

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Selama proses magang, tugas dan proyek yang diberikan dilaksanakan dengan mencakup pemahaman terhadap sistem yang digunakan, pengembangan fitur, serta pelaksanaan pengujian dan *debugging*. Dalam menjalankan tugas, koordinasi dilakukan secara berkala dengan *team lead* selaku *supervisor*, yang bertanggung jawab dalam memberikan arahan dan bimbingan teknis. Selain itu, setiap aktivitas yang dilakukan terdokumentasi dengan baik sebagai bagian dari proses evaluasi. Selanjutnya dilakukan evaluasi berkala, di mana *team lead* memberikan saran dan masukkan terhadap pekerjaan yang telah diselesaikan. Evaluasi ini bertujuan untuk menilai perkembangan kerja serta memberikan rekomendasi perbaikan untuk meningkatkan kualitas hasil yang dicapai. Pada tahap akhir program magang, laporan dan dokumentasi teknis disusun, berisi ringkasan pengalaman serta kontribusi yang telah diberikan dalam proyek yang dikerjakan.

3.2 Tugas yang Dilakukan

Kontribusi yang diberikan berfokus pada pengembangan aplikasi UFINS, sebuah aplikasi *finance* yang dirancang untuk mendukung berbagai aspek operasional perusahaan melalui sejumlah modul. Beberapa modul utama dalam UFINS meliputi:

- Management Setup
- General Ledger
- Budgeting System
- Procurement System
- Account Payable System
- Tax
- Fixed Asset
- Pre Paid

- Inventory
- Vendor Management

Dalam proyek UFINS, keterlibatan difokuskan pada modul *Procurement System*. Kontribusi yang diberikan mencakup pengembangan dan penyempurnaan fitur untuk mendukung proses pengadaan (*procurement*). *Procurement System* adalah modul yang mengelola proses pengadaan barang dan jasa secara transparan dan efisien. Proses dimulai dari registrasi *vendor*, pertemuan/penjelasan teknis proses lelang (*aanwijzing*), hingga *upload quotation*. *Vendor* juga mengikuti klarifikasi dan *beauty contest* untuk memastikan kesesuaian penawaran dengan kebutuhan. Tahapan selanjutnya meliputi *negosiasi*, *awarding* (penentuan pemenang), revisi *quotation*, konfirmasi pengguna, dan *penyanggahan* bagi *vendor* yang tidak terpilih. Proses ini dirancang untuk menjamin keterbukaan dan akuntabilitas dalam pengadaan. Kontribusi yang diberikan meliputi pengembangan fitur *audit trail* untuk merekam seluruh aktivitas dalam sistem. Fitur ini memastikan transparansi dan integritas data, mencakup pencatatan *log*, integrasi dengan sistem, serta pengujian untuk menampilkan data secara akurat sesuai kebutuhan pengguna.

Pelaksanaan kerja magang diuraikan seperti pada Tabel 3.1.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu Ke -	Pekerjaan yang dilakukan
1	Penyesuaian dan perkenalan tech stack yang digunakan di tempat magang.
2	Memulai penggeraan service Audit Trail dengan fokus pada API Audit Trail Transaction Flow.
3	Implementasi fitur reporting untuk API Audit Trail Transaction Flow yang melibatkan Excel builder, query builder, dan repository untuk Audit Trail Transaction Flow, untuk memfasilitasi query atau akses ke database, serta membangun struktur tabel dan kolom dalam file Excel.
4	Menyempurnakan struktur file Excel dengan membangun tabel dan kolom secara dinamis berdasarkan data dari Audit Trail Transaction Flow.
5	Melakukan debugging menyeluruh terhadap seluruh implementasi API Audit Trail Transaction Flow, mulai dari controller, DAO, hingga repository dan class lain yang terkait.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang (lanjutan)

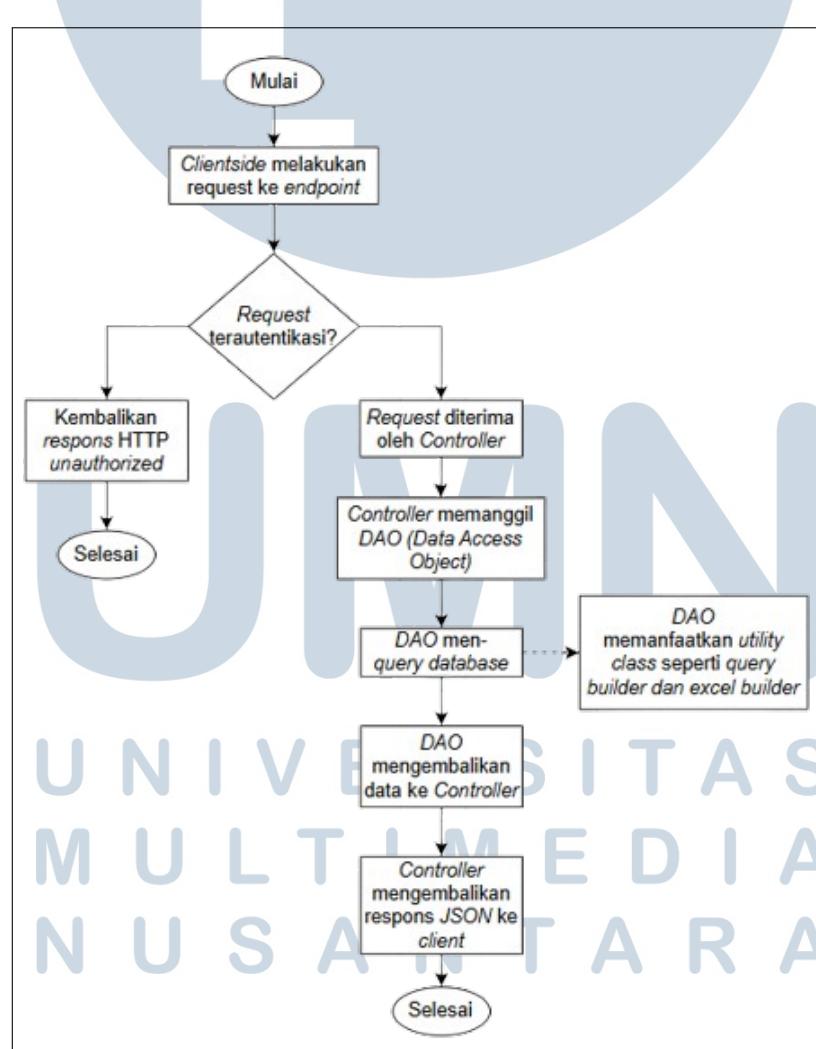
Minggu Ke -	Pekerjaan yang dilakukan
6	Memulai penggerjaan API Audit Trail Step Workflow dalam service Audit Trail.
7	Mengimplementasikan fitur reporting yang mencakup pembuatan Excel builder, query builder, serta repository untuk mengakses database dan membangun struktur tabel serta kolom pada file Excel.
8	Membuat dan menyempurnakan template laporan PDF menggunakan Jaspersoft Studio untuk diimplementasikan ke dalam code backend.
9	Melakukan debugging dan testing menyeluruh terhadap seluruh implementasi API Audit Trail Step Workflow, dan revisi API agar dapat menampilkan field workflowName yang berasal dari master_bidding_workflow_h melalui join tabel dan pengeditan pada query builder dan Excel builder.
10	Memulai penggerjaan API Audit Trail Contract dalam service Audit Trail.
11	Mengimplementasikan fitur reporting dengan menggunakan Excel builder, query builder, dan repository untuk mendukung akses database serta membentuk struktur tabel dan kolom pada file Excel.
12	Membuat dan menyempurnakan template laporan PDF menggunakan Jaspersoft Studio untuk diimplementasikan ke dalam code backend.
13	Melakukan debugging dan testing menyeluruh terhadap seluruh implementasi API Audit Trail Contract.
14	Memulai penggerjaan API Audit Parameter.
15	Mengimplementasikan fitur reporting dengan menggunakan Excel builder, query builder, dan repository untuk mendukung akses database serta membentuk struktur tabel dan kolom pada file Excel.
16	Melakukan debugging dan testing menyeluruh terhadap seluruh implementasi API Audit Parameter.

3.3 Uraian Pelaksanaan Magang

Dalam sistem UFINS, *audit trail* merupakan fitur penting yang berfungsi untuk merekam seluruh aktivitas atau perubahan data yang dilakukan oleh pengguna di berbagai modul sistem. Fitur ini berperan dalam menjaga transparansi, akuntabilitas, serta memudahkan proses pelacakan apabila terjadi kesalahan atau

penyalahgunaan. Pencatatan dilakukan secara otomatis setiap kali terjadi interaksi dengan sistem, baik melalui proses input, perubahan, maupun penghapusan data. Modul-modul yang tercakup dalam *audit trail* mencakup berbagai aspek utama dalam UFINS, seperti pengelolaan anggaran, kontrak, aset, hingga proses persetujuan dan login pengguna. Data hasil pencatatan ini kemudian dapat disajikan dalam bentuk laporan menggunakan *JasperReports*, sehingga mempermudah proses audit internal maupun eksternal. Dengan demikian, *audit trail* menjadi salah satu komponen krusial dalam mendukung keamanan dan integritas data dalam aplikasi.

Untuk memahami proses kerja *audit trail* dalam modul UFINS, dibuat sebuah *flowchart* yang menggambarkan alur komunikasi antara *client* dan *server* saat permintaan data audit dilakukan.



