

BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Kegiatan program kerja magang dilakukan di Kantor Akuntan Publik XYZ dan ditempatkan pada divisi *Global Technology Solution (GTS)* pada tim *Application Development* sebagai *database analyst* yang dibimbing langsung oleh Bapak Iman Budiman yang menduduki posisi sebagai *Manager*. Tim *Application Development* terdiri dari 10 karyawan tetap dan 3 peserta magang.

Database analyst pada Kantor Akuntan Publik XYZ berperan untuk memelihara penyimpanan data, membuat desain *database*, dan mengumpulkan, mengatur, dan menafsirkan informasi statistik berdasarkan data dalam *database*, serta mengakomodir kebutuhan *database development* di Kantor Akuntan Publik XYZ. Selain peran utama tersebut.

3.2 Tugas dan Uraian Kerja Magang

Selama kerja magang, mahasiswa mempelajari beberapa hal baru antara lain penggunaan *visualization tools* Microsoft Power BI, penggunaan *Relational Database Management System* Microsoft SQL Server, *development* pada *database middle tier*, dan lain sebagainya. Tabel 3.1 berikut merupakan *timeline* kerja magang yang dilakukan oleh peserta magang.

Tabel 3. 1 Timeline kerja magang

No	Pekerjaan yang dilakukan	Minggu	Tanggal Mulai	Tanggal Selesai
1	<i>Briefing</i> dan pengenalan perusahaan Kantor Akuntan Publik XYZ	1	1 Agustus 2022	5 Agustus 2022
2	Membuat dan menambahkan <i>stored procedure</i> untuk aplikasi Konsultasi Pajak	1 – 9	2 Agustus 2022	26 September 2022

No	Pekerjaan yang dilakukan	Minggu	Tanggal Mulai	Tanggal Selesai
	2.1 <i>Kick off enhancement</i> aplikasi Konsultasi Pajak	1	2 Agustus 2022	2 Agustus 2022
	Mengumpulkan <i>requirements</i> untuk melakukan perubahan pada <i>stored procedure</i>	1 – 2	3 Agustus 2022	12 Agustus 2022
	2.2 Melakukan perubahan pada <i>stored procedure</i>	3 – 5	15 Agustus 2022	2 September 2022
	2.3 <i>Beta Fixing</i>	6 – 7	5 September 2022	14 September 2022
	2.4 <i>User assessment test (UAT) fixing</i>	7 – 8	15 September 2022	23 September 2022
	2.5 Perubahan pada aplikasi Konsultasi Pajak di <i>upload</i> ke server <i>production</i>	9	26 September 2022	26 September 2022
3	Membuat visualisasi report pada IM	2 – 5	11 Agustus 2022	6 September 2022
	3.1 Membuat visual atau <i>dashboard</i>	2 – 4	11 Agustus 2022	24 Agustus 2022
	3.2 Melakukan revisi dari visual atas <i>dashboard</i> yang ada	5	25 Agustus 2022	6 September 2022
4	Mengubah <i>stored procedure</i> untuk <i>Survey & Hot Review</i> pada aplikasi Konsultasi Pertanyaan	3 – 5	16 Agustus 2022	7 September 2022
5	Mengubah dan menambahkan <i>stored procedure</i> untuk aplikasi Permintaan Akses dan Permintaan Pemulihan (ARS)	6 – Continue	6 September 2022	Continue

No	Pekerjaan yang dilakukan	Minggu	Tanggal Mulai	Tanggal Selesai
	5.1 <i>Kick off</i> serta memahami <i>requirement</i> yang ada	6	6 September 2022	9 September 2022
	5.2 Membuat dan mengubah <i>stored procedure</i> sesuai dengan <i>requirement</i> yang ada	7 – 14	12 September 2022	4 November 2022
	5.3 <i>Beta fixing</i> atas <i>error</i> yang ada	15 – 20	7 November 2022	16 Desember 2022
	5.4 <i>User Acceptance Test (UAT)</i>	21 – <i>Continue</i>	19 Desember 2022	<i>Continue</i>
6	Membuat <i>stored procedure</i> serta merancang struktur <i>database</i> untuk aplikasi Kamus Besar Perpajakan	6 – <i>Continue</i>	8 September 2022	<i>Continue</i>
	6.1 <i>Kick off</i> serta memahami <i>requirements</i>	6 – 7	8 September 2022	12 September 2022
	6.2 Membahas desain <i>database</i> yang sesuai dengan senior & <i>project manager (PM)</i>	7 – 8	13 September 2022	23 September 2022
	6.3 Membuat desain <i>Entity Relationship Diagram (ERD)</i>	8	26 September 2022	29 September 2022
	6.4 Pengembangan <i>database</i>	8 – 11	30 September 2022	21 Oktober 2022
	6.5 Pengembangan <i>stored procedure, package, dan lainnya</i>	12 – 17	24 Oktober 2022	2 Desember 2022

No	Pekerjaan yang dilakukan	Minggu	Tanggal Mulai	Tanggal Selesai
	6.6 <i>Beta fixing</i>	18 – <i>Continue</i>	5 Desember 2022	<i>Continue</i>
7	Membuat dan menambahkan <i>stored procedure</i> untuk aplikasi Aplikasi Perpajakan	7 – <i>Continue</i>	12 September 2022	<i>Continue</i>
	7.1 <i>Kick off</i> dan memahami requirement yang ada	7	12 September 2022	16 September 2022
	7.2 Pengembangan <i>stored procedure</i>	8 – <i>Continue</i>	19 September 2022	<i>Continue</i>
8	Melakukan update data pada Enhancement NTPM (Nilai Transaksi Pasar Modal)	10 – 12	13 Oktober 2022	28 Oktober 2022
9	Membuat visualisasi report <i>stock card</i> pada IM	10 – 12	10 Oktober 2022	28 Oktober 2022
10	Membuat dan menambahkan <i>stored procedure</i> pada aplikasi <i>Morning News</i>	16 – <i>Continue</i>	23 November 2022	<i>Continue</i>
	10.1 <i>Kick off</i> dan memahami requirement	16	23 November 2022	25 November 2022
	10.2 Pengembangan <i>stored procedure</i>	17 – 19	28 November 2022	16 Desember 2022
	10.3 <i>Beta Fixing</i>	20 – <i>Continue</i>	19 Desember 2022	<i>Continue</i>

3.2.1 Briefing dan pengenalan perusahaan Kantor Akuntan Publik XYZ (Minggu 1)

Pada minggu pertama program kerja magang, *coach* dan *buddy* magang memberikan *briefing* untuk mengenalkan perusahaan yaitu berupa *tour* area kantor, *tour* area sekitar kantor, dan perkenalan dengan anggota divisi *Global Technology Solution* (GTS). Juga terdapat perkenalan terhadap pekerjaan yang akan dilaksanakan oleh peserta magang sebagai *database analyst*, terdapat juga perkenalan terhadap beberapa aplikasi yang digunakan oleh peserta magang antara lain tools RDBMS (*Relational Database Management System*) Microsoft SQL Server dan tools visualisasi yaitu Microsoft Power BI. Alasan penggunaan kedua aplikasi tersebut karena tim *application development* pada KAP XYZ sudah menggunakan kedua aplikasi tersebut.

3.2.2 Membuat dan menambahkan stored procedure untuk aplikasi Konsultasi Pajak (Minggu 1-8)

Project pertama yang di berikan untuk di buat adalah sebuah aplikasi berbasis web yang user akan memasukkan informasi yang diperlukan dan nantinya akan digunakan untuk membahas sebuah topik, setelah waktu berdiskusi selesai maka akan membuahkan sebuah kesimpulan yang dapat dibaca dan dilihat oleh pegawai internal Kantor Akuntan Publik XYZ. Peserta magang membuat 3 jenis *stored procedure create, read, update* saja karena untuk melakukan penghapusan data pada KAP XYZ menggunakan metode *soft delete* atau hanya penandaan bahwa data tersebut tidak akan di tampilkan pada saat melakukan “SELECT”, *stored procedure* digunakan untuk keperluan fitur *select, insert, update* dokumen, serta juga untuk keperluan *home page, admin page*, dan lain sebagainya. *Stored procedure* merupakan suatu blok program yang dapat dipanggil secara berulang-ulang dalam bentuk script, *stored procedure* ini akan

mempersingkat pemrograman karena programmer hanya perlu memanggil stored procedure dengan parameternya tanpa perlu mengetikkan *syntax* SQL yang Panjang, beberapa keuntungan dalam penggunaan *stored procedure* antara lain adalah pada sisi *performance*, *productivity* atau kemudahan dalam penggunaan, *scalability*, *maintainability*, *interoperability*, *replication*, dan *security*. Peserta magang menggunakan aplikasi RDBMS (*Relational Database Management System*) *Microsoft SQL Server* karena diantaranya pengendalian *database* secara terpusat, pembuatan *clustering* data, mempermudah pengelolaan *database*, *software* yang fleksible, terdapat fitur untuk *backup database*, fitur *recovery* dan *restore* data, dan juga karena *software* yang telah digunakan perusahaan [6].

