

ACTIVITY PERTEMUAN 6

NAMA : Felix Windriyareksa Hardyan

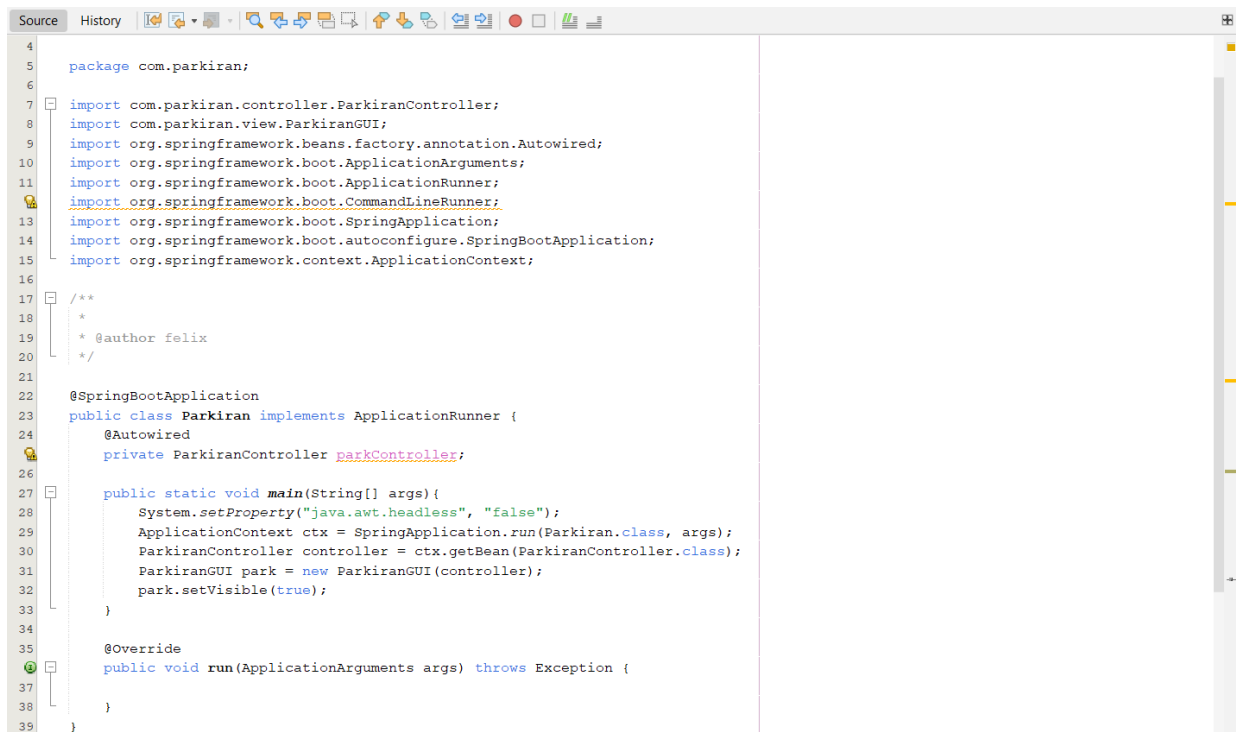
NPM : 50421506

KELAS : 4IA12

MATERI : Spring Architecture

MATA PRAKTIKUM : Rekayasa Perangkat Lunak 2

1. Parkiran.java



```
4
5 package com.parkiran;
6
7 import com.parkiran.controller.ParkiranController;
8 import com.parkiran.view.ParkiranGUI;
9 import org.springframework.beans.factory.annotation.Autowired;
10 import org.springframework.boot.ApplicationArguments;
11 import org.springframework.boot.ApplicationRunner;
12 import org.springframework.boot.CommandLineRunner;
13 import org.springframework.boot.SpringApplication;
14 import org.springframework.boot.autoconfigure.SpringBootApplication;
15 import org.springframework.context.ApplicationContext;
16
17 /**
18  *
19  * @author felix
20  */
21
22 @SpringBootApplication
23 public class Parkiran implements ApplicationRunner {
24     @Autowired
25     private ParkiranController parkController;
26
27     public static void main(String[] args) {
28         System.setProperty("java.awt.headless", "false");
29         ApplicationContext ctx = SpringApplication.run(Parkiran.class, args);
30         ParkiranController controller = ctx.getBean(ParkiranController.class);
31         ParkiranGUI park = new ParkiranGUI(controller);
32         park.setVisible(true);
33     }
34
35     @Override
36     public void run(ApplicationArguments args) throws Exception {
37     }
38
39 }
```