Gambar 3.1. Bagan Alur Proses Komunikasi API Audit Trail

Alur dimulai saat *client* mengirimkan *request* ke *endpoint* tertentu. Permintaan ini diperiksa untuk memastikan bahwa pengguna telah terautentikasi melalui validasi *token* JWT. Jika token tidak valid atau pengguna tidak memiliki hak akses, sistem akan langsung mengembalikan respons *Unauthorized* dan proses berhenti. Namun jika autentikasi berhasil, *controller* akan meneruskan permintaan ke lapisan *DAO* (Data Access Object), yang bertugas membangun kriteria pencarian secara dinamis dan mengambil data dari *database* sesuai dengan parameter yang diberikan. Data hasil pencarian dikembalikan ke *controller* dan kemudian diformat ke dalam bentuk yang sesuai, seperti *JSON*, PDF, atau Excel, tergantung pada jenis *endpoint* yang dipanggil. Hasil akhir ini dikirim kembali ke *client* sebagai respons.

3.3.1 Audit Trail Transaction Flow

Fokus pekerjaan selama magang diarahkan pada pengembangan fitur *audit trail transaction flow* dalam modul Procurement System UFINS. Tiga *method* baru ditambahkan ke dalam *AudittrailController*, yaitu *viewAudittrailTF*, *downloadAudittrailTF*, dan *downloadAudittrailTFExcel*. Ketiganya bertanggung jawab untuk menampilkan serta mengunduh data audit berdasarkan permintaan pengguna.

- *viewAudittrailTF*:

Pada Kode 3.1 terdapat metode *viewAudittrailTF*. Metode ini menangani permintaan POST yang ditujukan untuk melihat data audit trail transaction flow. Proses dimulai dengan validasi *JWT token* dari *header* permintaan guna memastikan autentikasi. Setelah itu, dilakukan *query* ke database melalui objek *auditDAO* berdasarkan kriteria yang diberikan. Bila data ditemukan, maka akan dikembalikan bersama total jumlah entri. Bila tidak, respons yang menunjukkan data kosong dikembalikan.

```
1  @SuppressWarnings("unchecked")
2  @RequestMapping(value = AudittrailConstant.
3      PROCUREMENT_AUDITTRAIL_FLOW_TRANSACTION_VIEW, method =
4          RequestMethod.POST)
5  @CrossOrigin(origins = "*")
6  @Logging
7  @Authenticate
8  public String viewAudittrailTF (HttpServletRequest request,
9      HttpServletResponse response, @RequestBody View View)
10 throws Exception {
```

```

7   try
8   {
9     String tokens = request.getHeader(
10    AuthenticationConstant.AUTHORIZATION);
11    String nezam = request.getHeader(AuthenticationConstant
12    .NEZNAM);
13    authRes = AuthenticationApi.authentication(
14    authentication , tokens);
15    if (authRes == null) {
16      return senderAuditTransactionFlow .
17      responseNotAuthorized(request , response);
18    }
19    if (!nezam.equalsIgnoreCase(authRes .
20    getAuthenticatedMethod())) {
21      return senderAuditTransactionFlow .
22      responseNotAuthorized(request , response);
23    }
24    Map<String , Object> data = auditDAO .
25    getDataAudittrailTransactionFlowByCriteria(View .getSearch
26    () , View .getStatus() , View .getStartDate() ,View .getEndDate
27    () , View .getBranchid() , authRes .getBranchId());
28    List<AudittrailTransactionFlow> listAuditTF = (List<
29    AudittrailTransactionFlow >) data .get("data");
30    long total = (long) data .get("total");
31    if (total == 0) {
32      return senderAuditTransactionFlow .responseEmpty(
33      request , response);
34    }
35    return senderAuditTransactionFlow .responseListCount(
36      request , response , listAuditTF , total);
37  }
38  catch (Exception e) {
39    e.printStackTrace();
40    return senderAuditTransactionFlow .
41    sendErrorInternalServerError(request , response , e , logger);
42  }
43}

```

N U S A N D I A R A Kode 3.1: Metode viewAudittrailTF

- downloadAudittrailTF:

Pada Kode 3.2 terdapat metode downloadAudittrailTF. Metode ini

digunakan untuk menghasilkan dan mengunduh laporan *Audit Trail Transaction Flow* dalam format PDF, berdasarkan parameter pencarian seperti `search`, `status`, `startDate`, `endDate`, dan `branchId`. Setelah menerima permintaan, sistem melakukan *autentikasi* menggunakan token dari header request melalui `AuthenticationApi` untuk memperoleh `userId` dan `branchId`. Selanjutnya, tanggal proses diambil dari database dan diformat, kemudian data audit yang sesuai dicari dengan method `getDataAudittrailTransactionFlowByCriteria()`. Data ini diubah menjadi objek `AudittrailTransactionFlow`, lalu dikombinasikan dengan parameter laporan seperti. Template laporan `.jrxml` dikompilasi dan diisi menggunakan *JasperReports*, menghasilkan objek `JasperPrint` yang kemudian dikonversi menjadi file PDF.

```
1  @SuppressWarnings({ "unused", "unchecked" })
2  @RequestMapping(value = AudittrailConstant.
3      PROCUREMENT_AUDITTRAIL_FLOW_TRANSACTION_DOWNLOAD.PDF,
4      method = RequestMethod.POST)
5  @CrossOrigin(origins = "*")
6  @Logging
7  @Authenticate
8  public void downloadAudittrailTF (HttpServletRequest
9      request , HttpServletResponse response , @RequestBody View
10     View) throws Exception {
11      String tokens = request.getHeader(AuthenticationConstant.
12          AUTHORIZATION);
13      authRes = AuthenticationApi.authentication(authentication
14          , tokens);
15      SimpleDateFormat dateFormat3 = new SimpleDateFormat("yyyy
16          -MM-dd HH:mm:ss");
17      ProcessDate pros = reportdao.getProcessDate();
18      Date prosdate = SetProcessDate.getprosdate(pros);
19      String datetime = dateFormat3.format(prosdate);
20      String userid = authRes.getUserId();
21      String startDate = View.getStartDate();
22      String endDate = View.getEndDate();
23      List<JasperPrint> jasperPrints = new ArrayList<>();
24      Map<String , Object> data = auditDAO.
25          getDataAudittrailTransactionFlowByCriteria(View.getSearch
26              () , View.getStatus() , View.getStartDate() , View.getEndDate
27              () , View.getBranchid() , authRes.getBranchId());
28      List<AudittrailTransactionFlow> listAuditTF = (List<
29          AudittrailTransactionFlow>) data.get("data");
```

```

19
20     Map<String , Object> parameters = new HashMap<String , Object>();
21     parameters .put("P_USER" , userid);
22     parameters .put("P_PROCESSDATE" , datetime);
23     parameters .put("P_STARTDATE" , startDate);
24     parameters .put("P_ENDDATE" , endDate);
25     String fileName = templateLocation + "
26     Audittrail_Transaction_Flow.jrxml";
27     JasperReport jp = JasperCompileManager.compileReport(
28         fileName);
29     JasperPrint print = JasperFillManager.fillReport(jp ,
30         parameters , new JRBeanCollectionDataSource(listAuditTF));
31
32     jasperPrints.add(print);
33
34
35     PDF pdf = new PDF(response , "REPORT_AUDITTRAIL_TF_" ,".pdf"
36     ,print);
37 }

```

Kode 3.2: Metode downloadAudittrailTF

- downloadAudittrailTFExcel:

Pada Kode 3.3 terdapat metode downloadAudittrailTFExcel. Metode ini menangani pengunduhan data dalam format Excel. Alur prosesnya mirip dengan PDF, namun menggunakan *generator* Excel untuk membuat file dan mengembalikannya dalam bentuk InputStreamResource.

```

1     @SuppressWarnings({ "unused" , "unchecked" })
2     @RequestMapping(value = AudittrailConstant .
3         PROCUREMENT_AUDITTRAIL_FLOW_TRANSACTION_DOWNLOAD_EXCEL,
4         method = RequestMethod .POST)
5     @CrossOrigin(origins = "*")
6     @Logging
7     @Authenticate
8     public ResponseEntity<InputStreamResource>
9         downloadAudittrailTFExcel(HttpServletRequest request ,
10         HttpServletResponse response , @RequestBody View View)
11         throws Exception {
12         String tokens = request.getHeader(AuthenticationConstant .
13             AUTHORIZATION);
14         authRes = AuthenticationApi.authentication(authentication
15             , tokens);
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
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
94
95
96
97
98
99
100
101
102
103
104
105
106
107
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
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
259
260
261
262
263
264
265
266
267
268
269
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
287
288
289
289
290
291
292
293
294
295
296
297
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
311
312
313
313
314
314
315
315
316
316
317
317
318
318
319
319
320
320
321
321
322
322
323
323
324
324
325
325
326
326
327
327
328
328
329
329
330
330
331
331
332
332
333
333
334
334
335
335
336
336
337
337
338
338
339
339
340
340
341
341
342
342
343
343
344
344
345
345
346
346
347
347
348
348
349
349
350
350
351
351
352
352
353
353
354
354
355
355
356
356
357
357
358
358
359
359
360
360
361
361
362
362
363
363
364
364
365
365
366
366
367
367
368
368
369
369
370
370
371
371
372
372
373
373
374
374
375
375
376
376
377
377
378
378
379
379
380
380
381
381
382
382
383
383
384
384
385
385
386
386
387
387
388
388
389
389
390
390
391
391
392
392
393
393
394
394
395
395
396
396
397
397
398
398
399
399
400
400
401
401
402
402
403
403
404
404
405
405
406
406
407
407
408
408
409
409
410
410
411
411
412
412
413
413
414
414
415
415
416
416
417
417
418
418
419
419
420
420
421
421
422
422
423
423
424
424
425
425
426
426
427
427
428
428
429
429
430
430
431
431
432
432
433
433
434
434
435
435
436
436
437
437
438
438
439
439
440
440
441
441
442
442
443
443
444
444
445
445
446
446
447
447
448
448
449
449
450
450
451
451
452
452
453
453
454
454
455
455
456
456
457
457
458
458
459
459
460
460
461
461
462
462
463
463
464
464
465
465
466
466
467
467
468
468
469
469
470
470
471
471
472
472
473
473
474
474
475
475
476
476
477
477
478
478
479
479
480
480
481
481
482
482
483
483
484
484
485
485
486
486
487
487
488
488
489
489
490
490
491
491
492
492
493
493
494
494
495
495
496
496
497
497
498
498
499
499
500
500
501
501
502
502
503
503
504
504
505
505
506
506
507
507
508
508
509
509
510
510
511
511
512
512
513
513
514
514
515
515
516
516
517
517
518
518
519
519
520
520
521
521
522
522
523
523
524
524
525
525
526
526
527
527
528
528
529
529
530
530
531
531
532
532
533
533
534
534
535
535
536
536
537
537
538
538
539
539
540
540
541
541
542
542
543
543
544
544
545
545
546
546
547
547
548
548
549
549
550
550
551
551
552
552
553
553
554
554
555
555
556
556
557
557
558
558
559
559
560
560
561
561
562
562
563
563
564
564
565
565
566
566
567
567
568
568
569
569
570
570
571
571
572
572
573
573
574
574
575
575
576
576
577
577
578
578
579
579
580
580
581
581
582
582
583
583
584
584
585
585
586
586
587
587
588
588
589
589
590
590
591
591
592
592
593
593
594
594
595
595
596
596
597
597
598
598
599
599
600
600
601
601
602
602
603
603
604
604
605
605
606
606
607
607
608
608
609
609
610
610
611
611
612
612
613
613
614
614
615
615
616
616
617
617
618
618
619
619
620
620
621
621
622
622
623
623
624
624
625
625
626
626
627
627
628
628
629
629
630
630
631
631
632
632
633
633
634
634
635
635
636
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394

```

```
9 Map<String , Object> data = auditDAO.  
10    getDataAudittrailTransactionFlowByCriteria(View.getSearch()  
11    () , View.getStatus() , View.getStartDate() , View.getEndDate()  
12    () , View.getBranchid() , authRes.getBranchId()));  
13    List<AudittrailTransactionFlow> listAuditTF = (List<  
14    AudittrailTransactionFlow >) data .get("data");  
15    String filename = "REPORT_AUDITTRAIL_TF_";  
16    Date now = new Date();  
17    String dateMerge = new SimpleDateFormat(ParameterConstant  
18    .DATETIME_YMDHMSSSS_MERGE) .format(now);  
19    String fileExportName = filename + dateMerge + ".xlsx";  
20    logger.info("Start Print EXCEL Data Audittrail TF " + new  
21    Date() .getTime());  
22    InputStreamResource file = new InputStreamResource(  
23    auditDAO.loadListDataAudittrailTransactionFlowDownload(  
24    listAuditTF));  
25    logger.info("End Print EXCEL Data Audittrail TF " + new  
26    Date() .getTime());  
27    return ResponseEntity.ok().header(HttpHeaders.  
28    CONTENT_DISPOSITION , "attachment; filename=" +  
29    fileExportName).contentType(MediaType.parseMediaType("br/>30    application/vnd-ms.excel")).body(file);  
31  }  
32 }
```

Kode 3.3: Metode downloadAudittrailTFEExcel

Kemudian, dikembangkan *method* baru dalam class AudittrailDAOImpl untuk mengambil dan menghitung data berdasarkan kriteria pencarian tertentu yaitu `getDataAudittrailTransactionFlowByCriteria` dan `loadListDataAudittrailTransactionFlowDownload`. Potongan kode dapat dilihat pada Kode 3.4

- `getDataAudittrailTransactionFlowByCriteria:`
Method ini digunakan untuk mengambil data berdasarkan beberapa kriteria seperti `search`, `status`, `startDate`, `endDate`, `branch`, dan `branchAuth`. Hasil dari *method* ini berupa `Map` yang berisi total data dan daftar hasil pencarian.
 - `loadListDataAudittrailTransactionFlowDownload:`
Method ini mengubah daftar objek `AudittrailTransactionFlow` menjadi format Excel yang siap diunduh sebagai `InputStream`.

```

1  @Override
2  public Map<String , Object>
3      getDataAudittrailTransactionFlowByCriteria( String search ,
4          String status , String startDate , String endDate , List<
5              String > branch , String branchAuth) throws AppException {
6          // TODO Auto-generated method stub
7          Map<String , Object> retValue = new HashMap<String , Object
8              >();
9          Map<String , Object> criteria = new HashMap<String , Object
10             >();
11         List<AudittrailTransactionFlow> list = new ArrayList<
12             AudittrailTransactionFlow >();
13         auditTransactionFlowRepository .findAll(
14             builderAuditTransactionFlow .
15                 queryBuilderAudittrailTransactionFlow( criteria ,
16                     search ,
17                     status ,
18                     startDate ,
19                     endDate ,
20                     branch ,
21                     branchAuth)) .forEach( audittrailId -> list .add(
22                         audittrailId));
23         long t = auditTransactionFlowRepository .count(
24             builderAuditTransactionFlow .
25                 queryBuilderAudittrailTransactionFlow( criteria ,
26                     search ,
27                     status ,
28                     startDate ,
29                     endDate ,
30                     branch ,
31                     branchAuth));
32         retValue .put("total" , t );
33         retValue .put("data" , list );
34         return retValue ;
35     }
36
37     @Override
38     public InputStream
39         loadListDataAudittrailTransactionFlowDownload( List<
40             AudittrailTransactionFlow > listData) throws AppException {
41         // TODO Auto-generated method stub
42         ByteArrayInputStream in =
43             ExcelBuilderAudittrailTransactionFlow .

```

```

30     ParameterAudittrailTransactionFlowToExcel( listData );
31     return in;
32 }

```

Kode 3.4: Metode getDataAudittrailTransactionFlowByCriteria

Kemudian, diimplementasikan *class* baru bernama *ExcelBuilderAudittrailTransactionFlow*, *method* *queryBuilderAudittrailTransactionFlow* dalam *class* *QueryBuilderAudittrail*, dan membuat *class* *AudittrailTransactionFlowRepository*.

- *ExcelBuilderAudittrailTransactionFlow*:
Class ini berfungsi sebagai utilitas untuk membangun file Excel dari data transaksi audit. Disediakan *method* *ParameterAudittrailTransactionFlowToExcel* untuk mengatur isi dan format file, serta *hasExcelFormat* untuk validasi. Hasil atau *screenshot* dari kode dapat dilihat pada Lampiran 7.
- *queryBuilderAudittrailTransactionFlow*:
Method *queryBuilderAudittrailTransactionFlow* membangun Specification dinamis dengan menggabungkan beberapa Predicate berdasarkan *search*, *status*, *transactionDate*, serta *branchAuth*. *Field* seperti *title*, *userid*, dan *rfqNo* difilter menggunakan *cb.like*, sementara filter *status*, cabang, dan tanggal menggunakan *cb.equal* atau *cb.between*. Hasil pencarian diurutkan berdasarkan *transactionDate* secara menurun. Hasil atau *screenshot* dari kode dapat dilihat pada Lampiran 8.

3.3.2 Audit Trail Step Workflow

Pengembangan *API audit trail step workflow* pada modul *Procurement System UFINS* dimulai dengan menambahkan tiga *method* ke dalam *AudittrailController*, yaitu *viewAudittrailsW*, *downloadAudittrailsW*, dan *downloadAudittrailsWExcel*.

- *viewAudittrailsW*:

Pada Kode 3.5 terdapat metode *viewAudittrailsW*. Metode ini menangani

permintaan POST untuk menampilkan data *Audit Trail Step Workflow* dengan validasi token JWT sebagai autentikasi pengguna. Data diambil melalui metode `getDataAudittrailStepWorkflowByCriteria()` dengan parameter `search`, `status`, `startDate`, `endDate`, dan `branchId`. Data tersebut kemudian dikonversi menjadi daftar `AudittrailStepWorkflow`, dihitung totalnya, dan dikembalikan dalam format *list with count*. Jika tidak ditemukan data yang sesuai, sistem akan mengembalikan respons kosong.

```
1  @SuppressWarnings("unchecked")
2  @RequestMapping(value = AudittrailConstant.
3      PROCUREMENT_AUDITTRAIL_STEP_WORKFLOW_VIEW, method =
4      RequestMethod.POST)
5  @CrossOrigin(origins = "*")
6  @Logging
7  @Authenticate
8  public String viewAudittrailSW(HttpServletRequest request
9      , HttpServletResponse response, @RequestBody View View)
10     throws Exception {
11     try {
12         String tokens = request.getHeader(
13             AuthenticationConstant.AUTHORIZATION);
14         String nezam = request.getHeader(
15             AuthenticationConstant.NEZNAM);
16         authRes = AuthenticationApi.authentication(
17             authentication, tokens);
18         if (authRes == null || !nezam.equalsIgnoreCase(
19             authRes.getAuthenticatedMethod())) {
20             return senderAuditStepWorkflow.
21                 responseNotAuthorized(request, response);
22         }
23
24         Map<String, Object> data = auditDAO.
25             getDataAudittrailStepWorkflowByCriteria(
26                 View.getSearch(), View.getStatus(), View.
27                 getStartDate(),
28                     View.getEndDate(), View.getBranchid(),
29                     authRes.getBranchId());
30
31         List<AudittrailStepWorkflow> listAuditSW = (List<
32             AudittrailStepWorkflow>) data.get("data");
33         long total = (long) data.get("total");
34
35         if (total == 0) {
```

```

23             return senderAuditStepWorkflow.responseEmpty(
24     request, response);
25 }
26
27         return senderAuditStepWorkflow.responseListCount(
28     request, response, listAuditSW, total);
29     } catch (Exception e) {
30         e.printStackTrace();
31         return senderAuditStepWorkflow.
32             sendErrorInternalServer(request, response, e, logger);
33     }
34 }
```