Name	Schema	Create Date	Policy Health Status	Owner	Date Last Modified
System Stored Procedures					
sp_TM_ActionButton	dbo	7/25/2016 11:09 AM		dbo	11/22/2022 3:07 PM
sp_KB_TB_TaxConsultation_Delete	dbo	8/2/2022 3:02 PM		dbo	8/30/2022 3:14 PM
sp_KB_TB_TaxConsultation_Submit	dbo	8/2/2022 4:53 PM		dbo	8/30/2022 3:08 PM
sp_SP_TB_TaxConsultation_Update	dbo	8/2/2022 12:02 PM		dbo	8/29/2022 3:08 PM
sp_TM_MailQueue_SendMail	dbo	7/25/2016 11:09 AM		dbo	8/30/2022 2:17 PM
sp_TM_MailQueue_Insert	dbo	7/25/2016 11:09 AM		dbo	8/28/2022 4:17 PM
sp_KB_TB_TaxConsultation_Status_Published	dbo	8/2/2022 11:31 AM		dbo	8/28/2022 2:40 PM
sp_KB_TB_TaxConsultation_ForSearch	dbo	8/28/2022 2:44 PM		dbo	8/24/2022 1:46 PM
SP_PH_TaxPrecedent_ForSearch	dbo	4/12/2021 10:12 AM		dbo	8/24/2022 1:22 PM
sp_KB_TB_TaxConsultation_Home	dbo	8/22/2022 3:28 PM		dbo	8/24/2022 9:55 AM
SP_PH_TaxConsultation	dbo	8/22/2022 3:54 PM		dbo	8/23/2022 3:54 PM
sp_KB_TB_TaxConsultation_Select	dbo	8/2/2022 3:10 PM		dbo	8/23/2022 2:20 PM
SP_PH_TaxConsultation	dbo	8/22/2022 11:06 AM		dbo	8/23/2022 11:55 AM
sp_TB_HISTORY_Select	dbo	7/25/2016 11:09 AM		dbo	8/22/2022 10:00 AM
sp_PH_DocCase_L2	dbo	10/13/2021 5:02 PM		dbo	8/22/2022 8:52 AM
sp_KB_TB_TaxConsultation_Select_By_Partici...	dbo	8/5/2022 2:50 PM		dbo	8/8/2022 10:36 AM
sp_CP_RequestAccess_DraftSubmit	dbo	8/8/2021 3:28 PM		dbo	8/2/2022 5:08 PM

Gambar 3. 1 List stored procedure yang diubah atau dibuat

Pada gambar 3.1 merupakan list *stored procedure* yang diubah atau dibuat oleh peserta magang terdapat 13 *stored procedure* yang berfungsi untuk melakukan *create*, *update*, dan *select* data, berikut merupakan *stored procedure* yang dibuat oleh peserta magang dan penjelasannya.

Tabel 3. 2 Stored procedure yang dibuat dan penjelasannya

No	Nama <i>stored procedure</i>	Penjelasan
1	Sp_TM_ActionButton	Digunakan untuk menjadi <i>trigger</i> dalam mengirimkan email ke orang – orang yang telah di tentukan

No	Nama <i>stored procedure</i>	Penjelasan
2	Sp_KB_TB_TaxConsultation_Delete	Digunakan untuk melakukan <i>soft delete</i> dari tabel KB_TB_TaxConsultation kepada data yang telah dipilih melalui ID
3	Sp_KB_TB_TaxConsultation_Submit	Digunakan untuk memasukkan atau <i>insert</i> data ke dalam tabel KB_TB_TaxConsultation
4	Sp_KB_TB_TaxConsultation_Update	Digunakan untuk melakukan <i>update</i> data pada tabel KB_TB_TaxConsultation
5	sp_TM_MailQueue_SendMail	Digunakan untuk melakukan pengiriman <i>email</i> sesuai dengan data yang ada pada tabel TM_MailQueue
6	sp_TM_MailQueue_Insert	Digunakan untuk memasukkan atau <i>insert</i> data ke tabel TM_MailQueue
7	sp_KB_TB_TaxConsultation_Status_Published	Digunakan untuk meng- <i>update</i> atau melakukan penggantian status pada data di tabel KB_TB_TaxConsultation menjadi <i>published</i>
8	sp_KB_TB_TaxConsultation_ForSearch	Digunakan untuk fitur <i>search</i> atau pencarian dimana <i>stored procedure</i> ini akan melakukan <i>filter</i> sesuai dengan <i>keyword</i> yang dimasukkan oleh pengguna

No	Nama <i>stored procedure</i>	Penjelasan
9	sp_KB_TB_TaxConsultation_Home	<i>Stored procedure</i> ini akan digunakan di <i>home page</i> pada aplikasi web, dan data yang di tampilkan akan menyesuaikan siapa pengguna atau <i>user</i> nya
10	SP_PA_TaxConsultation	<i>Stored procedure</i> yang digunakan untuk menampilkan pada halaman <i>admin</i> karena pada <i>stored procedure</i> ini tidak terdapat <i>filter</i> pada data yang di tampilkan
11	sp_KB_TB_TaxConsultation_select	<i>Stored procedure</i> ini digunakan untuk menampilkan data pada tabel KB_TB_TaxConsultation berdasarkan nomor ID yang dipilih
12	SP_PH_TaxConsultation	<i>Stored procedure</i> ini merupakan <i>stored procedure</i> yang digunakan untuk halaman <i>home</i> dimana data yang ditampilkan akan di <i>filter</i> berdasarkan kata kunci yang dimasukan dan juga siapa pengguna pada saat itu
13	sp_KB_TB_TaxConsultation_select_by_Participant	Merupakan <i>stored procedure</i> yang akan menampilkan data dari tabel KB_TB_TaxConsultation, tetapi data yang ditampilkan

No	Nama <i>stored procedure</i>	Penjelasan
		akan di <i>filter</i> berdasarkan pengguna atau <i>user</i>

```

1 USE [TaxConsultation]
2 GO
3 /***** Object: StoredProcedure [dbo].[sp_KB_TB_TaxConsultation_Submit]    Script Date: 11/11/2022 3:31:33 PM *****/
4 SET ANSI_NULLS ON
5 GO
6 SET QUOTED_IDENTIFIER ON
7 GO
8
9
10
11
12
13
14
15
16 ALTER PROCEDURE [dbo].[sp_KB_TB_TaxConsultation_Submit]
17     @intTaxConsultationID INT,
18     @vcClientName VARCHAR(MAX),
19     @vcParticipants VARCHAR(MAX),
20     @vcTopic VARCHAR(100),
21     @vcBackground VARCHAR(MAX),
22     @vcClientTeamPosition VARCHAR(MAX),
23     @vcConclusion VARCHAR(MAX),
24     @vcIssues VARCHAR(MAX),
25     @vcLegalBasis VARCHAR(MAX),
26     @vcTypeofTax VARCHAR(MAX),
27     @vcStartDate SMALLDATETIME,
28     @vcCurrentStatus VARCHAR(50),
29     @intRefID INT OUTPUT
30 AS
31 BEGIN TRANSACTION;
32
33 BEGIN TRY
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
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

```

Gambar 3. 2 Stored Procedure SP_KB_TB_TaxConsultation_Submit

Gambar 3.2 merupakan salah satu contoh *stored procedure* yang peserta magang buat untuk kepentingan *submit* sebuah dokumen dimana *stored procedure* akan di gunakan pada API (*Application Programming Interface*) API sendiri merupakan mekanisme yang memungkinkan dua komponen perangkat lunak untuk saling berkomunikasi menggunakan serangkaian definisi dan protocol. *Stored procedure* ini memerlukan beberapa *input* diantaranya, @intTaxConsultationID sebagai ID, @vcClientName sebagai nama *client*, @vcParticipants sebagai participant, @vcTopic sebagai topik dari dokumen, @vcBackground

sebagai *background* dari dokumen yang nantinya akan diisi oleh *user*, @vcClientTeamPosition sebagai posisi dari *client* termasuk di tim yang telah di sediakan, @vcConclusion yang diisi dari kesimpulan dokumen tersebut sebelum nantinya akan di publikasikan, dan lain sebagainya. Terdapat 13 *parameter* yang digunakan agar *stored procedure* tersebut sukses dijalankan parameter – parameter tersebut nantinya akan di olah dan dimasukkan ke dalam tabel yang menampung data – data untuk nantinya ditampilkan oleh *stored procedure* yang berbeda.