2. ParkiranController.java

```
Source History
4  */
5  package com.parkiran.controller;
6
7  import com.parkiran.model.ModelParkiran;
8  import com.parkiran.repository.ParkiranRepository;
9  import com.parkiran.service.ParkiranService;
10 import java.time.Duration;
11 import java.time.LocalDateTime;
12 import java.util.List;
13 import java.util.Optional;
14 import java.util.Scanner;
15 import org.springframework.beans.factory.annotation.Autowired;
16 import org.springframework.stereotype.Controller;
17
18 /**
19  *
20  * @author felix
21  */
22
23 @Controller
24 public class ParkiranController {
25     @Autowired
26     private ParkiranService svc;
27
28     public void masukParkir(ModelParkiran park) {
29         svc.masukParkir(park);
30     }
31
32     public String keluarParkir(String platNomor) {
33         return svc.keluarParkir(platNomor);
34     }
35
36     public List<ModelParkiran> getAllParkiran() {
37         return svc.getAllParkiran();
38     }
39 }
```

3. ModelParkiran.java

```
Source History
1  */
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4  */
5  package com.parkiran.model;
6  import jakarta.persistence.*;
7  import java.time.LocalDateTime;
8
9  /**
10   *
11   * @author felix
12   */
13
14 @Entity
15 @Table(name = "parkiran")
16 public class ModelParkiran {
17     @Id
18     @GeneratedValue(strategy = GenerationType.IDENTITY)
19     @Column(name = "id")
20     private int id;
21
22     @Column(name = "plat_nomor", nullable = false, length = 8)
23     private String platNomor;
24
25     @Column(name = "waktu_masuk", nullable = false)
26     private LocalDateTime waktuMasuk;
27
28     @Column(name = "jenis_kendaraan", nullable = false)
29     private int jenisKendaraan;
30
31
32     public ModelParkiran() {
33
34     }
35 }
```

```
Source History
36 public ModelParkiran(String platNomor, LocalDateTime waktuMasuk, int jenisKendaraan){
37     this.platNomor = platNomor;
38     this.waktuMasuk = waktuMasuk;
39     this.jenisKendaraan = jenisKendaraan;
40 }
41
42 public int getId() {
43     return id;
44 }
45
46 public void setId(int id) {
47     this.id = id;
48 }
49
50 public String getPlatNomor() {
51     return platNomor;
52 }
53
54 public void setPlatNomor(String platNomor) {
55     this.platNomor = platNomor;
56 }
57
58 public LocalDateTime getWaktuMasuk() {
59     return waktuMasuk;
60 }
61
62 public void setWaktuMasuk(LocalDateTime waktuMasuk) {
63     this.waktuMasuk = waktuMasuk;
64 }
65
66 public int getJenisKendaraan() {
67     return jenisKendaraan;
68 }
69
70 public void setJenisKendaraan(int jenisKendaraan) {
71     this.jenisKendaraan = jenisKendaraan;
72 }
```

4. ModelTabelParkiran.java

```
Source History
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5   package com.parkiran.model;
6
7   import java.util.List;
8   import javax.swing.table.AbstractTableModel;
9
10  /**
11   *
12   * @author felix
13   */
14
15  public class ModelTabelParkiran extends AbstractTableModel{
16      private List<ModelParkiran> parkiranList;
17      private String[] columnNames = {"Jenis Kendaraan", "Nomor Plat", "Waktu Masuk"};
18      public ModelTabelParkiran(List<ModelParkiran> parkiranList){
19          this.parkiranList = parkiranList;
20      }
21
22      @Override
23      public int getRowCount() {
24          return parkiranList.size();
25      }
26
27      @Override
28      public int getColumnCount() {
29          return columnNames.length;
30      }
31  }
```

```
Source History
32 @Override
33 public Object getValueAt(int rowIndex, int columnIndex) {
34     ModelParkiran parkiran = parkiranList.get(rowIndex);
35     switch (columnIndex) {
36         case 0:
37             int jenis = parkiran.getJenisKendaraan();
38             String jenis_kendaraan;
39             if (jenis == 1) {
40                 jenis_kendaraan = "Motor";
41             } else {
42                 jenis_kendaraan = "Mobil";
43             }
44             return jenis_kendaraan;
45         case 1:
46             return parkiran.getPlatNomor();
47         case 2:
48             return parkiran.getWaktuMasuk();
49         default:
50             return null;
51     }
52 }
53
54 @Override
55 public String getColumnName(int column) {
56     return columnNames[column];
57 }
58
59 @Override
60 public boolean isCellEditable(int rowIndex, int columnIndex) {
61     return false;
62 }
63
64 public void setParkiranList(List<ModelParkiran> parkiranList) {
65     this.parkiranList = parkiranList;
66     fireTableDataChanged();
67 }
68 }
```