Kode 3.5: Metode viewAudittrailSW

- downloadAudittrailSW:

Pada Kode 3.6 terdapat metode downloadAudittrailSW. Metode ini menghasilkan file PDF dari data *Audit Trail Step Workflow* berdasarkan parameter pencarian search, status, startDate, endDate, dan branchId. Setelah proses autentikasi pengguna, sistem mengambil tanggal proses terkini melalui utilitas getProcessDate() dan memformatnya menjadi processDate. Informasi ini, bersama userId, dimasukkan ke dalam parameter laporan. Data hasil pencarian dikonversi menjadi daftar AudittrailStepWorkflow, lalu dikaitkan dengan template JasperReports berformat .jrxml. Template ini dikompilasi dan diisi menggunakan JasperCompileManager serta JasperFillManager sehingga menghasilkan objek JasperPrint. File PDF kemudian dihasilkan dari objek ini dan dikirim ke *client* menggunakan utilitas PDF.

```

1 @SuppressWarnings({ "unused", "unchecked" })
2 @RequestMapping(value = AudittrailConstant.
3 PROCUREMENT_AUDITTRAIL_STEP_WORKFLOW_DOWNLOAD_PDF, method
4 = RequestMethod.POST)
5 @CrossOrigin(origins = "*")
6 @Logging
7 @Authenticate
8 public void downloadAudittrailSW(HttpServletRequest
request, HttpServletResponse response, @RequestBody View
View) throws Exception {
9     SimpleDateFormat dateFormat3 = new SimpleDateFormat(""
10 yyyy-MM-dd HH:mm:ss");
```

```

8     ProcessDate pros = reportdao.getProcessDate();
9     Date prosdate = SetProcessDate.getprosdate(pros);
10    String datetime = dateFormat3.format(prosdate);
11    String userid = authRes.getUserId();

12
13    Map<String , Object> data = auditDAO.
14      getDataAudittrailStepWorkflowByCriteria(
15        View.getSearch() , View.getStatus() , View.
16        getStartDate() ,
17        View.getEndDate() , View.getBranchid() , authRes.
18        getBranchId());

19    List<AudittrailStepWorkflow> listAuditSW = (List<
20      AudittrailStepWorkflow>) data.get("data");

21    Map<String , Object> parameters = new HashMap<>();
22    parameters.put("P_USER" , userid);
23    parameters.put("P_PROCESSDATE" , datetime);
24    parameters.put("P_STARTDATE" , View.getStartDate());
25    parameters.put("P_ENDDATE" , View.getEndDate());

26    String fileName = templateLocation + "
27 Audittrail_Step_Workflow.jrxml";
28    JasperReport jp = JasperCompileManager.compileReport(
29      fileName);
30    JasperPrint print = JasperFillManager.fillReport(jp ,
31      parameters , new JRBeanCollectionDataSource(listAuditSW));

32
33    PDF pdf = new PDF(response , "REPORT_AUDITTRAIL_SW_",
34      ".pdf" , print);
35  }

```

Kode 3.6: Metode downloadAudittrailSW

- downloadAudittrailsWExcel:

Pada Kode 3.7 terdapat metode downloadAudittrailsWExcel. Metode ini menangani permintaan pengunduhan laporan *Audit Trail Step Workflow* dalam format Excel. Setelah proses autentikasi, parameter pencarian dari objek *View* seperti *search*, *status*, *startDate*, *endDate*, dan *branchId* digunakan untuk mengambil data dari database melalui metode *getDataAudittrailStepWorkflowByCriteria()*. Data yang diperoleh

dikonversi menjadi daftar objek AudittrailStepWorkflow dan diproses oleh utilitas loadListDataAudittrailStepWorkflowDownload() untuk menghasilkan file Excel. File tersebut dibungkus dalam objek InputStreamResource dan dikirimkan ke *client* sebagai ResponseEntity.

```
1  @SuppressWarnings({ "unused", "unchecked" })
2  @RequestMapping(value = AudittrailConstant.
3      PROCUREMENT_AUDITTRAIL_STEP_WORKFLOW_DOWNLOAD_EXCEL,
4      method = RequestMethod.POST)
5  @CrossOrigin(origins = "*")
6  @Logging
7  @Authenticate
8  public ResponseEntity<InputStreamResource>
9  downloadAudittrailSWExcel(HttpServletRequest request,
10     HttpServletResponse response, @RequestBody View View)
11    throws Exception {
12        String tokens = request.getHeader(
13            AuthenticationConstant.AUTHORIZATION);
14        authRes = AuthenticationApi.authentication(
15            authentication, tokens);
16
17        Map<String, Object> data = auditDAO.
18        getDataAudittrailStepWorkflowByCriteria(
19            View.getSearch(), View.getStatus(), View.
20            getStartDate(),
21            View.getEndDate(), View.getBranchid(), authRes.
22            getBranchId());
23
24        List<AudittrailStepWorkflow> listAuditSW = (List<
25            AudittrailStepWorkflow>) data.get("data");
26
27        String filename = "REPORT_AUDITTRAIL_SW_";
28        String dateMerge = new SimpleDateFormat(
29            ParameterConstant.DATETIME_YMDHMSSSS_MERGE).format(new
30            Date());
31        String fileExportName = filename + dateMerge + ".xlsx
32        ";
33
34        logger.info("Start Print EXCEL Data Audittrail SW " +
35            new Date().getTime());
36        InputStreamResource file = new InputStreamResource(
37            auditDAO.loadListDataAudittrailStepWorkflowDownload(
38                listAuditSW));
```

```
22     logger.info("End Print EXCEL Data Audittrail SW " +  
23     new Date().getTime());  
24  
25     return ResponseEntity.ok()  
26         .header(HttpHeaders.CONTENT_DISPOSITION, "  
27         attachment; filename=" + fileExportName)  
28         .contentType(MediaType.parseMediaType("application/vnd-ms.excel"))  
29         .body(file);  
30     }  
31 }
```

Kode 3.7: Metode downloadAudittrailSWExcel

Kemudian, dikembangkan *method* baru dalam *class* AudittrailDAOImpl untuk mengambil dan menghitung data berdasarkan kriteria pencarian tertentu yaitu `getDataAudittrailStepWorkflowByCriteria` dan `loadListDataAudittrailStepWorkflowDownload`. Potongan kode dapat dilihat pada Kode 3.8.

- `getDataAudittrailStepWorkflowByCriteria`:
Method ini digunakan untuk mengambil data berdasarkan beberapa kriteria seperti `search`, `status`, `startDate`, `endDate`, `branch`, dan `branchAuth`. Hasil dari *method* ini berupa `Map` yang berisi total data dan daftar hasil pencarian.
 - `loadListDataAudittrailStepWorkflowDownload`:
Method ini mengubah daftar objek `AudittrailTransactionFlow` menjadi format Excel yang siap diunduh sebagai `InputStream`.

```
1     @Override
2     public Map<String, Object>
3         getDataAudittrailStepWorkflowByCriteria(String search,
4             String status, String startDate
5                 , String endDate, List<String> branch, String
6             branchAuth) throws AppException
7     {
8         // TODO Auto-generated method stub
9         Map<String, Object> retValue = new HashMap<String, Object>();
10        Map<String, Object> criteria = new HashMap<String, Object>();
11    }
```

```

8     List<AudittrailStepWorkflow> list = new ArrayList<
9         AudittrailStepWorkflow>();
10        auditStepWorkflowRepository.findAll(
11            builderAuditStepWorkflow.
12            queryBuilderAudittrailStepWorkflow(criteria,
13                search,
14                status,
15                startDate,
16                endDate,
17                branch,
18                branchAuth)).forEach(audittrailId -> list.add(
19                    audittrailId));
20        long t = auditStepWorkflowRepository.count(
21            builderAuditStepWorkflow.
22            queryBuilderAudittrailStepWorkflow(criteria,
23                search,
24                status,
25                startDate,
26                endDate,
27                branch,
28                branchAuth));
29        retVal.put("total", t);
30        retVal.put("data", list);
31        return retVal;
32    }
33

```

@Override
public InputStream
loadListDataAudittrailStepWorkflowDownload(List<
AudittrailStepWorkflow> listData) throws AppException {
ByteArrayInputStream in =
ExcelBuilderAudittrailStepWorkflow.
ParameterAudittrailStepWorkflowToExcel(listData);
return in;
}

Kode 3.8: Metode getDataAudittrailStepWorkflowByCriteria

NUSANTARA
Kemudian, diimplementasikan *class* baru bernama
ExcelBuilderAudittrailStepWorkflow, implementasi
method queryBuilderAudittrailStepWorkflow dalam
class QueryBuilderAudittrail, dan membuat *class*

AudittrailStepWorkflowRepository

- ExcelBuilderAudittrailStepWorkflow:

Class ini berfungsi sebagai utilitas untuk membangun file Excel dari data workflow. Disediakan method ParameterAudittrailStepWorkflowToExcel untuk mengatur isi dan format file, serta hasExcelFormat untuk validasi. Hasil atau screenshot dari kode dapat dilihat pada Lampiran 9.