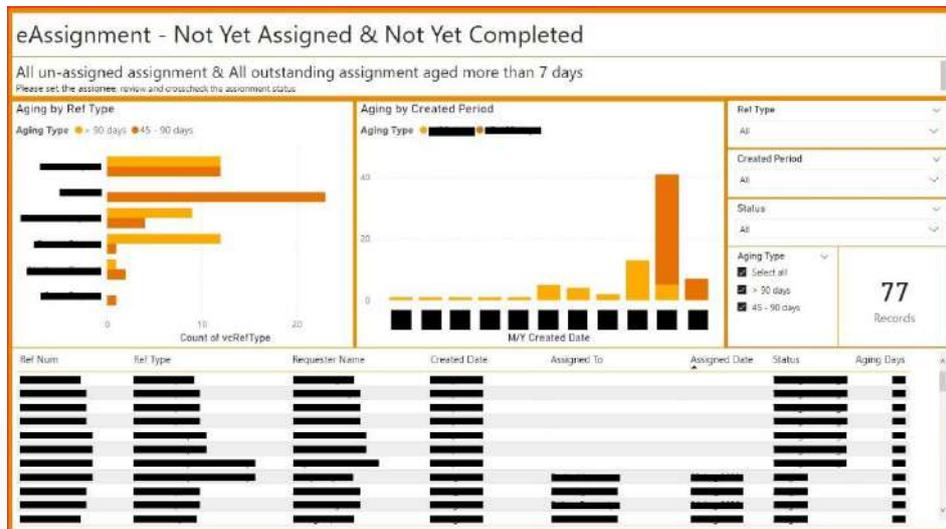
3.2.3 Membuat visualisasi report pada IM (Minggu 4-5)

Salah satu tugas yang diberikan oleh perusahaan adalah pembuatan *dashboard* yang bertujuan untuk memberikan *insight* seputar data yang ada menggunakan aplikasi *Microsoft Power Business Intelligence* atau yang biasanya disebut sebagai Microsoft Power BI, aplikasi Microsoft PowerBI digunakan karena KAP XYZ telah menggunakan aplikasi Microsoft Power BI terlebih dahulu. Microsoft Power BI adalah platform *business intelligence* (BI) yang menyediakan alat bagi pengguna bisnis nonteknis untuk menggabungkan, menganalisis, memvisualisasikan, dan berbagi data. Pada tugas ini data yang digunakan adalah data dari *hardware inventory*, pada gambar 3.3, 3.4, 3.5, 3.6, 3.7, 3.8 dibawah adalah contoh dari *dashboard* yang dibuat oleh peserta magang [7].



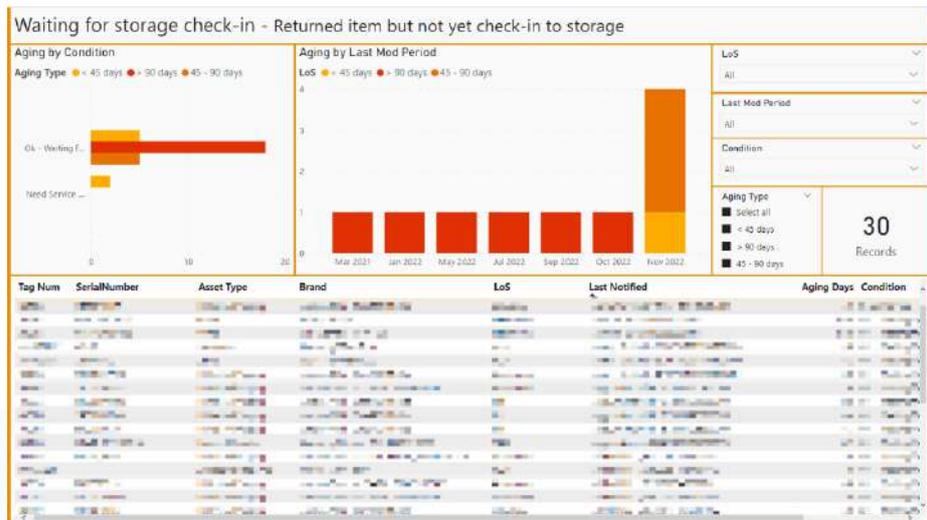
Gambar 3. 3 Dashboard inventory Logbook Report

Pada *dashboard* seperti mana terlihat pada gambar 3.3 merupakan salah satu tampilan *dashboard* yang didapat melalui *screenshot* pada aplikasi Microsoft PowerBI. Report ini bertujuan untuk menampilkan data dari *item* atau barang yang perlu untuk di tindak lanjuti. Laporan yang dibuat dari data *IM* ini menampilkan jumlah data berdasarkan periode dan tipe, juga terdapat data yang ditampilkan berbentuk tabel dengan *header* tabel berupa *In/Out*, *Request Type*, *Tag Num*, *Create Date*, *Item Name*, *Staff PIC*, *Storage PIC*, *Remarks*, *Aging Days*, dan *New Item*. Dan dapat dilihat pada *report* tersebut terdapat 4 data atau *records* dengan *aging type > 90 days*.



Gambar 3. 4 Dashboard inventory eAssignment

Pada *dashboard* yang terlihat pada gambar 3.4 yang juga didapat melalui hasil *screenshot* pada aplikasi Microsoft PowerBI. *Report* ini bertujuan untuk menampilkan *assignment* yang belum di *assign* atau belum di tetapkan dan juga *outstanding assignment* yang memiliki usia atau sejak tanggal di buat lebih dari 7 hari. Terdapat 77 data atau *records* dengan *aging type* > 90 days dan 45-90 days, juga terdapat 2 *bar charts* yang menunjukkan data *aging request* berdasarkan *Ref Type* dan juga tanggal pembuatan *request*.



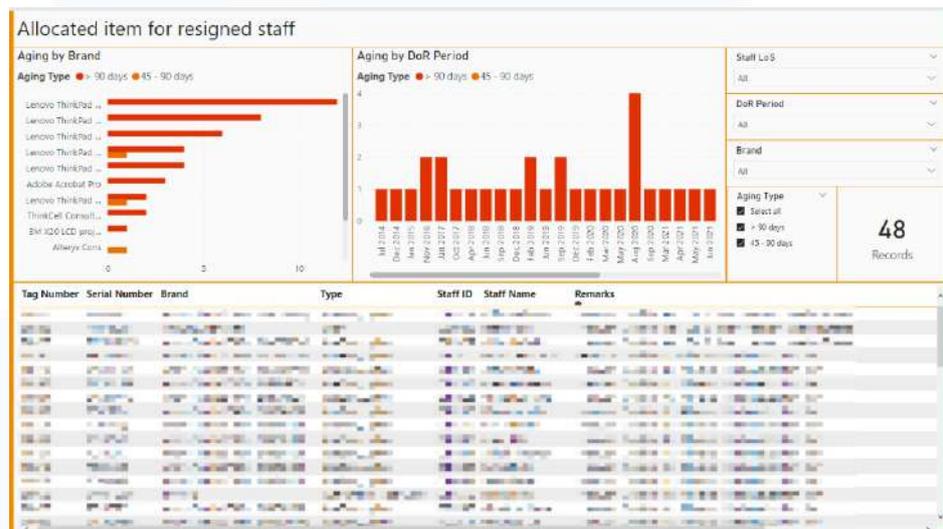
Gambar 3. 5 Dashboard inventory waiting for storage check-in

