pada program ini kami menampilkan daftar umat yang sudah mendaftarkan diri pada jadwal misa.

4. Method getCurrent atau Method Searching pada BST

Method ini kami gunakan ketika ingin mencari data umat yang sudah di tambahkan ke dalam pohon biner. Untuk pencariannya kami gunakan id dan karena parameternya objek yang kita tahu memiliki lebih dari 1 atribut maka kami mengimplementasikan interface Comparable pada kelas Umat.

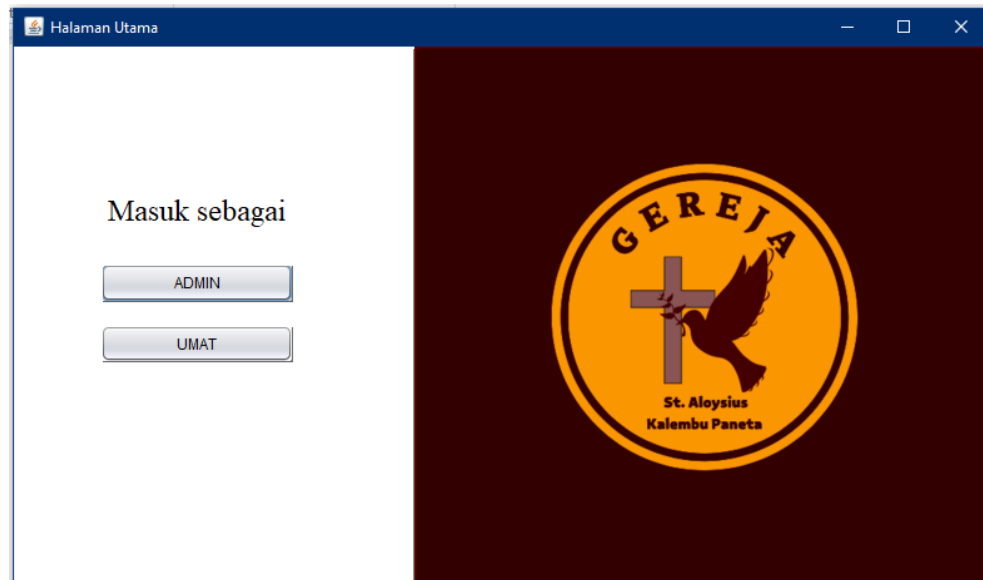
5. Method countNodes

Method ini kami gunakan ketika ingin menghitung jumlah dari nodes yang membangun pohon sehingga pada saat menambah data kita tidak melebihi kuota yang sudah ditentukan.

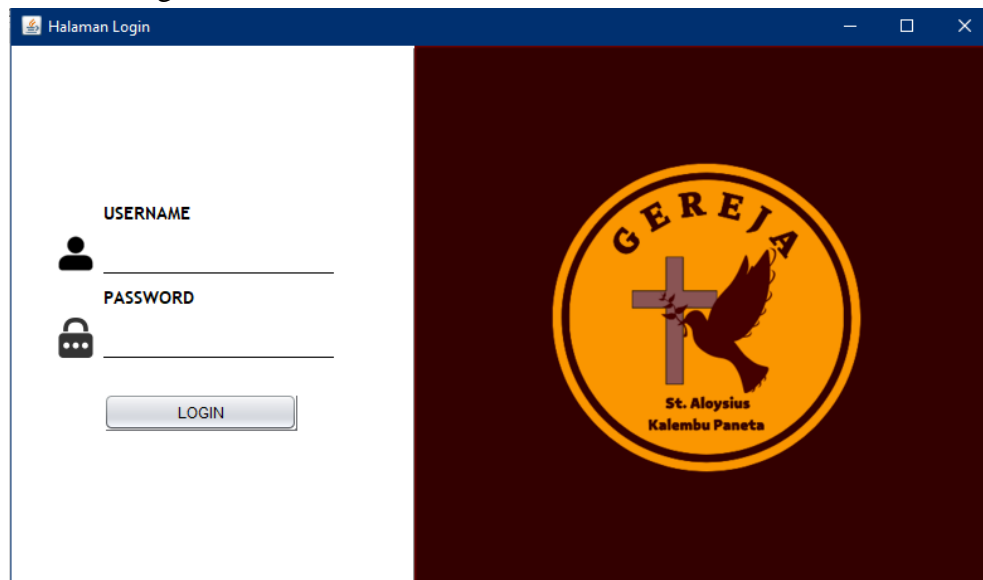
Untuk penyimpanan data digunakan arrayList dimana objek yang boleh ditampung hanyalah objek dari kelas jadwal saja. Didalam kelas jadwal kami gunakan komposisi dimana salah satu atributnya yaitu umat bertipe kelas Tree dan didalam kelas Tree tersebut kami menampung objek dari kelas Umat. Kelas Tree yang ada juga sudah dimodifikasi menggunakan *generic type* agar memudahkan untuk mengakses method dari objek yang ditampung. Tree dibangun menggunakan id umat, karena yang dibandingkan adalah objek maka kami mengimplementasikan interface Comparable ke kelas umat agar perbandingan dilakukan berdasarkan id.