- queryBuilderAudittrailStepWorkflow:

Method queryBuilderAudittrailStepWorkflow membangun Specification dinamis berdasarkan search, status, rentang transactionDate, serta otorisasi cabang (branchAuth atau listBranch). Pencarian dilakukan dengan cb.like pada beberapa field seperti branchid, rfqNo, workflowId, dan userid, yang digabungkan dengan cb.or. Filter tambahan diterapkan untuk status, cabang, dan tanggal menggunakan cb.equal atau cb.between. Jika branchAuth bernilai "000", maka pencarian dibatasi ke cabang dalam listBranch; jika tidak, hanya cabang otorisasi yang digunakan. Data juga di-join ke entitas workflow dan diurutkan berdasarkan transactionDate secara menurun. Hasil atau screenshot dari kode dapat dilihat pada Lampiran 10.

3.3.3 Audit Trail Contract

Pengembangan API audit trail contract pada modul *Procurement System UFINS* dimulai dengan menambahkan tiga method ke dalam AudittrailController, yaitu viewAudittrailContract, downloadAudittrailContract, dan downloadAudittrailContractExcel.

- viewAudittrailContract:

Pada Kode 3.9 terdapat metode viewAudittrailContract. Metode tersebut menangani permintaan POST untuk melihat data dengan validasi token JWT. Data difilter berdasarkan search, status, startDate, endDate, dan branchId.

```
1  @SuppressWarnings("unchecked")
2  @RequestMapping(value = AudittrailConstant.
PROCUREMENT_AUDITTRAIL_CONTRACT_VIEW, method =
RequestMethod.POST)
```

```

3   @CrossOrigin(origins = "*")
4   @Logging
5   @Authenticate
6   public String viewAudittrailContract (HttpServletRequest
request , HttpServletResponse response , @RequestBody View
View) throws Exception
7   {
8       try
9       {
10          String tokens = request.getHeader(
AuthenticationConstant.AUTHORIZATION);
11          String nezam = request.getHeader(
AuthenticationConstant.NEZNAME);
12          authRes = AuthenticationApi.authentication(
authentication , tokens);
13          if (authRes == null) {
14              return senderAuditManagement.responseNotAuthorized(
request , response);
15          }
16          if (!nezam.equalsIgnoreCase(authRes .
getAuthenticatedMethod())) {
17              return senderAuditManagement.responseNotAuthorized(
request , response);
18          }
19          Map<String , Object> data = auditDAO.
getDataAudittrailContractByCriteria(View.getSearch() , View
.getStatus() , View.getStartDate() , View.getEndDate() , View
.getBranchid() , authRes.getBranchId());
20          List<AudittrailContract> listAuditContract = (List<
AudittrailContract>) data.get("data");
21          long total = (long) data.get("total");
22          if (total == 0) {
23              return senderAuditContract.responseEmpty(request ,
response);
24          }
25          return senderAuditContract.responseListCount(request ,
response , listAuditContract , total);
26      }
27      catch (Exception e) {
28          e.printStackTrace();
29          return senderAuditContract.sendErrorInternalServer(
request , response , e , logger);
30      }

```

```
31     }
32 }
```

Kode 3.9: Metode viewAudittrailContract

- downloadAudittrailContract:

Pada Kode 3.10 terdapat metode downloadAudittrailContract. *Method* ini menghasilkan file laporan PDF dari data *Audit Trail Contract* yang difilter berdasarkan parameter seperti search, status, startDate, endDate, dan branchId. Setelah proses *autentikasi* melalui AuthenticationApi, sistem mengambil userId, branchId, serta tanggal proses terkini. Data kemudian diambil dari *database* dan dikonversi menjadi daftar objek AudittrailContract. Parameter laporan disiapkan dan disisipkan ke dalam *template JasperReports* berformat .jrxml, yang kemudian dikompilasi dan diisi menggunakan JasperFillManager. Hasilnya dikonversi menjadi file PDF.

```
1  @SuppressWarnings("unchecked")
2  @RequestMapping(value = AudittrailConstant.
3      PROCUREMENT_AUDITTRAIL_CONTRACT_VIEW, method =
4      RequestMethod.POST)
5  @CrossOrigin(origins = "*")
6  @Logging
7  @Authenticate
8  public String viewAudittrailContract (HttpServletRequest
9      request , HttpServletResponse response , @RequestBody View
10     View) throws Exception
11  {
12      try
13      {
14          String tokens = request.getHeader(
15              AuthenticationConstant.AUTHORIZATION);
16          String nezam = request.getHeader(
17              AuthenticationConstant.NEZNAM);
18          authRes = AuthenticationApi.authentication(
19              authentication , tokens);
20          if (authRes == null) {
21              return senderAuditManagement.responseNotAuthorized(
22                  request , response);
23          }
24          if (!nezam.equalsIgnoreCase(authRes.
25              getAuthenticatedMethod())) {
```

```

17         return senderAuditManagement.responseNotAuthorized(
18             request, response);
19     }
20     Map<String, Object> data = auditDAO.
21     getDataAudittrailContractByCriteria(View.getSearch(), View
22         .getStatus(), View.getStartDate(), View.getEndDate(), View
23         .getBranchid(), authRes.getBranchId());
24     List<AudittrailContract> listAuditContract = (List<
25         AudittrailContract>) data.get("data");
26     long total = (long) data.get("total");
27     if (total == 0) {
28         return senderAuditContract.responseEmpty(request,
29             response);
30     }
31     return senderAuditContract.responseListCount(request,
32         response, listAuditContract, total);
33 }

```

Kode 3.10: Metode downloadAudittrailContract

- downloadAudittrailContractExcel:

Pada Kode 3.11 terdapat metode downloadAudittrailContractExcel. *Method* ini menangani pembuatan dan pengunduhan laporan *Audit Trail Contract* dalam format Excel, berdasarkan parameter seperti search, status, startDate, endDate, dan branchId. Setelah proses autentikasi melalui AuthenticationApi, data diambil dari database menggunakan method getDataAudittrailContractByCriteria(). Data tersebut diubah menjadi stream Excel melalui method loadListDataAudittrailContractDownload() dan dibungkus dalam InputStreamResource. Nama file dibentuk secara dinamis dengan timestamp dan dikembalikan ke client sebagai file Excel.

```

1 @SuppressWarnings({ "unused", "unchecked" })
2 @RequestMapping(value = AudittrailConstant.
3     PROCUREMENT_AUDITTRAIL_CONTRACT_DOWNLOAD_EXCEL, method =
4     RequestMethod.POST)

```

```

3   @CrossOrigin(origins = "*")
4   @Logging
5   @Authenticate
6   public ResponseEntity<InputStreamResource>
7   downloadAudittrailContractExcel(HttpServletRequest request
8   , HttpServletResponse response, @RequestBody View View)
9   throws Exception
10  {
11      String tokens = request.getHeader(
12          AuthenticationConstant.AUTHORIZATION);
13      authRes = AuthenticationApi.authentication(
14          authentication, tokens);
15
16      Map<String, Object> data = auditDAO.
17      getDataAudittrailContractByCriteria(View.getSearch(), View
18          .getStatus(), View.getStartDate(), View.getEndDate(), View
19          .getBranchid(), authRes.getBranchId());
20
21      List<AudittrailContract> listAuditContract = (List<
22          AudittrailContract>) data.get("data");
23      String filename = "REPORT_AUDITTRAIL_CONTRACT_";
24      Date now = new Date();
25      String dateMerge = new SimpleDateFormat(
26          ParameterConstant.DATETIME_YMDHMSSSS_MERGE).format(now);
27      String fileExportName = filename + dateMerge + ".xlsx";
28      logger.info("Start Print EXCEL Data Audittrail Contract
29      " + new Date().getTime());
30      InputStreamResource file = new InputStreamResource(
31          auditDAO.loadListDataAudittrailContractDownload(
32              listAuditContract));
33      logger.info("End Print EXCEL Data Audittrail Contract "
34      + new Date().getTime());
35      return ResponseEntity.ok().header(HttpHeaders.
36          CONTENT_DISPOSITION, "attachment; filename=" +
37          fileExportName).contentType(MediaType.parseMediaType("
38          application/vnd.ms-excel")).body(file);
39  }
40
41

```

NUSANTARA

Kemudian, dikembangkan *method* baru dalam class AudittrailDAOImpl untuk mengambil dan menghitung data berdasarkan kriteria pencarian

tertentu, yaitu `getDataAudittrailContractByCriteria` dan `loadListDataAudittrailContractDownload`.