5. ParkiranRepository.java

```
Source History
1  /**
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5   package com.parkiran.repository;
6
7   import com.parkiran.model.ModelParkiran;
8   import java.util.Optional;
9   import org.springframework.data.jpa.repository.JpaRepository;
10  import org.springframework.stereotype.Repository;
11
12  /**
13   *
14   * @author felix
15   */
16
17  @Repository
18  public interface ParkiranRepository extends JpaRepository<ModelParkiran, Long> {
19      public Optional<ModelParkiran> findByPlatNomor(String platNomor);
20  }
```

6. ParkiranService.java

```
Source History
1  /**
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5   package com.parkiran.service;
6
7   import com.parkiran.model.ModelParkiran;
8   import com.parkiran.repository.ParkiranRepository;
9   import java.time.Duration;
10  import java.time.LocalDateTime;
11  import java.util.List;
12  import java.util.Optional;
13  import org.springframework.beans.factory.annotation.Autowired;
14  import org.springframework.stereotype.Service;
15
16  /**
17   *
18   * @author felix
19   */
20
21  @Service
22  public class ParkiranService {
23
24      @Autowired
25      private ParkiranRepository repo;
26
27      public void masukParkir(ModelParkiran park) {
28          repo.save(park);
29      }
30
31      public String keluarParkir(String platNomor) {
32          Optional<ModelParkiran> optionalParkiran = repo.findByPlatNomor(platNomor);
33          if (optionalParkiran.isPresent()) {
34              ModelParkiran parkiran = optionalParkiran.get();
35              LocalDateTime waktuKeluar = LocalDateTime.now();
36
37              Duration durasi = Duration.between(parkiran.getWaktuMasuk(), waktuKeluar);
38              long jamParkir = durasi.toHours();
39              if (durasi.toMinutesPart() > 0) {
40                  jamParkir++;
41              }
42
43              int tarifPerJam = (parkiran.getJenisKendaraan() == 1) ? 5000 : 10000;
44              long totalBiaya = jamParkir * tarifPerJam;
45              String total = totalBiaya + "";
46
47              repo.delete(parkiran);
48
49              return total;
50          } else {
51              return "Kendaraan dengan plat nomor " + platNomor + " tidak ditemukan.";
52          }
53      }
54
55      public List<ModelParkiran> getAllParkiran() {
56          return repo.findAll();
57      }
58  }
59
```