G. Output

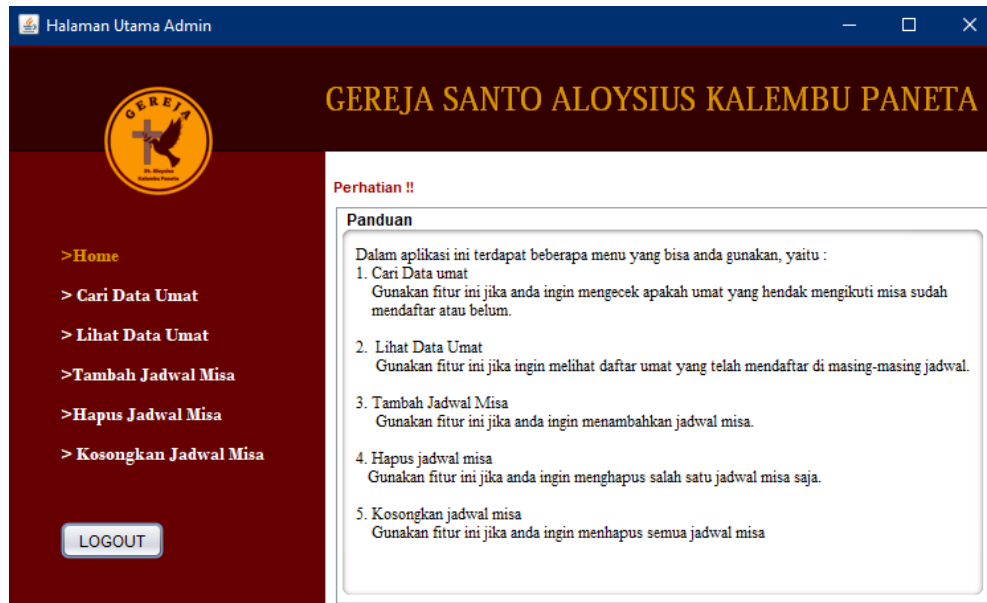
1. Halaman utama



2. Halaman login



3. Halaman utama admin



4. Cari data umat



5. Lihat data

The screenshot shows a web application window titled "Lihat Data Umat". The header features the logo of "GEREJA SANTO ALOYSIUS KALEMBU PANETA" on the left and the church's name in large yellow letters on a dark red background on the right. A sidebar on the left contains a menu with the following items: ">Home", "> Cari Data Umat", "> Lihat Data Umat" (highlighted in yellow), ">Tambah Jadwal Misa", ">Hapus Jadwal Misa", and "> Kosongkan Jadwal Misa". At the bottom of the sidebar is a "LOGOUT" button. The main content area is titled "Data Umat" and displays the message "Belum ada data" (No data yet) inside a white box with a thin border.

6. Tambah jadwal misa

The screenshot shows a web application window titled "Tambah Jadwal Misa". The header and sidebar are identical to the previous screenshot. The main content area contains a form for adding a mass schedule. The form includes the following fields and controls:

- "Pilih hari" (Select day) with a dropdown menu showing "Senin" (Monday).
- "Pilih Imam" (Select Priest) with a dropdown menu showing "Cyprianus Kuntoro Adi".
- A red text label "Format : pukul 20.00 WIB" above a text input field for "Masukkan waktu" (Enter time).
- "Masukkan tanggal" (Enter date) with three dropdown menus for day, month, and year, showing "01", "01", and "2026" respectively. A red text label "Format :DD-MM-YYYY" is to the right.
- "Masukkan Kuota" (Enter Quota) with a text input field containing the placeholder "Masukkan angka" (Enter number).
- A "SIMPAN" (Save) button at the bottom right of the form.

 A "LOGOUT" button is also present in the sidebar.