- `getDataAudittrailContractByCriteria`
Method ini digunakan untuk mengambil data berdasarkan beberapa kriteria seperti `search`, `status`, `startDate`, `endDate`, `branch`, dan `branchAuth`. Hasil dari *method* ini berupa `Map` yang berisi total data dan daftar hasil pencarian.
- `loadListDataAudittrailContractDownload`
Method ini mengubah daftar objek `AudittrailContract` menjadi format Excel yang siap diunduh sebagai `InputStream`.

```
1  @Override
2  public Map<String , Object>
3      getDataAudittrailContractByCriteria( String search , String
4          status , String startDate
5          , String endDate , List<String> branch , String
6          branchAuth) throws AppException
7  {
8      // TODO Auto-generated method stub
9      Map<String , Object> retValue = new HashMap<String , Object
10     >();
11      Map<String , Object> criteria = new HashMap<String , Object
12     >();
13      List<AudittrailContract> list = new ArrayList<
14          AudittrailContract >();
15      auditContractRepository .findAll(builderAuditContract .
16      queryBuilderAudittrailContract(criteria ,
17          search ,
18          status ,
19          startDate ,
20          endDate ,
21          branch ,
22          branchAuth)).forEach( audittrailId -> list.add(
23              audittrailId));
24      long t = auditContractRepository .count(
25      builderAuditContract .queryBuilderAudittrailContract(
26          criteria ,
27          search ,
28          status ,
29          startDate ,
30          endDate ,
```

```

21         branch ,
22         branchAuth));
23     retValue.put("total", t);
24     retValue.put("data", list);
25     return retValue;
26 }
27
28 @Override
29 public InputStream loadListDataAudittrailContractDownload(
30     List<AudittrailContract> listData) throws AppException {
31     // TODO Auto-generated method stub
32     ByteArrayInputStream in = ExcelBuilderAudittrailContract.
33     ParameterAudittrailContractToExcel(listData);
34     return in;
35 }
```

Kode 3.12: Metode getDataAudittrailContractByCriteria

Kemudian, diimplementasikan *class* baru bernama *ExcelBuilderAudittrailContract*, implementasi *method* *queryBuilderAudittrailContract* dalam *class* *QueryBuilderAudittrail*, dan membuat *class* *AudittrailContractRepository*.

- *ExcelBuilderAudittrailContract*:
Class ini berfungsi sebagai utilitas untuk membangun file Excel dari data kontrak. Disediakan *method* *ParameterAudittrailContractToExcel* untuk mengatur isi dan format file, serta *hasExcelFormat* untuk validasi. Hasil atau *screenshot* dari kode dapat dilihat pada Lampiran 11.
- *queryBuilderAudittrailContract*:
Metode *queryBuilderAudittrailContract* membangun *Specification* dinamis berdasarkan parameter pencarian (*search*), *status*, *transactionDate*, dan *branchAuth*. Pencarian menggunakan *cb.like* pada beberapa *field* dan digabung dengan *cb.or*. Filter tambahan diterapkan berdasarkan *status*, *branchAuth*, serta rentang tanggal (*cb.equal* atau *cb.between*). Hasil diurutkan menurun berdasarkan *transactionDate*. Cuplikan kode terdapat pada Lampiran 12.

3.3.4 Audit Trail Parameter

Pengembangan API audit trail parameter dimulai dengan menambahkan tiga method ke dalam AudittrailController, yaitu reportAuditParameterView, reportAuditParameterDownload, dan downloadAudittrailParameterExcel.

- reportAuditParameterView:

Menangani permintaan *POST* untuk melihat data dengan validasi token *JWT*. Data difilter berdasarkan search, status, startDate, endDate, dan branchId.

```
1  @SuppressWarnings("unchecked")
2  @RequestMapping(value = AudittrailConstant.
3      PROCUREMENT_AUDITTRAIL_PARAMETER_VIEW, method =
4      RequestMethod.POST)
5  @CrossOrigin(origins = "*")
6  @Logging
7  @Authenticate
8  public String reportAuditParameterView (
9      HttpServletRequest request, HttpServletResponse response ,
10     @RequestBody View View) throws Exception {
11     try {
12         String tokens = request.getHeader(
13             AuthenticationConstant.AUTHORIZATION);
14         String nezam = request.getHeader(
15             AuthenticationConstant.NEZNAME);
16         authRes = AuthenticationApi.authentication(
17             authentication , tokens);
18         if (authRes == null) return senderAuditParameter.
19             responseNotAuthorized(request , response);
20         if (!nezam.equalsIgnoreCase(authRes .
21             getAuthenticatedMethod())))
22             return senderAuditParameter.responseNotAuthorized(
23                 request , response);
24
25         Map<String , Object> data = auditDAO.
26             getDataAuditParameter(View .getSearch() , View .getStatus() ,
27             View .getStartDate() , View .getEndDate() , View .getBranchid() ,
28             authRes .getBranchId());
29
30         List<AudittrailParameter> listAuditParameter = (List<
31             AudittrailParameter>) data .get("data");
```

```

18     long total = (long) data.get("total");
19     if (total == 0) return senderAuditParameter.
20     responseEmpty(request, response);
21     return senderAuditParameter.responseListCount(request
22     , response, listAuditParameter, total);
23   }
24   catch (Exception e) {
25     e.printStackTrace();
26     return senderAuditParameter.sendErrorInternalServer(
27     request, response, e, logger);
28   }
29 }
```

Kode 3.13: Metode reportAuditParameterView

- reportAuditParameterDownload:

Method ini menghasilkan file laporan PDF dari data *Audit Trail Parameter* yang difilter berdasarkan parameter seperti search, status, startDate, endDate, dan branchId. Setelah proses *autentikasi* melalui AuthenticationApi, sistem mengambil userId, branchId, serta tanggal proses terkini. Data kemudian diambil dari *database* dan dikonversi menjadi daftar objek AudittrailParameter. Parameter laporan disiapkan dan disisipkan ke dalam *template JasperReports* berformat .jrxml, yang kemudian dikompilasi dan diisi menggunakan JasperFillManager. Hasilnya dikonversi menjadi file PDF.

```

1 @SuppressWarnings({ "unchecked", "unused" })
2 @RequestMapping(AudittrailConstant.
3   PROCUREMENT_AUDITTRAIL_PARAMETER_DOWNLOAD_PDF)
4 @CrossOrigin(origins = "*")
5 @Logging
6 @Authenticate
7 public void reportAuditParameterDownload(
8   HttpServletRequest request, HttpServletResponse response,
9   @RequestBody View View) throws Exception {
10
11   String tokens = request.getHeader(
12     AuthenticationConstant.AUTHORIZATION);
13   authRes = AuthenticationApi.authentication(
14     authentication, tokens);
15
16   SimpleDateFormat dateFormat3 = new SimpleDateFormat("
17     yyyy-MM-dd HH:mm:ss");
18   String user = authRes.getUserId();
19   ProcessDate pros = reportdao.getProcessDate();
20   Date prosdate = SetProcessDate.getprosdate(pros);
21   String datetime = dateFormat3.format(prosdate);
22   List<JasperPrint> jasperPrints = new ArrayList<>();
23   String startDate = View.getStartDate();
24   String endDate = View.getEndDate();
```

```

19     Map<String, Object> data = auditDAO.
20     getDataAuditParameter(View.getSearch(), View.getStatus(),
21     View.getStartDate(), View.getEndDate(), View.getBranchid(),
22     , authRes.getBranchId());
23
24     List<AudittrailParameter> listAuditParameter = (List<
25     AudittrailParameter>) data.get("data");
26
27     String fileName = templateLocation + "
28     Audittrail_Parameter.jrxml";
29     Map<String, Object> parameters = new HashMap<String,
30     Object>();
31     parameters.put("P_REPORT_LOCATION", templateLocation +
32     "logo.png");
33     parameters.put("P_USER", user);
34     parameters.put("P_PROCESSDATE", datetime);
35     JasperReport jp = JasperCompileManager.compileReport(
36     fileName);
37     JasperPrint print = JasperFillManager.fillReport(jp,
38     parameters, new JRBeanCollectionDataSource(
39     listAuditParameter));
40     jasperPrints.add(print);
41
42     PDF pdf = new PDF(response, "
43     REPORT_AUDITTRAIL_PARAMETER_", ".pdf", print);
44 }

```

Kode 3.14: Metode reportAuditParameterDownload

- downloadAudittrailParameterExcel:

Method ini menangani pembuatan dan pengunduhan laporan Audit Trail Parameter dalam format Excel, berdasarkan parameter seperti search, status, startDate, endDate, dan branchId. Setelah proses autentikasi melalui AuthenticationApi, data diambil dari database menggunakan method getDataAudittrailParameterByCriteria(). Data tersebut diubah menjadi stream Excel melalui method loadListDataAudittrailParameterDownload() dan dibungkus dalam InputStreamResource. Nama file dibentuk secara dinamis dengan timestamp dan dikembalikan ke client sebagai file Excel.

```

1     @SuppressWarnings({ "unchecked", "unused" })
2     @RequestMapping(AudittrailConstant.
3     PROCUREMENT_AUDITTRAIL_PARAMETER_DOWNLOAD_PDF)
4     @SuppressWarnings("unchecked")
5     @RequestMapping(AudittrailConstant.
6     PROCUREMENT_AUDITTRAIL_PARAMETER_DOWNLOAD_EXCEL)
7     @CrossOrigin(origins = "*")
8     @Authenticate
9     @Logging
10    public ResponseEntity<InputStreamResource>
11    downloadAudittrailParameterExcel(HttpServletRequest
12    request, HttpServletResponse response, @RequestBody View
13    View) throws Exception {
14
15        String tokens = request.getHeader(
16        AuthenticationConstant.AUTHORIZATION);

```

```

11     authRes = AuthenticationApi.authentication(
12         authentication, tokens);
13     Map<String, Object> data = auditDAO.
14     getDataAuditParameter(View.getSearch(), View.getStatus(),
15     View.getStartDate(), View.getEndDate(), View.getBranchid()
16     , authRes.getBranchId());
17     List<AudittrailParameter> listAuditParameter = (List<
18     AudittrailParameter>) data.get("data");
19     String filename = "REPORT_AUDITTRAIL_PARAMETER_";
20     Date now = new Date();
21     String dateMerge = new SimpleDateFormat(
22         ParameterConstant.DATETIME_YMDHMSSSS_MERGE).format(now);
23     String fileExportName = filename + dateMerge + ".xlsx";
24     logger.info("Start Print EXCEL Data Audittrail
25     Parameter " + new Date().getTime());
26     InputStreamResource file = new InputStreamResource(
27         auditDAO.loadListDataAudittrailParameterDownload(
28             listAuditParameter));
29     logger.info("End Print EXCEL Data Audittrail Parameter"
30     + new Date().getTime());
31     return ResponseEntity.ok().header(HttpHeaders.
32         CONTENT_DISPOSITION, "attachment; filename=" +
33         fileExportName).contentType(MediaType.parseMediaType("
34         application/vnd.ms-excel")).body(file);
35 }
```

Kode 3.15: Metode downloadAudittrailParameterExcel

Kemudian, diimplementasikan *class* baru bernama ExcelBuilderAudittrailParameter, implementasi *method* queryBuilderAudittrailParameter dalam *class* QueryBuilderAudittrail, dan membuat *class* AudittrailParameterRepository.