7. ParkiranGUI.java

```
Source History
1 package com.parkiran.view;
2
3 import com.parkiran.controller.ParkiranController;
4 import com.parkiran.model.ModelParkiran;
5 import com.parkiran.model.ModelTabelParkiran;
6 import java.awt.*;
7 import java.time.LocalDateTime;
8 import java.util.ArrayList;
9 import javax.swing.*;
10 import java.util.List;
11 import javax.swing.table.JTableHeader;
12 import org.springframework.beans.factory.annotation.Autowired;
13 import org.springframework.stereotype.Component;
14
15 @Component
16 public class ParkiranGUI extends JFrame {
17
18     @Autowired
19     private ParkiranController parkiranController;
20
21     private JTextField platNomorField;
22     private JRadioButton motorRadioButton;
23     private JRadioButton mobilRadioButton;
24     private JButton masukButton;
25     private JButton keluarButton;
26     private JTextField platKeluarField;
27     private JTable parkiranTable;
28     private ModelTabelParkiran tableModel;
29
30     public ParkiranGUI(ParkiranController parkiranController) {
31         this.parkiranController = parkiranController;
32         // Set up JFrame
33         setTitle("Parkiran Management");
34         setSize(600, 400);
35         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
36         setLocationRelativeTo(null);
37
38         // Main layout
39         setLayout(new BorderLayout());
40
41         // Panel for form input
42         JPanel formPanel = new JPanel(new GridLayout(6, 2, 5, 5));
43         JLabel platNomorLabel = new JLabel("Plat Nomor:");
44         platNomorField = new JTextField();
45         formPanel.add(platNomorLabel);
46         formPanel.add(platNomorField);
47
48         JLabel jenisKendaraanLabel = new JLabel("Jenis Kendaraan:");
49         motorRadioButton = new JRadioButton("Motor");
50         mobilRadioButton = new JRadioButton("Mobil");
51
52         ButtonGroup kendaraanGroup = new ButtonGroup();
53         kendaraanGroup.add(motorRadioButton);
54         kendaraanGroup.add(mobilRadioButton);
55
56         formPanel.add(jenisKendaraanLabel);
57         JPanel radioPanel = new JPanel(new FlowLayout(FlowLayout.LEFT));
58         radioPanel.add(motorRadioButton);
59         radioPanel.add(mobilRadioButton);
60         formPanel.add(radioPanel);
61
62         masukButton = new JButton("Masuk");
63         keluarButton = new JButton("Keluar");
64
65         JLabel platKeluarLabel = new JLabel("Plat Nomor Keluar:");
66         platKeluarField = new JTextField();
67         formPanel.add(platKeluarLabel);
68         formPanel.add(platKeluarField);
69         formPanel.add(masukButton);
70         formPanel.add(keluarButton);
71
72         add(formPanel, BorderLayout.NORTH);
73
74     }
```

```
Source History
73
74 // Table for displaying parkir data
75 tableModel = new ModelTabelParkiran(new ArrayList<>());
76 parkirTable = new JTable(tableModel);
77 JScrollPane scrollPane = new JScrollPane(parkirTable);
78
79 // Set header style
80 JTableHeader header = parkirTable.getTableHeader();
81 header.setFont(new Font("SansSerif", Font.BOLD, 14));
82
83 add(scrollPane, BorderLayout.CENTER);
84
85 // Add action listeners
86 masukButton.addActionListener(e -> handleMasukParkir(kendaraanGroup));
87 keluarButton.addActionListener(e -> handleKeluarParkir());
88 updateTableData();
89
90
91 private void handleMasukParkir(ButtonGroup kendaraanGroup) {
92     String platNomor = platNomorField.getText();
93
94     int jenisKendaraan;
95     if (motorRadioButton.isSelected()) {
96         jenisKendaraan = 1; // Motor
97     } else if (mobilRadioButton.isSelected()) {
98         jenisKendaraan = 2; // Mobil
99     } else {
100         JOptionPane.showMessageDialog(this, "Pilih jenis kendaraan!", "Error", JOptionPane.ERROR_MESSAGE);
101         return;
102     }
103
104     ModelParkiran parkir = new ModelParkiran(platNomor, LocalDateTime.now(), jenisKendaraan);
105
106     // Call controller method
107     parkirController.masukParkir(parkir);
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
```

```
Source History
109 JOptionPane.showMessageDialog(this, "Kendaraan berhasil masuk parkir!", "Success", JOptionPane.INFORMATION_MESSAGE);
110
111 // Refresh table data
112 updateTableData();
113
114 // Clear input fields
115 platNomorField.setText("");
116 kendaraanGroup.clearSelection();
117
118
119 private void handleKeluarParkir() {
120     String platNomorKeluar = platKeluarField.getText();
121
122     // Call controller method and get the result
123     String result = parkirController.keluarParkir(platNomorKeluar);
124
125     if (result.startsWith("Kendaraan")) {
126         JOptionPane.showMessageDialog(this, result, "Error", JOptionPane.ERROR_MESSAGE);
127     } else {
128         JOptionPane.showMessageDialog(this, "Biaya Parkir: Rp " + result, "Success", JOptionPane.INFORMATION_MESSAGE);
129     }
130
131     // Refresh table data
132     updateTableData();
133
134     // Clear input field
135     platKeluarField.setText("");
136
137
138 private void updateTableData() {
139     // Call controller to fetch updated data
140     List<ModelParkiran> updatedData = parkirController.getAllParkiran();
141     tableModel.setParkiranList(updatedData);
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

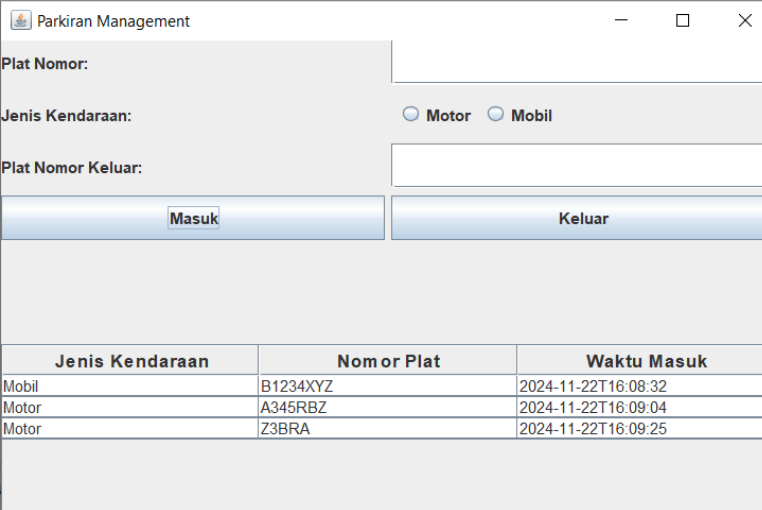
```