Pada *dashboard* yang terlihat pada gambar 3.5 yang juga didapat melalui hasil *screenshot* pada aplikasi Microsoft PowerBI, merupakan *dashboard* yang menampilkan *report* dari data *item* atau barang yang menunggu untuk dimasukkan ke tempat penyimpanan atau *item* yang sudah dikembalikan namun belum dilakukan *storage check-in*. Terdapat 30 data yang ada pada visual ini yang dapat dilihat melalui 3 cara yaitu melalui *bar chart Aging by Condition* dapat dilihat bahwa kondisi dari *item* yang dikembalikan di dominasi oleh barang dengan kondisi OK, *bar chart Aging by last mod period* dimana pada bulan November 2022 memiliki jumlah *return* atau pengembalian *item* ter-banyak yaitu sebanyak 4 *item* data – data yang ada pada *bar chart* akan dibagi menjadi 3 berdasarkan *aging type* atau berdasarkan berapa hari semenjak item telah di kembalikan, pembagian *aging type* sendiri dibagi menjadi 3 yaitu:

- < 45 days
- 45 – 90 days
- > 90 days

Data yang ada juga dapat dilihat melalui tabel yang terletak dibawah kedua *bar chart* dimana terdapat keterangan yang lebih lengkap yaitu:

- *Tag number* yang ber-isi nomor yang identik terhadap tiap *item* atau barang
- *Serial number* ber-isi nomor seri
- *Asset type* ber-isi tipe aset
- *Brand* ber-isi nama merek *item* atau barang tersebut
- *LoS* ber-isi nama divisi pemilik barang tersebut
- *Last modified* ber-isi tanggal terakhir data tersebut di lakukan perubahan
- *Aging days* ber-isi jumlah hari sejak *item* atau barang tersebut menunggu untuk melakukan *check-in*
- *Condition* ber-isi status kondisi dari barang *atau item* tersebut



Gambar 3. 6 Dashboard inventory allocated item for resigned staff

Pada gambar 3.6 merupakan *dashboard* dari *inventory outstanding allocated item for resigned staff* yang didapat melalui hasil *screenshot* pada aplikasi Microsoft PowerBI. Pada *dashboard* ini memiliki jumlah data sebanyak 48 data dimana terlihat dari 2 bar chart yaitu *aging by brand* dan *aging by Date of Resigned (DoR) period* yang keduanya *filter* oleh *aging type* yaitu *> 90 days* dan *45 – 90 days*. Pada bar chart

Aging by Brand dapat terlihat bahwa *item* yang paling banyak berasal dari Lenovo ThinkPad, dan pada bar chart *Aging by Date of Resigned (DoR) Period* dari period Juli 2014 sampai Juni 2021 terlihat cukup sama yaitu pada jumlah 1 akan tetapi terdapat beberapa periode yang memiliki perbedaan yaitu pada November 2016, Januari 2017, Februari 2019, September 2019 memiliki jumlah 2 per bulannya, dan pada Agustus 2020 terdapat 4 *items*, serta terdapat tabel yang ber-isi data – data:

- *Tag number* yang ber-isi nomor yang identik terhadap tiap item atau barang.
- *Serial number* ber-isi nomor seri.
- *Brand* ber-isi nama merek *item* atau barang tersebut.
- *Type* yang ber-isi tipe barang.
- *Staff ID* yang ber-isi nomor ID dari *staff*
- *Staff name* yang ber-isi nama dari *staff*
- *Remarks* yang ber-isi catatan yang sifatnya *optional*



Gambar 3. 7 Dashboard inventory outstanding spare request

Pada gambar 3.7 merupakan *dashboard inventory outstanding spare request* yang didapat melalui *screenshot* pada aplikasi Microsoft PowerBI, *dashboard inventory outstanding spare* merupakan visualisasi yang menunjukkan data *request* atau permintaan untuk *item* atau barang cadangan. Dimana pada *dashboard* ini terdapat jumlah data sebanyak 13 buah yang dapat dilihat melalui 2 bar chart dan juga 1 tabel, bar chart pertama yaitu *outstanding by status* dimana di bagi berdasarkan *aging type > 90 days* dan *aging type 45 – 90 days* dan dapat dilihat pada status *waiting for item return* terlihat memiliki jumlah yang jauh lebih banyak daripada *request* dengan status *waiting for verification*, pada bar chart *aging by create period* juga di bagi berdasarkan *aging type > 90 days* dan *aging type 45 – 90 days* dan dapat dilihat bahwa *request* ter-rendah terdapat pada periode Februari 2020, Juli 2020, Mei 2021, dan Oktober 2021 yang masing – masing hanya memiliki 1 *request* dan *request* terbanyak berada pada periode November 2022 dengan jumlah 5 *request* yang juga satu – satu nya data dengan *aging type 45 – 90 days* karena dari periode November 2022 sampai pada saat tanggal pengambilan *screenshot* pada aplikasi Microsoft PowerBI sudah lebih dari 45 hari tetapi belum lebih dari 90 hari, data lebih lengkap per *request* dapat dilihat pada tabel dibawah bar chart yang memiliki kolom *reservation number*, *SnowRefNumber*, *requester ID*, *Requester Name*, *Requester LoS*, *Request Date*, *Request PIC*, dan Status



Gambar 3. 8 Dashboard inventory waiting user collection

Pada gambar 3.8 merupakan *dashboard inventory outstanding waiting user collection* yang didapat dari hasil *screenshot* pada aplikasi Microsoft PowerBI, dimana visualisasi ini menjelaskan secara visual terhadap data dari *notebook* yang belum diambil oleh *new hire* atau *staff* yang baru di rekrut. Visualisasi ini memiliki jumlah data sebanyak 7 data dimana dapat terlihat dari 3 cara yaitu bar chart *aging by status* dimana data pada visual hanya memiliki 1 status yaitu *ready to collect*, bar chart *aging by Date of Hire (DOH)* yang juga hanya memiliki 2 periode *Date of Hire (DoH)* yaitu pada November 2022, dan tabel yang terletak dibawah dari kedua bar chart dengan kolom *reservation number*, *tag number*, *serial number*, *brand*, *status*, *staff ID*, *staff name*, *Los*, dan *date of hire*.

3.2.4 Mengubah stored procedure untuk Survey & Hot Review pada aplikasi Konsultasi Pertanyaan (Minggu 3-7)

Tugas yang diberikan kepada peserta magang kali ini dilakukan pada minggu ke 3 sampai minggu ke 7, peserta magang membuat *stored procedure* yang terkait dengan *survey* yang ada pada aplikasi web

Konsultasi Pertanyaan. Pada *enhancement* kali ini peserta magang melakukan penambahan 2 buah *stored procedure* yaitu *sp_PA_Survey_headerHotReview* dan *sp_PA_Survey*

```

1 USE [EnquiryConsultation]
2 GO
3 /***** Object: StoredProcedure [dbo].[sp_PA_Survey]    Script Date: 11/11/2022 3:39:21 PM *****/
4 SET ANSI_NULLS ON
5 GO
6 SET QUOTED_IDENTIFIER ON
7 GO
8
9
10 /*****
11 AUTHOR      : Philip
12 DATE       : 10 Aug 2022
13 PURPOSE    : vc key di isi survey/hotreview
14
15 --Updated by Philip on 1 Sep 2022
16 --select all reviews / hot reviews that have been / have not been submitted
17 *****/
18
19 ALTER PROC [dbo].[sp_PA_Survey]
20 @intDeptID INT,
21 @intFormSettingID INT,
22 @vcKey VARCHAR(50)
23 AS
24 BEGIN
25     DECLARE @vcDBname VARCHAR(50),
26             @intTemplateID INT,
27             @intAppID INT,
28             @vcLinkSurvey VARCHAR(500),
29             @colsUpivot AS NVARCHAR(MAX),
30             @query AS NVARCHAR(MAX),
31             @colsPivotSurvey AS NVARCHAR(MAX);
32
33     -----
34
35     SELECT @vcDBname = vcDBName_Survey
36     FROM dbo_EC_FormSetting
37     WHERE intFormSetting = @intFormSettingID;
38
39     -----
40
41     SELECT @intAppID = intAppID
42     FROM SurveyTemplate.dbo_TB_Application
43     WHERE vcHeader = @vcDBname
44           AND btActive = 1;
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

```

Gambar 3. 9 Stored procedure *sp_PA_Survey*

Pada gambar 3.9 diatas merupakan salah satu contoh *stored procedure* yang didapat melalui *screenshot* yang dilakukan pada aplikasi SQL Server Management Studio (SSMS). Dimana *stored procedure* *sp_PA_Survey* memiliki 3 parameter yaitu *@intDeptId* dengan tipe data *integer*, *@intFormsettingID* dengan tipe data *integer*, dan *@vcKey* dengan tipe data *varchar* sepanjang 50 karakter. Hasil atau *output* pada *stored procedure* ini akan menampilkan semua *review* atau *hot review* berdasarkan *@vcKey*, data nantinya ditampilkan merupakan data yang sudah di *submit* maupun yang belum di *submit stored procedure* tersebut

akan digunakan pada halaman *admin* pada aplikasi web *Konsultasi Pertanyaan*.