- ExcelBuilderAudittrailParameter:
Class ini berfungsi sebagai utilitas untuk membangun file Excel dari data parameter. Disediakan *method* ParameterAudittrailParameterToExcel untuk mengatur isi dan format file, serta hasExcelFormat untuk validasi. Hasil atau *screenshot* dari kode dapat dilihat pada Lampiran 13.
- queryBuilderAudittrailParameter:
Metode queryBuilderAudittrailParameter membangun *Specification* dinamis berdasarkan parameter pencarian (search), status, transactionDate, dan branchAuth. Pencarian menggunakan cb.like pada beberapa field dan digabung dengan cb.or. Filter tambahan diterapkan berdasarkan status, branchAuth, serta rentang tanggal (cb.equal atau cb.between). Hasil diurutkan menurun berdasarkan transactionDate. Cuplikan kode terdapat pada Lampiran 14.

3.3.5 Hasil

Setiap *API endpoint* akan melalui proses *debugging* dan uji fungsionalitas. Tiga *method* utama yang diuji pada setiap *endpoint* adalah *view*, *download PDF* dan *download Excel*. Berikut adalah contoh pengujian yang dilakukan menggunakan *Postman*.

A API Audit Trail Step Workflow

API Audit Trail Step Workflow memiliki 3 *endpoint* utama yaitu:

- procurement/audittrail/step/workflow/view
- procurement/audittrail/step/workflow/download/pdf
- procurement/audittrail/step/workflow/download/excel

Method view digunakan untuk mengambil dan menampilkan data berdasarkan kriteria pencarian. *Hit API* menggunakan metode **POST** ke alamat **http://localhost:3023/procurement/audittrail/step/workflow/view**, yang akan memanggil *method* `viewAudittrailSW` dan mengembalikan data dalam format **JSON**, seperti contoh berikut:

```
1 {
2     "auditId": 127,
3     "transactionDate": 1747737894959,
4     "branchid": "001",
5     "date": 1747674000000,
6     "rfqNo": "001RFQ202505000113",
7     "workflowId": "BID1",
8     "stepId": "T",
9     "userid": "SIT2",
10    "status": "APPROVE",
11    "message": "SUCCESS",
12    "workflowName": "BARANG"
13 }
```

Method download PDF digunakan untuk menghasilkan dan mengunduh laporan audit trail dalam format PDF, dengan *template* yang dibuat menggunakan *Jaspersoft Studio*. API dipanggil melalui metode **POST** ke alamat **http://localhost:3023/procurement/audittrail/step/workflow/download/pdf**,

yang memanggil *method* downloadAudittrailSW dan mengembalikan file PDF. Jika diuji melalui Postman, file ditampilkan dalam bentuk mentah (*raw*) dan harus diunduh terlebih dahulu agar dapat dibuka dengan PDF *reader*. Output dapat dilihat pada Gambar 3.1.

AUDITTRAIL STEP WORKFLOW										2025-01-02 14:20:09
No	Transaction Date	Branchid	RFQ No	Workflow ID	Workflow Name	Step ID	User Request	User ID	Status	Message
1	2025-05-20 17:44:54	001	001RFQ202505000113	BID1	BARANG	T	-	SIT2	APPROVE	SUCCESS
2	2025-05-20 17:44:33	001	001RFQ202505000113	BID1	BARANG	T	-	SIT1	APPROVE	SUCCESS
3	2025-05-20 17:39:06	001	001RFQ202505000113	BID1	BARANG	T	-	SIT2	APPROVE	SUCCESS
4	2025-05-20 17:38:29	001	001RFQ202505000113	BID1	BARANG	T	-	SIT1	APPROVE	SUCCESS
5	2025-05-20 17:16:32	001	001RFQ202505000113	BID1	BARANG	T	-	SIT2	APPROVE	SUCCESS

Gambar 3.2. Contoh Output Hasil PDF API Step Workflow

Method download Excel dengan alamat **http://localhost:3023/procurement/audittrail/step/workflow/download/excel** akan memanggil *method* downloadAudittrailsExcel dan mengembalikan file Excel. Saat diuji di Postman, file akan muncul dalam format mentah dan perlu diunduh terlebih dahulu untuk dapat dibuka, seperti pada Gambar 3.2 dibawah.

AUDIT_ID	TRANSACTION_DATE	BRANCHID	DATE	RFQ_NO	WORKFLOW_ID	WORKFLOW_NAME	STEP_ID	USERID	STATUS	MESSAGE	USER_REQ
1	127 2025-05-20 17:44:54	001	2025-05-20	001RFQ202505000113	BID1	BARANG	T	SIT2	APPROVE	SUCCESS	-
2	126 2025-05-20 17:44:33	001	2025-05-20	001RFQ202505000113	BID1	BARANG	T	SIT1	APPROVE	SUCCESS	-
3	125 2025-05-20 17:39:06	001	2025-05-20	001RFQ202505000113	BID1	BARANG	T	SIT2	APPROVE	SUCCESS	-
4	124 2025-05-20 17:38:29	001	2025-05-20	001RFQ202505000113	BID1	BARANG	T	SIT1	APPROVE	SUCCESS	-
5	123 2025-05-20 17:16:32	001	2025-05-20	001RFQ202505000113	BID1	BARANG	T	SIT2	APPROVE	SUCCESS	-

Gambar 3.3. Contoh Output Hasil Excel API Step Workflow

B API Audit Trail Contract

API Audit Trail Contract memiliki 3 *endpoint* utama yaitu:

- procurement/audittrail/contract/view
- procurement/audittrail/contract/download/pdf
- procurement/audittrail/contract/download/excel

Method view digunakan untuk mengambil dan menampilkan data berdasarkan kriteria pencarian. Hit API menggunakan metode **POST** ke alamat **http://localhost:3023/procurement/audittrail/contract/view**, yang akan memanggil *method* viewAudittrailContract dan mengembalikan data dalam format **JSON**, seperti contoh berikut:

```

1 {
2   "auditId": 23,
3   "transactionDate": 1739871186160,
4   "branchid": "001",
5   "contractNo": "001AGR202502000015",
6   "contractDate": 1739811600000,
7   "receivedBy": "SIT2",
8   "vendorId": "VD20240000",
9   "poNo": "001PO202501000019",
10  "status": "APPROVE"
11 },

```

Method download PDF digunakan untuk menghasilkan dan mengunduh laporan audit trail dalam format PDF, dengan *template* yang dibuat menggunakan *Jaspersoft Studio*. API dipanggil melalui metode **POST** ke alamat <http://localhost:3023/procurement/audittrail/contract/download/pdf>, yang memanggil *method* downloadAudittrailSW dan mengembalikan file PDF. Jika diuji melalui Postman, file ditampilkan dalam bentuk mentah (*raw*) dan harus diunduh terlebih dahulu agar dapat dibuka dengan PDF reader. Output dapat dilihat pada Gambar

AUDITTRAIL CONTRACT										2025-01-02 11:49:40
No	Transaction Date	Branch ID	Contract No.	Contract Date	Received By	Vendor ID	PO No.	User ID	Status	Message
1	2025-02-18 16:33:06	001	001AGR202502000015	2/18/25, 12:00 AM	SIT2	VD20240000	001PO202501000019	-	APPROVE	-
2	2025-02-18 16:31:17	001	001AGR202502000015	2/18/25, 12:00 AM	SIT2	VD20240000	001PO202501000019	-	REJECT	-
3	2025-02-18 16:30:35	001	001AGR202502000015	2/18/25, 12:00 AM	SIT2	VD20240000	001PO202501000019	-	INSERT	-
4	2024-12-24 00:26:59	001	001AGR202412000014	12/24/24, 12:00 AM	SIT1	VD20240004	001PO202403000017	-	APPROVE	-
5	2024-12-24 00:26:53	001	001AGR202412000014	12/24/24, 12:00 AM	SIT1	VD20240004	001PO202403000017	-	REJECT	-

Gambar 3.4. Contoh Output Hasil PDF API Contract

Method download Excel dengan alamat <http://localhost:3023/procurement/audittrail/contract/download/excel> akan memanggil *method* downloadAudittrailContractExcel dan mengembalikan file Excel. Saat diuji di Postman, file akan muncul dalam format mentah dan perlu diunduh terlebih dahulu untuk dapat dibuka seperti pada Gambar 3.4 di bawah.