126         JOptionPane.showMessageDialog(this, result, "Error", JOptionPane.ERROR_MESSAGE);
127     } else {
128         JOptionPane.showMessageDialog(this, "Biaya Parkir: Rp " + result, "Success", JOptionPane.INFORMATION_MESSAGE);
129     }
130
131     // Refresh table data
132     updateTableData();
133
134     // Clear input field
135     platKeluarField.setText("");
136 }
137
138 private void updateTableData() {
139     // Call controller to fetch updated data
140     List<ModelParkiran> updatedData = parkiranController.getAllParkiran();
141     tableModel.setParkiranList(updatedData);
142 }
143
144 public static void main(String[] args) {
145     // Assume ParkiranController is already properly configured via Spring
146     ParkiranController controller = new ParkiranController(); // Placeholder
147     ParkiranGUI gui = new ParkiranGUI(controller);
148     gui.setVisible(true);
149 }
150 }

```

8. Output Program

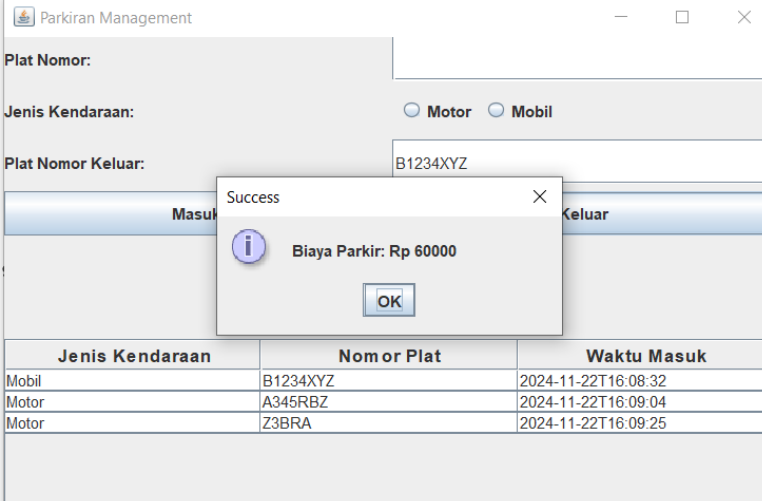
a. Masuk Parkir



The screenshot shows the 'Parkiran Management' application window. It has a title bar with a close button. The window contains several input fields and buttons. The 'Plat Nomor' field is empty. The 'Jenis Kendaraan' field has two radio buttons: 'Motor' and 'Mobil'. The 'Plat Nomor Keluar' field is empty. Below these fields are two buttons: 'Masuk' and 'Keluar'. At the bottom of the window is a table with three columns: 'Jenis Kendaraan', 'Nomor Plat', and 'Waktu Masuk'.

Jenis Kendaraan	Nomor Plat	Waktu Masuk
Mobil	B1234XYZ	2024-11-22T16:08:32
Motor	A345RBZ	2024-11-22T16:09:04
Motor	Z3BRA	2024-11-22T16:09:25

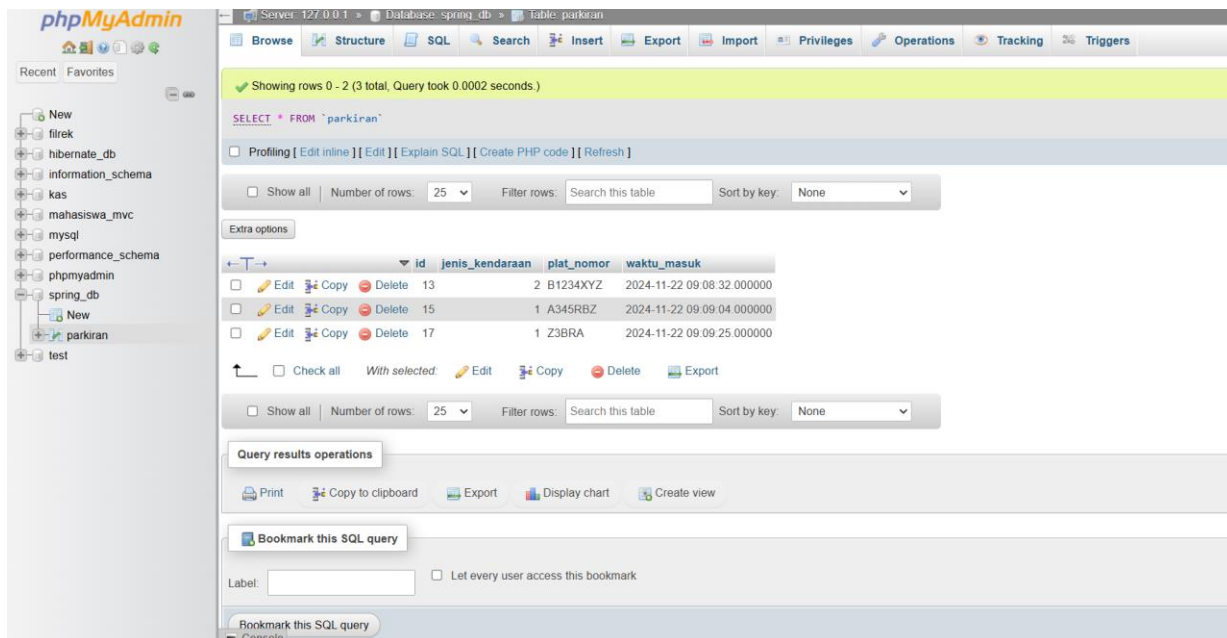
b. Keluar Parkir