RequestID	Client Name	Friend	Staff Name	Send Date	Subnet Date	P002	P003	P004	P005	P006	P007	P008	P009	P010	P011	P012
1						NULL										
2						NULL										
3						NULL										
4						NULL										
5						NULL										
6						NULL										
7						NULL										
8						NULL										
9						NULL										
10						NULL										
11						NULL										
12						NULL										
13						NULL										
14						NULL										
15						NULL										
16						NULL										
17						NULL										

Gambar 3.10 Hasil dari Stored procedure *sp_PA_Survey*

Gambar 3.10 merupakan salah satu contoh hasil *stored procedure* yang didapat melalui *screenshot* yang dilakukan pada aplikasi SQL Server Management Studio (SSMS). *Screenshot* tersebut merupakan *output* dari *stored procedure* *sp_PA_Survey* tersebut yang nantinya akan digunakan oleh *programmer* untuk ditampilkan di *User Interface* (UI) atau tampilan web. Pada *project* ini peserta magang membuat ataupun meng-update beberapa *stored procedure* yang di gunakan untuk keperluan *create*, *read*, *update* saja karena untuk melakukan penghapusan data pada KAP XYZ menggunakan metode *soft delete* atau hanya penandaan bahwa data tersebut tidak akan di tampilkan pada saat melakukan “SELECT”.

```

9 /*****
10 AUTHOR : Philip
11 DATE : 18 Aug 2022
12 PURPOSE : used in grid admin
13 *****/
14
15 ALTER PROC [dbo].[sp_PA_Survey_headerHotReview]
16 @intDeptID INT,
17 @intFormSettingID INT
18 AS
19 BEGIN
20
21 DECLARE @vcDBname VARCHAR (50),
22 @intAppID int
23 SELECT @vcDBname = vcDBName_Survey
24 FROM EC_FormSetting
25 WHERE intFormSetting = @intFormSettingID;
26
27 -----
28
29 SELECT @intAppID = intAppID
30 FROM SurveyTemplate.dbo.TB_Application
31 WHERE vcHeader = @vcDBname
32 AND btActive = 1;
33
34 SELECT 'Client Name' AS 'field' ,
35 'Client Name' AS 'title'
36 UNION ALL
37 SELECT 'Period' AS 'field' ,
38 'Period' AS 'title'
39 UNION ALL
40 SELECT 'Staff Name' AS 'field' ,
41 'Staff Name' AS 'title'
42 UNION ALL
43 SELECT 'Send Date' AS 'field' ,
44 'Send Date' AS 'title'
45 UNION ALL
46 SELECT 'Submit Date' AS 'field' ,
47 'Submit Date' AS 'title'
48 UNION ALL
49 SELECT 'R'+LTRIM(STR(intSurveyQuestionID)) ,
50 vcQuestion
51 FROM SurveyTemplate.dbo.SURVEY_QUESTION
52 WHERE btActive = 1
53 AND intAppID = @intAppID;
54
55 END;
56

```

Gambar 3. 11 sp_PA_Survey_headerHotReview

Pada gambar 3.11 diatas merupakan salah satu contoh *stored procedure* yang didapat melalui *screenshot* yang dilakukan pada aplikasi SQL Server Management Studio (SSMS). Dimana *stored procedure* sp_PA_Survey_headerHotReview memiliki 2 parameter yaitu @intDeptId dengan tipe data integer, @intFormsettingID dengan tipe data integer. Hasil atau *output* pada *stored procedure* ini akan menampilkan *header* dari setiap *survey* yang ada digunakan pada halaman *admin* pada aplikasi web *Konsultasi Pertanyaan*

1 EXEC [sp_PA_Survey_headerHotReview] 0, 13

93 %

Results Messages

	field	title
1	Client Name	Client Name
2	Period	Period
3	Staff Name	Staff Name
4	Send Date	Send Date
5	Submit Date	Submit Date
6	R9022	1. [REDACTED]
7	R9023	2. [REDACTED]
8	R9024	3. [REDACTED]
9	R9025	4. [REDACTED]
10	R9026	5. [REDACTED]
11	R9027	6. [REDACTED]
12	R9028	7. [REDACTED]
13	R9029	8. [REDACTED]
14	R9030	9. [REDACTED]
15	R9031	10. [REDACTED]
16	R9032	11. [REDACTED]

Gambar 3.12 Output *sp_PA_Survey_headerHotReview*

Gambar 3.12 merupakan salah satu contoh hasil *stored procedure* yang didapat melalui *screenshot* yang dilakukan pada aplikasi SQL Server Management Studio (SSMS). *Screenshot* tersebut merupakan *output* dari *stored procedure* *sp_PA_Survey_headerHotReview* tersebut yang nantinya akan digunakan oleh *programmer* untuk ditampilkan di *User Interface* (UI) atau tampilan web pada halaman *admin*. Terdapat 32 *data* yang di bagi menjadi *field* dan *title*, dari 32 *data* tersebut terdapat 5 pertanyaan mengenai data dari pengguna pada saat mengisi *survey*, dan tanggal pengisian *survey*, dan terdapat 11 pertanyaan tentang *survey* yang dilakukan.

3.2.5 Mengubah dan menambahkan stored procedure untuk aplikasi Permintaan Akses dan Permintaan Pemulihan (ARS) (Minggu 6 – *Continue*)

Project ke-4 dimana peserta magang terlibat dalam proses pembuatan adalah ARS (Permintaan Akses dan Permintaan Pemulihan) dimana *project* ini merupakan *project* untuk melakukan *enhancement* atau perubahan yaitu pada bagian *flow* untuk melakukan persetujuan dimulai pada tanggal 6 September 2022, dimana terdapat penambahan *role* diantaranya, *additional approver* dan *security reviewer*. ARS (Permintaan Akses dan Permintaan Pemulihan) merupakan aplikasi berbasis web yang digunakan untuk meminta akses ataupun meminta untuk memulihkan akses seperti *Google drive*, email dan akses ke aplikasi web internal lainnya. Pada *enhancement* kali ini peserta magang membuat atau mengubah 12 *stored procedure* diantaranya

Tabel 3. 3 *Stored procedure* yang di ubah atau dibuat pada aplikasi web Permintaan Akses dan Permintaan Pemulihan

No	Nama <i>stored procedure</i>	Penjelasan
1	sp_AR_ApproverDelete	Digunakan untuk menghapus data <i>approver</i>
2	sp_AR_Access_SaveDraft	Digunakan untuk menyimpan data <i>access request</i> kedalam tabel AR_AccessRequest dengan status <i>draft</i>
3	sp_AR_Access_Submit	Digunakan untuk memasukkan data <i>access request</i> ke tabel AR_AccessRequest

No	Nama <i>stored procedure</i>	Penjelasan
		dengan status dokumen <i>submitted</i>
4	sp_AR_Restore_Submit	Digunakan untuk memasukkan data <i>restore request</i> ke dalam tabel AR_RestoreRequest dengan status <i>submitted</i>
5	sp_AR_Access_ReturnToRequester	Digunakan untuk mengubah atau meng- <i>update</i> status pada <i>access request</i> menjadi <i>returned to requester</i>
6	sp_AR_Restore_ReturnToRequester	Digunakan untuk mengubah atau meng- <i>update</i> status pada <i>restore request</i> menjadi <i>returned to requester</i>
7	sp_AR_SendToAssessor	Digunakan untuk mengirimkan <i>request</i> kepada <i>assessor</i> untuk pada <i>access request</i> dan <i>restore request</i>
8	sp_AR_SubmitAssessment	Digunakan oleh <i>assessor</i> yang sebelumnya telah menerima <i>request</i> untuk mengembalikan dokumen <i>request</i>