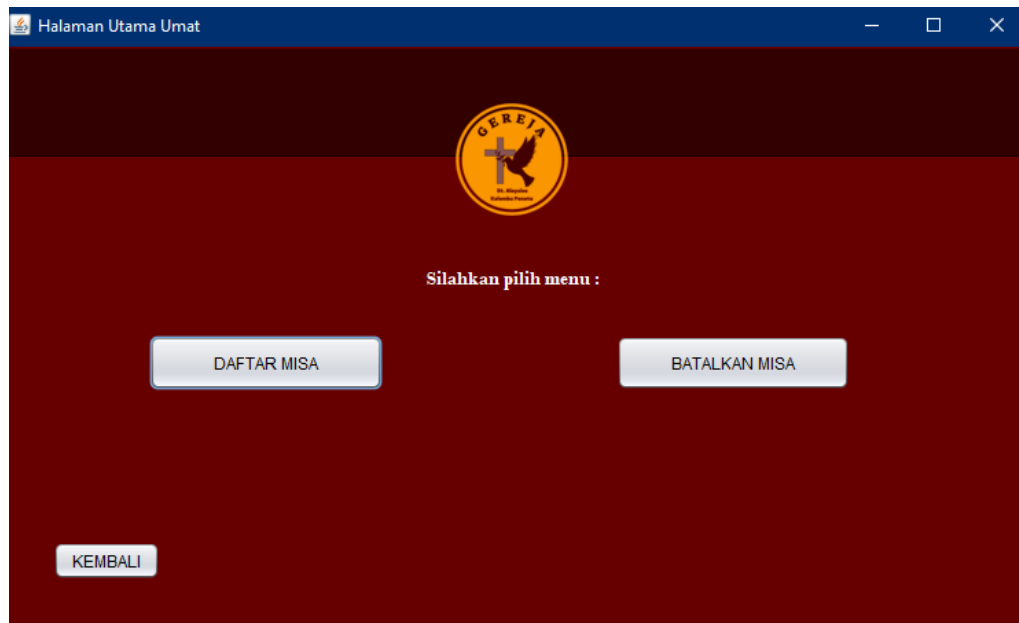
7. Hapus jadwal misa



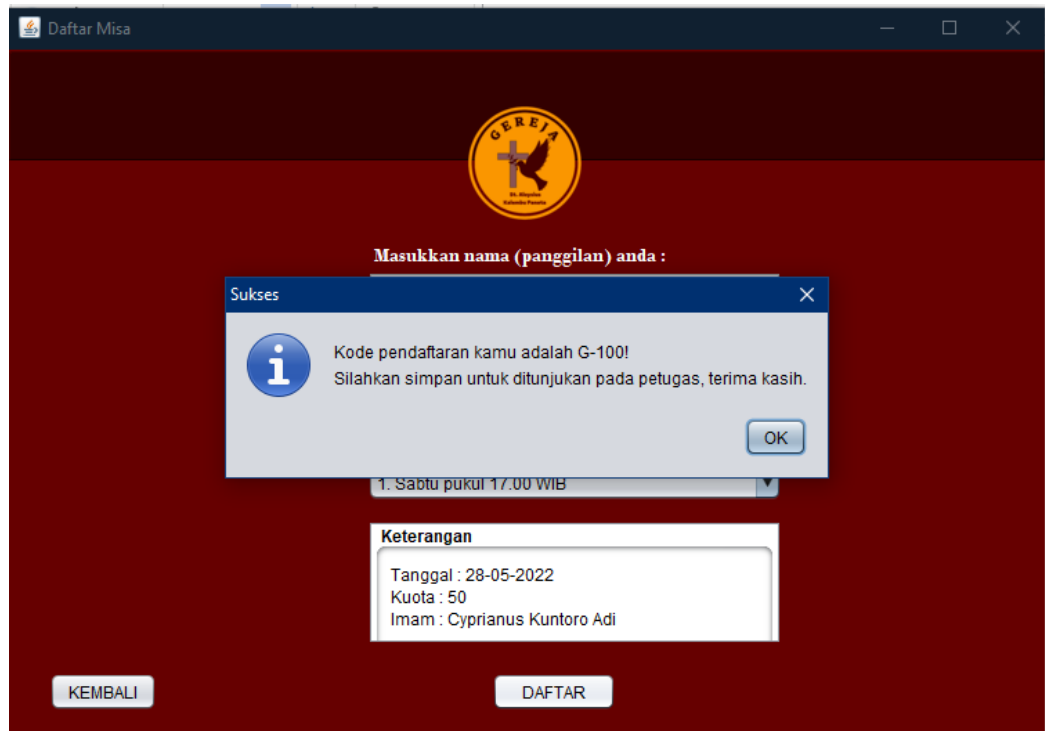
8. Kosongkan jadwal misa



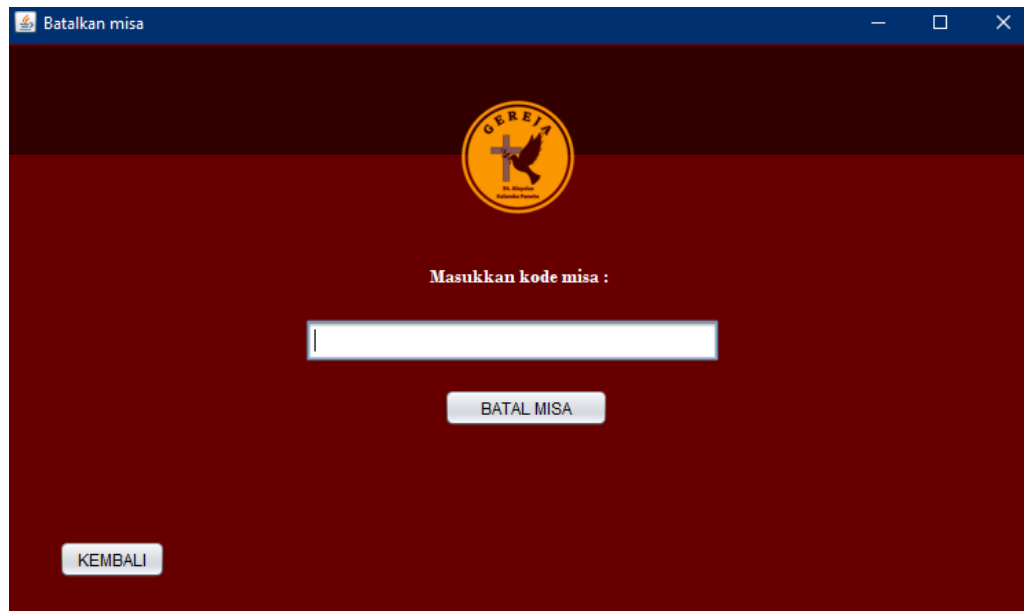
9. Halaman utama umat

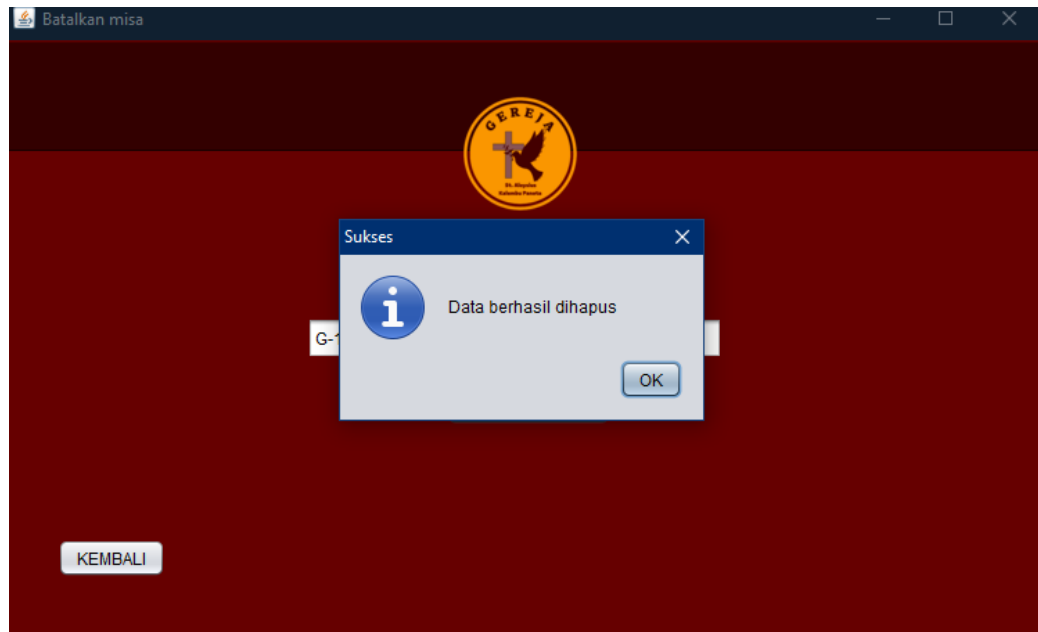


10. Daftar misa



11. Batalkan misa





H. Kesimpulan

Program pendaftaran misa ini dibuat atas dasar kepedulian kami terhadap dampak yang ditimbulkan oleh covid-19, kami ingin orang beribadah dengan nyaman dan sungguh-sungguh. Selain itu ilmu yang kami pelajari di perkuliahan bisa kami implementasikan dengan baik dan benar sehingga bisa berguna bagi orang lain.

Dengan adanya aplikasi ini diharapkan :

1. umat bisa terbantu jika ingin melaksanakan ibadah secara *offline*.
2. Pihak gereja dimudahkan dalam pengelolaan data misa.
3. Pemerintah dibantu untuk meminimalisir penularan covid-19 walaupun kegiatannya *offline*.