The screenshot shows the 'Parkiran Management' application window. The 'Plat Nomor' field is empty. The 'Jenis Kendaraan' field has two radio buttons: 'Motor' and 'Mobil'. The 'Plat Nomor Keluar' field contains the text 'B1234XYZ'. Below these fields are two buttons: 'Masuk' and 'Keluar'. A success message dialog box is displayed in the foreground, with the title 'Success' and the message 'Biaya Parkir: Rp 60000'. The dialog box has an 'OK' button. At the bottom of the window is a table with three columns: 'Jenis Kendaraan', 'Nomor Plat', and 'Waktu Masuk'.

Jenis Kendaraan	Nomor Plat	Waktu Masuk
Mobil	B1234XYZ	2024-11-22T16:08:32
Motor	A345RBZ	2024-11-22T16:09:04
Motor	Z3BRA	2024-11-22T16:09:25

9. Phpmyadmin



The screenshot displays the phpMyAdmin web interface. On the left, a sidebar shows a tree view of databases and tables, with 'spring_db' and its table 'parkiran' selected. The main panel shows the 'Structure' tab for the 'parkiran' table. A yellow status bar at the top indicates 'Showing rows 0 - 2 (3 total, Query took 0.0002 seconds.)'. Below this, the SQL query 'SELECT * FROM `parkiran`' is shown. A toolbar contains links for 'Profiling', 'Edit inline', 'Edit', 'Explain SQL', 'Create PHP code', and 'Refresh'. A control bar allows filtering rows and sorting by key. The table data is displayed in a grid with columns: id, jenis_kendaraan, plat_nomor, and waktu_masuk. Each row has interactive icons for edit, copy, and delete. Below the table, there are options to 'Check all', 'With selected', 'Edit', 'Copy', 'Delete', and 'Export'. A 'Query results operations' section includes 'Print', 'Copy to clipboard', 'Export', 'Display chart', and 'Create view'. At the bottom, there is a 'Bookmark this SQL query' section with a label input and a checkbox 'Let every user access this bookmark'.

id	jenis_kendaraan	plat_nomor	waktu_masuk
13	2	B1234XYZ	2024-11-22 09:08:32.000000
15	1	A345RBZ	2024-11-22 09:09:04.000000
17	1	Z3BRA	2024-11-22 09:09:25.000000