No	Nama <i>stored procedure</i>	Penjelasan
		dengan <i>assessment</i> didalamnya
9	sp_AR_Approver_Add_Additional	Digunakan untuk menambahkan data <i>additional approver</i> pada <i>access request</i> dan <i>restore request</i>
10	sp_AR_Update_SecurityIO	Digunakan untuk menambahkan data <i>security reviewer</i> sebagai <i>approver</i> dan <i>information owner</i> sebagai <i>approver</i> pada <i>access request</i> dan <i>restore request</i>
11	sp_AR_AdditionalApprover_Approve	Digunakan oleh <i>additional approver</i> untuk menyetujui <i>request</i>
12	sp_AR_SecurityIO_Approve	Digunakan oleh <i>security reviewer</i> dan <i>information owner</i> untuk menyetujui <i>request</i>

Tabel 3.3 merupakan 12 *stored procedure* yang dibuat ataupun di ubah oleh peserta magang pada aplikasi web Permintaan Akses dan Permintaan Pemulihan. Terdapat 5 *stored procedure* yang digunakan untuk melakukan *insert* data, 9 *stored procedure* yang dapat digunakan untuk melakukan *update* data, dan 1 *stored procedure* yang digunakan

untuk melakukan *soft delete* data. Berikut merupakan contoh dari ke-12 *stored procedure* yang telah di buat ataupun di ubah oleh peserta magang.

```

9  /*****
10 AUTHOR : Philip Budiono
11 DATE   : 1 Nov 2022
12 PURPOSE : Insert additional approver data to AR_Approver and update request STATUS to In approval progress (ARS/RES 013)
13 *****/
14 ALTER PROCEDURE [dbo].[sp_AR_Approver_Add_Additional]
15     @intARID int,
16     @chARType varchar(5), -- ACS / RES
17     @vcApproverName varchar(50),
18     @vcRemarks varchar(max),
19     @vcCurrentUser varchar(50),
20     @intRefId INT OUTPUT
21 AS
22 BEGIN TRANSACTION;
23
24 BEGIN TRY
25 -----
26
27     --History Parameter
28     DECLARE @chRefType CHAR(3),
29             @chStatusCodeOld CHAR(6),
30             @chStatusCode CHAR(6),
31             @vcHistory VARCHAR(500);
32
33     SET @vcHistory = 'Additional approver has been added by ' + @vcCurrentUser
34     SET @chRefType = @chARType
35     SET @chStatusCodeOld = ISNULL(dbo.fn_getCurrStatus(@chRefType,@intARID),'')
36
37
38     --update Status in AR_AccessRequest/AR_RestoreRequest
39     IF ( @chARType = 'ARS')
40     BEGIN
41         SET @chStatusCode = 'ARS013'
42
43         --update status in [AR_AccessRequest]
44         UPDATE AR_AccessRequest
45         SET
46             chStatus = @chStatusCode,
47             vcLastUpdateBy = @vcCurrentUser,
48             sdLastUpdateDate = GETDATE()
49         WHERE intAccessId = @intARID
50     END;
51
52     IF ( @chARType = 'RES')
53     BEGIN
54         SET @chStatusCode = 'RES013'
55
56         --update status in [AR_RestoreRequest]
57         UPDATE AR_RestoreRequest
58         SET
59             chStatus = @chStatusCode,
60             vcLastUpdateBy = @vcCurrentUser,
61             sdLastUpdateDate = GETDATE()
62         WHERE intRestoreId = @intARID
63
64     END;
65 -----

```

Gambar 3. 13 Stored procedure ARS sp_AR_Approver_Add_Additional 1

Gambar 3.13 merupakan *stored procedure* yang didapat melalui *screenshot* yang dilakukan pada aplikasi SQL Server Management Studio (SSMS). *Stored procedure* bernama sp_AR_Approver_Add_Additional digunakan untuk menambah *additional approver* untuk kepentingan *access* atau *restore*, sp_AR_Approver_Add_Additional memiliki 6 parameter dimana salah satu nya akan menjadi *output* yaitu:

- @intARID dengan tipe data *integer*

- @chARType dengan tipe data *varchar* yang dapat menampung 5 karakter
- @vcApproverName dengan tipe data *varchar* yang dapat menampung 50 karakter
- @vcRemarks dengan tipe data *varchar* yang dapat menampung karakter dengan jumlah yang tidak terhingga atau *MAX*
- @vcCurrentUser dengan tipe data *varchar* yang dapat menampung 50 karakter
- @intRefID dengan tipe data *integer* yang sekaligus berfungsi menjadi *output*

```

66
67 --add to AR_Approver
68 INSERT INTO AR_Approver
69 (
70     [intARID]
71     , [chARType]
72     , [chApproveType]
73     , [intOrder]
74     , [vcApproverName]
75     , [vcRemarks]
76     , [vcCreatedBy]
77     , [sdCreatedDate]
78     , [vcLastModBy]
79     , [sdLastModDate]
80     , [btActive]
81     , [btCurrent]
82 )
83 VALUES
84 (
85     @intARID,
86     @chARType,
87     'ADD',
88     1,
89     @vcApproverName,
90     @vcRemarks,
91     @vcCurrentUser,
92     GETDATE(),
93     @vcCurrentUser,
94     GETDATE(),
95     1,
96     1
97 )
98 SET @intRefId = SCOPE_IDENTITY();
99
100 -----
101
102 -- INSERT Hist
103 EXEC dbo.sp_TB_HISTORY_insert @chRefType, @intRefId, @chstatusCodeOld, @chstatusCode, @vcHistory, '', @vcCurrentUser;
104 -----
105
106 END TRY
107
108 BEGIN CATCH
109     IF @@TRANCOUNT > 0
110         ROLLBACK TRANSACTION;
111
112     -----
113     --Catch Error Details
114     DECLARE @vcErrProc VARCHAR(MAX),
115             @vcCurrProc VARCHAR(MAX);
116
117     SELECT @vcErrProc = ERROR_PROCEDURE(),
118            @vcCurrProc = OBJECT_NAME(@@PROCID);
119
120     EXEC dbo.sp_TB_LOG_ERROR_save @chRefType, @intRefId, @vcErrProc, @vcCurrProc, @vcCurrentUser
121 END CATCH
122

```

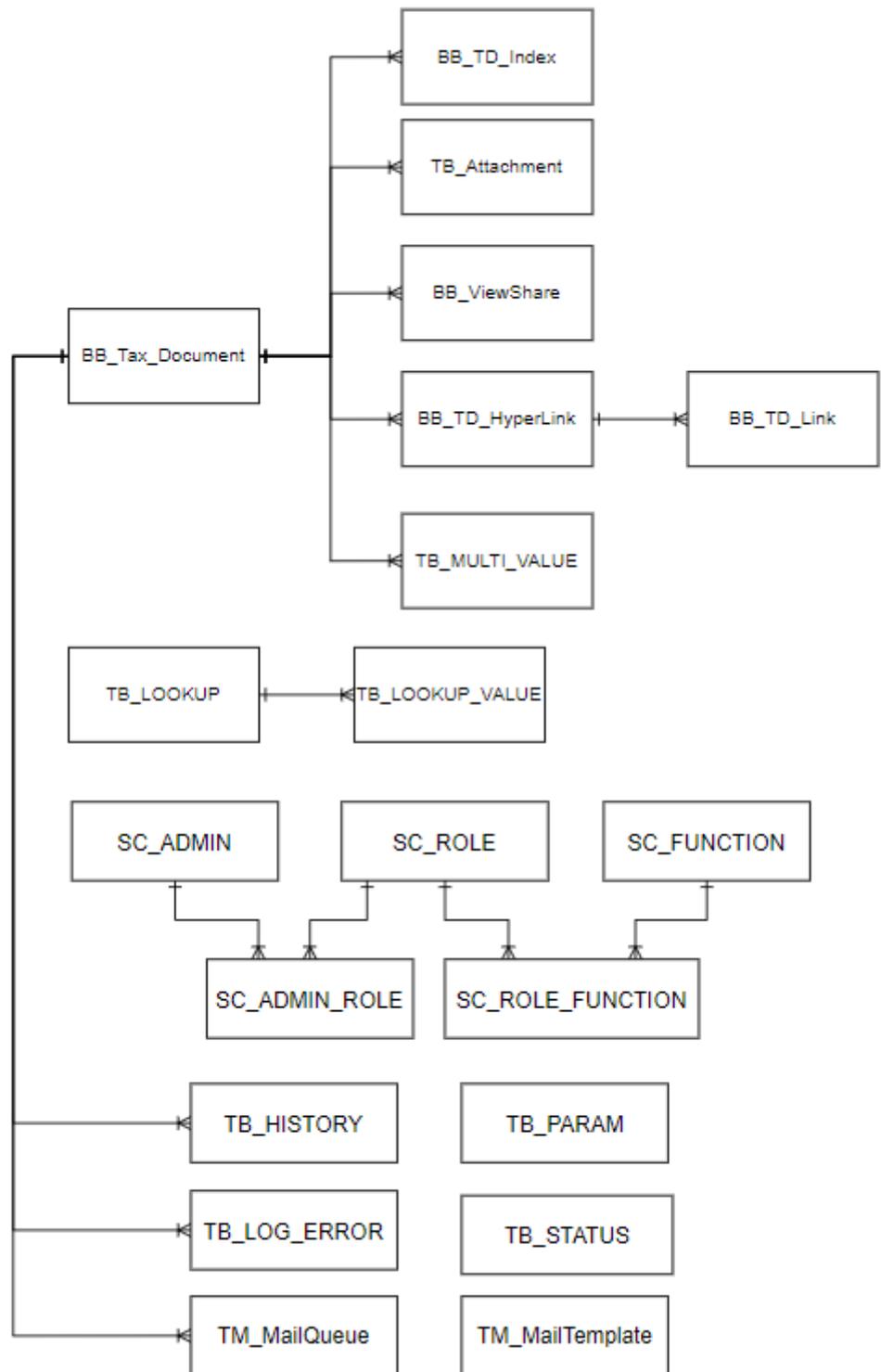
Gambar 3. 14 Stored procedure ARS *sp_AR_Approver_Add_Additional 2*