C API Audit Trail Parameter

API Audit Trail Parameter memiliki 3 *endpoint* utama, yaitu:

AUDIT_ID	TRANSACTION_DATE	BRANCHID	CONTRACT_NO	CONTRACT_DATE	RECEIVED_BY	VENDORID	PO_NO	USERID	STATUS	MESSAGE
2	23 2025-02-18 16:33:06	001	001AGR2025020C2025-02-18	SIT2	VD20240000	001PO202501000019	-	APPROVE-		
3	22 2025-02-18 16:31:17	001	001AGR2025020C2025-02-18	SIT2	VD20240000	001PO202501000019	-	REJECT	-	
4	21 2025-02-18 16:30:35	001	001AGR2025020C2025-02-18	SIT2	VD20240000	001PO202501000019	-	INSERT	-	
5	20 2024-12-24 00:26:59	001	001AGR2024120C2024-12-24	SIT1	VD20240004	001PO202403000017	-	APPROVE-		
6	19 2024-12-24 00:26:53	001	001AGR2024120C2024-12-24	SIT1	VD20240004	001PO202403000017	-	REJECT	-	
7	18 2024-12-24 00:26:19	001	001AGR2024120C2024-12-24	SIT1	VD20240004	001PO202403000017	-	INSERT	-	
8	17 2024-12-23 14:51:34	001	001AGR2024120C2024-12-23	SIT1	VD20240009	001PO202403000016	-	APPROVE-		
9	16 2024-12-23 14:49:14	001	001AGR2024120C2024-12-23	SIT1	VD20240009	001PO202403000016	-	REJECT	-	
10	15 2024-12-23 14:47:38	001	001AGR2024120C2024-12-23	SIT1	VD20240009	001PO202403000016	-	INSERT	-	

Gambar 3.5. Contoh Output Hasil Excel API Contract

- procurement/audittrail/parameter/view
- procurement/audittrail/parameter/download/pdf
- procurement/audittrail/parameter/download/excel

Method view digunakan untuk mengambil dan menampilkan data berdasarkan kriteria pencarian. *Hit API* menggunakan metode **POST** ke alamat **http://localhost:3023/procurement/audittrail/parameter/view**, yang akan memanggil *method reportAuditParameterView* dan mengembalikan data dalam format **JSON**, seperti contoh berikut:

```

1 {
2   "audittrailId": 1566,
3   "action": "APPROVE",
4   "transactionDate": 1750072179885,
5   "tableName": "REF_STORAGE_LOCATION_APP",
6   "userId": "SIT2",
7   "auditDesc": "ACTIVE INV REPLISHMENT GROUP",
8   "fieldName": "STORAGEROOM_ID",
9   "oldValue": "TEST1",
10  "newValue": "-",
11  "keyValue": "[STORAGEROOMID]",
12  "branchid": "000"
13 }

```

Method ini menghasilkan dan mengunduh laporan *audit trail* dalam format PDF menggunakan *template* dari *Jaspersoft Studio*. *API* dipanggil dengan metode **POST** ke **http://localhost:3023/procurement/audittrail/parameter/download/pdf**, yang memicu *method reportAuditParameterDownload* dan mengembalikan file PDF. Saat diuji di Postman, file ditampilkan dalam format *raw* dan perlu diunduh terlebih dahulu untuk dibuka di PDF reader.

PT Bank CTBC Indonesia AUDITTRAIL PARAMETER									CTBC BANK	
No	Transaction Date	Table Name	Field Name	Old Value	New Value	User ID	Branch ID	Action	Audit Desc	Key Value
1	2025-06-16 18:09:39.885	REF_STORAGE_LOCATION_APP	STORAGEROOM_ID	TEST1	-	SIT2	000	APPROVE	ACTIVE INV REPLISHMENT	[STORAGEROOMID]
2	2025-06-16 18:09:39.885	REF_STORAGE_LOCATION_APP	DESCRIPTION2	AAA	-	SIT2	000	APPROVE	ACTIVE INV REPLISHMENT	[STORAGEROOMID]
3	2025-06-16 18:09:39.885	REF_STORAGE_LOCATION_APP	DESCRIPTION	DESC1ED	-	SIT2	000	APPROVE	ACTIVE INV REPLISHMENT	[STORAGEROOMID]
4	2025-06-16 18:09:39.885	REF_STORAGE_LOCATION_APP	STORAGEROOM_PIC	NANDAED	-	SIT2	000	APPROVE	ACTIVE INV REPLISHMENT	[STORAGEROOMID]
5	2025-06-16 18:09:39.885	REF_STORAGE_LOCATION_APP	BRANCH_ID	999	-	SIT2	000	APPROVE	ACTIVE INV REPLISHMENT	[STORAGEROOMID]
6	2025-06-16 18:09:36.807	REF_STORAGE_LOCATION	BRANCH_ID	999	999	SIT1	000	ACTIVE	ACTIVE INV REPLISHMENT	[STORAGEROOMID]
7	2025-06-16 18:09:36.807	REF_STORAGE_LOCATION	DESCRIPTION2	AAA	AAA	SIT1	000	ACTIVE	ACTIVE INV REPLISHMENT	[STORAGEROOMID]

Gambar 3.6. Contoh Output Hasil PDF API Parameter

Method ini mengunduh laporan audit trail dalam format Excel melalui alamat <http://localhost:3023/procurement/audittrail/parameter/download/excel>, yang memanggil *method* downloadAudittrailParameterExcel.

1	TRANSACTION_DATE	TABLE_NAME	FIELD_NAME	OLD_VALUE	NEW_VALUE	USER	ACTION	AUDIT_DESC	KEY_VALUE
2	2025-06-16 18:09:39	REF_STORAGE_LOCATION_APP	STORAGEROOM_ID	TEST1	-	SIT2	APPROVE	ACTIVE INV REPLISHMENT GROUP	[STORAGEROOMID]
3	2025-06-16 18:09:39	REF_STORAGE_LOCATION_APP	DESCRIPTION2	AAA	-	SIT2	APPROVE	ACTIVE INV REPLISHMENT GROUP	[STORAGEROOMID]
4	2025-06-16 18:09:39	REF_STORAGE_LOCATION_APP	DESCRIPTION	DESC1ED	-	SIT2	APPROVE	ACTIVE INV REPLISHMENT GROUP	[STORAGEROOMID]
5	2025-06-16 18:09:39	REF_STORAGE_LOCATION_APP	STORAGEROOM_PIC	NANDAED	-	SIT2	APPROVE	ACTIVE INV REPLISHMENT GROUP	[STORAGEROOMID]
6	2025-06-16 18:09:39	REF_STORAGE_LOCATION_APP	BRANCH_ID	999	-	SIT2	APPROVE	ACTIVE INV REPLISHMENT GROUP	[STORAGEROOMID]
7	2025-06-16 18:09:36	REF_STORAGE_LOCATION	BRANCH_ID	999	999	SIT1	ACTIVE	ACTIVE INV REPLISHMENT GROUP	[STORAGEROOMID]
8	2025-06-16 18:09:36	REF_STORAGE_LOCATION	DESCRIPTION2	AAA	AAA	SIT1	ACTIVE	ACTIVE INV REPLISHMENT GROUP	[STORAGEROOMID]
9	2025-06-16 18:09:36	REF_STORAGE_LOCATION	DESCRIPTION	DESC1ED	DESC1ED	SIT1	ACTIVE	ACTIVE INV REPLISHMENT GROUP	[STORAGEROOMID]

Gambar 3.7. Contoh Output Hasil Excel API Parameter

3.4 Kendala dan Solusi yang Ditemukan

- Kendala:** Data tidak muncul pada tampilan frontend meskipun proses pengambilan data dari *database* berhasil dilakukan.

Solusi: Data yang dikirim dari backend belum sesuai dengan struktur atau format yang diharapkan oleh frontend. Penyesuaian dilakukan pada response *controller* agar struktur data dan nama *field* sesuai dengan kebutuhan tampilan.

- Kendala:** Nama variabel yang tidak sinkron atau tidak konsisten sehingga menyebabkan *server* mengembalikan kode error 500 (*Internal Server Error*) saat *request*. Kode error 500 merupakan indikasi bahwa telah terjadi kesalahan *serverside*, biasanya disebabkan oleh *exceptions*, kesalahan logika, atau masalah konfigurasi pada *backend*.

Solusi: Dilakukan penelusuran kesalahan dengan membaca *stack trace* untuk mengidentifikasi bagian kode yang bermasalah. Selanjutnya, dilakukan penyesuaian agar nama-nama variabel yang digunakan di seluruh komponen menjadi konsisten. Proses ini diulangi sampai *server* memberikan respons dengan kode 200 (*OK*).

- **Kendala:** Ditemukan masalah kompatibilitas versi pada *Jaspersoft Studio* yang menyebabkan beberapa elemen laporan tidak dapat dimuat atau dijalankan dengan benar. Selain itu, diperlukan penyesuaian nama-nama *field* di *Jaspersoft* agar sesuai dengan nama variabel yang terdapat pada kelas *entity* di *backend*.

Solusi: Masalah kompatibilitas diatasi dengan menjalankan *Jaspersoft Studio* dalam mode *compatibility*, yang memungkinkan akses dan penggunaan format atau fitur dari versi *Jaspersoft Studio* yang lama. Selain itu, nama-nama *field* dalam file *.jrxml* disesuaikan dengan atribut pada kelas *entity* agar data dapat ditampilkan dengan benar saat proses *rendering* laporan.