Gambar 3.14 merupakan *stored procedure* yang didapat melalui *screenshot* yang dilakukan pada aplikasi SQL Server Management Studio (SSMS). *Stored procedure* ini akan memasukkan data orang yang dipilih sebagai *additional approver* untuk kepentingan permintaan *access* atau *restore*. *Programmer* nantinya akan menjalankan atau meng – *execute* *stored procedure* ini sesuai dengan berapa nama yang akan dimasukkan.

Pada tanggal 21 Desember 2022 semua perubahan dan juga *requirement* telah dilakukan dan *project* ARS (Permintaan Akses dan Permintaan Pemulihan) pun masuk ke tahap selanjutnya yaitu *beta testing*, *beta testing* sendiri memiliki tujuan yaitu menemukan dan mengidentifikasi sebanyak mungkin bug atau masalah dalam sistem atau produk sehingga meminimalisir munculnya kesalahan pada saat tahap selanjutnya, pada tahap *beta testing* dilakukan oleh karyawan tetap yang memiliki peranan sebagai *quality assurance*. Akan tetapi sampai laporan magang MBKM ini selesai dibuat aplikasi web Permintaan Akses dan Permintaan Pemulihan belum melakukan *launching*, tetapi peserta magang masih melanjutkan magang di KAP XYZ.

3.2.6 Membuat *package*, *stored procedure* serta merancang struktur *database* untuk aplikasi Kamus Besar Perpajakan (Minggu 6 – *continue*)

Pada minggu ke-6 peserta magang membuat *stored procedure* yang digunakan untuk aplikasi web Kamus Besar Perpajakan, Kamus Besar Perpajakan merupakan aplikasi web yang ber-isi peraturan – peraturan pajak yang ada dan berlaku di Indonesia seperti, PPh, PPN, KUP, undang – undang, dan lain sebagainya. Aplikasi Kamus Besar Perpajakan nantinya tidak dapat di akses oleh khalayak umum akan tetapi hanya digunakan untuk *staff* dari KAP XYZ saja



Gambar 3. 15 Entity Relationship Diagram (ERD) Kamus Besar Perpajakan

Gambar 3.15 merupakan ERD (*Entity Relationship Diagram*) yang digunakan pada aplikasi web Kamus Besar Perpajakan dimana terdapat 20 tabel yang terdiri dari 10 tabel *master* yaitu sebagai berikut:

- TB_Lookup
- TB_Lookup_Value
- SC_Admin
- SC_Role
- SC_Admin_Role
- SC_Function
- SC_Role_Function
- TB_Param
- TB_Status
- TM_MailTemplate

Serta 10 tabel *transactional* yaitu:

- BB_Tax_Document
- BB_TD_Hyperlink
- BB_TD_Link
- BB_TD_Index
- TB_Attachment,
- BB_ViewShare
- TB_Multi_Value
- TB_History
- TB_Log_Error
- TM_MailQueue

Relasi yang terdapat antar tabel BB_Tax_Document dengan BB_TD_Index adalah *one to many* yang berarti 1 data pada BB_Tax_Document dapat memiliki satu atau banyak data pada tabel BB_TD_Index dimana tabel BB_TD_Index menyimpan data untuk index dari artikel yang nantinya digunakan oleh *programmer* untuk

mencari *index* dari artikel. Relasi yang ada antara BB_Tax_Document dan TB_Attachment adalah *one to many* dimana TB_Attachment digunakan untuk menyimpan *attachment* yang ada untuk artikel tersebut yang bisa berbentuk gambar ataupun file, relasi yang ada antara tabel BB_Tax_Document dan BB_ViewShare merupakan *one to many* dimana tabel BB_ViewShare mencatat apabila artikel dibuka atau dilihat oleh *user* atau pengguna, serta mencatat apabila artikel dibagikan dan nantinya data tersebut dapat digunakan untuk melihat jumlah *view* atau *share* yang dapat dihitung menggunakan fitur COUNT yang ada pada SQL. Relasi pada tabel BB_Tax_Document dengan BB_TD_Hyperlink juga merupakan relasi *one to many* dimana BB_TD_Hyperlink menyimpan kata yang telah di tandai oleh *user* atau pengguna pada saat memasukkan data artikel dan tabel BB_TD_Hyperlink juga memiliki relasi lain selain BB_Tax_Document yaitu tabel BB_TD_Link dimana BB_TD_Link menyimpan *link* referensi yang akan di masukkan oleh *user* dan 1 data BB_TD_Hyperlink dapat memiliki banyak *link* referensi yang akan di simpan di BB_TD_Link. Juga ada relasi antara tabel BB_Tax_Document dengan tabel TB_Multi_Value dengan relasi *one to many* dimana tabel TB_Multi_Value menyimpan data seperti kategori dari artikel yang dapat memiliki lebih dari 1 kategori. Juga terdapat relasi antara tabel TB_Lookup dan tabel TB_Lookup_Value dimana kedua tabel ini menyimpan data – data yang sekira-nya tidak memerlukan tabel tersendiri untuk menyimpannya, seperti kategori dan jenis peraturan dan nantinya data – data tersebut dapat di *read* atau di baca menggunakan *stored procedure* yang telah disiapkan. Tabel SC_Admin berfungsi untuk menyimpan data *user* atau pengguna seperti nama, divisi, dan data lainnya. Tabel SC_Role yang menyimpan data *role* dari pengguna seperti pengguna biasa atau *normal user*, admin, atau lainnya. SC_Function yang menyimpan data fungsi dari pengguna atau *user*. Ketiga tabel tersebut saling terhubung melalui 2 tabel yaitu SC_Admin_Role yang menghubungkan antara data di SC_Admin dan SC_Role, dan juga

SC_Role_Function yang menghubungkan data dari tabel SC_Role dan juga SC_Function. TB_History yang menyimpan data *history* dari ID pada data BB_Tax_Document seperti, dokumen telah dibuat oleh *user A*, dokumen telah di publikasi oleh *user B*, dokumen telah di hapus oleh *user A* yang memiliki hubungan one to many antara tabel BB_Tax_Document dan TB_History. TB_Log_Error yang memiliki hubungan *one to many* dengan BB_Tax_Document. Selain 17 tabel tersebut juga terdapat 3 tabel yang tidak memiliki relasi ke tabel lain yaitu:

- TB_Param yang menyimpan data *parameter* seperti *link* atau tautan ke aplikasi lain bila di butuhkan
- TB_Status yang menyimpan status apa saja yang berada di aplikasi web Kamus Besar Perpajakan
- TM_MailTemplate yang digunakan untuk menyimpan data *template* untuk melakukan email

```

19 ALTER PROC [dbo].[sp_List_Lookup_Desc] @vcLookup VARCHAR(50)
20 AS
21 BEGIN
22
23     SELECT DISTINCT
24         vcValue,vcDesc, intSequence
25     FROM   dbo.TB_LOOKUP_VALUE
26     WHERE  btActive = 1
27           AND intLookupID = ( SELECT intLookupID
28                             FROM   dbo.TB_LOOKUP
29                             WHERE  vcName = @vcLookup
30                             )
31     ORDER BY intSequence, vcValue
32
33 END;
34

```

Gambar 3. 16 Stored procedure *sp_List_Lookup_Desc*

Gambar 3.16 merupakan *stored procedure* *sp_List_Lookup_Desc* yang didapat melalui *screenshot* yang dilakukan pada aplikasi SQL Server Management Studio (SSMS). *Stored procedure* *sp_List_Lookup_Desc* memiliki 1 parameter yaitu @vcLookup untuk mem – *filter* data dari tabel TB_Lookup dan TB_Lookup_Value, *stored*

procedure `sp_List_Lookup_Desc` nantinya akan menampilkan 3 kolom data yaitu `vcValue`, `vcDesc`, dan `intSequence` yang digunakan untuk urutan penampilan data, data – data tersebut didapat dari `TB_Lookup_Value` yang menyimpan data – data seperti kategori artikel, dan jenis peraturan.

	Column Name	Data Type	Allow Nulls
▶	intLookupID	int	<input type="checkbox"/>
	vcName	varchar(50)	<input checked="" type="checkbox"/>
	vcDesc	varchar(500)	<input checked="" type="checkbox"/>
	btIsEditable	bit	<input checked="" type="checkbox"/>
	vcCreateBy	varchar(50)	<input checked="" type="checkbox"/>
	sdCreateDate	smalldatetime	<input checked="" type="checkbox"/>
	vcLastModBy	varchar(50)	<input checked="" type="checkbox"/>
	sdLastModDate	smalldatetime	<input checked="" type="checkbox"/>
	btActive	bit	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Gambar 3. 17 Desain tabel `TB_Lookup`

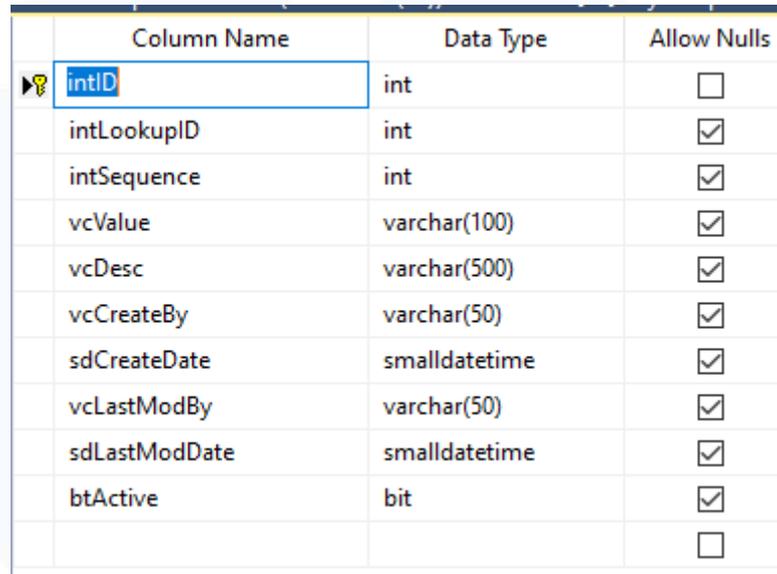
Gambar 3.17 merupakan desain tabel `TB_Lookup` yang didapat melalui *screenshot* yang dilakukan pada aplikasi SQL Server Management Studio (SSMS) tabel ini menyimpan data seperti berikut:

- **intLookupID** dengan tipe data *integer* diisi id data yang di *generate* secara otomatis apabila memasukkan data
- **vcName** dengan tipe data *varchar* yang menjadi pembeda antara data 1 dan data lain dengan contoh isian adalah ‘TRCategory’ untuk kategori, ‘TRRegulationType’ yang akan menyimpan data tipe peraturan atau regulasi.
- **vcDesc** dengan tipe data *varchar* yang ber-isi deskripsi dari data tersebut dimana kolom ini dapat diisi dengan data *string* kosong

- **btIsEditable** dengan tipe data *bit* atau yang biasanya dikenal dengan *Boolean* yang menentukan apakah data tersebut dapat di ubah
- **vcCreateBy** dengan tipe data *varchar* yang ber-isi dari nama pembuat data tersebut
- **sdCreateDate** dengan tipe data *smalldatetime* dengan *format* (YYY-MM-DD HH:MM:SS) yang menyimpan data kapan data tersebut dibuat
- **vcLastModBy** dengan tipe data *varchar* yang menyimpan data siapa *user* atau pengguna terakhir yang mengubah data tersebut
- **sdLastModDate** dengan tipe data *smalldatetime* dengan *format* (YYY-MM-DD HH:MM:SS) yang menyimpan data tanggal dan jam terakhir data tersebut diubah
- **btActive** dengan tipe data *bit* atau yang biasanya dikenal dengan *Boolean* yang menyimpan data tentang apakah data



tersebut aktif, dan biasanya digunakan untuk melakukan *soft delete*



Column Name	Data Type	Allow Nulls
intID	int	<input type="checkbox"/>
intLookupID	int	<input checked="" type="checkbox"/>
intSequence	int	<input checked="" type="checkbox"/>
vcValue	varchar(100)	<input checked="" type="checkbox"/>
vcDesc	varchar(500)	<input checked="" type="checkbox"/>
vcCreateBy	varchar(50)	<input checked="" type="checkbox"/>
sdCreateDate	smalldatetime	<input checked="" type="checkbox"/>
vcLastModBy	varchar(50)	<input checked="" type="checkbox"/>
sdLastModDate	smalldatetime	<input checked="" type="checkbox"/>
btActive	bit	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Gambar 3. 18 Desain tabel TB_LookupValue

Pada gambar 3.18 merupakan desain tabel dari TB_Lookup_Value yang didapat melalui *screenshot* yang dilakukan pada aplikasi SQL Server Management Studio (SSMS), sesuai dengan namanya tabel ini menyimpan *value* atau isi berdasarkan data pada tabel sebelumnya, dan pada tabel TB_Lookup_Value menyimpan 10 data yaitu:

- **intID** dengan tipe data *integer* yang menjadi ID pada tabel TB_Lookup_Value dan akan me-*generate* secara otomatis apabila memasukkan data agar tidak terdapat ID yang sama
- **intLookupID** dengan tipe data *integer* sebagai *foreign key* dari tabel TB_Lookup karena kedua tabel ini memiliki relasi yaitu *one to many* dimana 1 ID data pada tabel TB_Lookup dapat memiliki banyak data di tabel TB_Lookup_Value
- **intSequence** dengan tipe data *integer* yang berguna sebagai urutan pada saat data di tampilkan

- **vcValue** dengan tipe data *varchar* yaitu *value* dari tabel TB_Lookup, contohnya dapat di lihat pada gambar dibawah, pada intLookupID 1 yaitu TRRegulationStatus terdapat *value* ‘Reg Status 1’, ‘Reg Status 2’, ‘Reg Status 3’, ‘Reg Status 4’, ‘Reg Status 5’, ‘Reg Status 6’, dan ‘Reg Status 7’
- **vcDesc** dengan tipe data *varchar* yang ber-isi deskripsi dari data tersebut dimana kolom ini dapat diisi dengan data *string* kosong
- **vcCreateBy** dengan tipe data *varchar* yang ber-isi dari nama pembuat data tersebut
- **sdCreateDate** dengan tipe data *smalldatetime* dengan *format* (YYY-MM-DD HH:MM:SS) yang menyimpan data kapan data tersebut dibuat
- **vcLastModBy** dengan tipe data *varchar* yang menyimpan data siapa *user* atau pengguna terakhir yang mengubah data tersebut
- **sdLastModDate** dengan tipe data *smalldatetime* dengan *format* (YYY-MM-DD HH:MM:SS) yang menyimpan data tanggal dan jam terakhir data tersebut diubah
- **btActive** dengan tipe data *bit* atau yang biasanya dikenal dengan *Boolean* yang menyimpan data tentang apakah data tersebut aktif, dan biasanya digunakan untuk melakukan *soft delete*

intID	intLookupID	intSequence	vcValue	vcDesc	vcCreateBy	sdCreateDate	vcLastModBy	sdLastModDate	btActive
1	1	1	Reg Status 1		System	2022-09-19 00:00:00	System	2022-09-19 00:00:00	1
2	1	2	Reg Status 2		System	2022-09-19 00:00:00	System	2022-09-19 00:00:00	1
3	1	3	Reg Status 3		System	2022-09-19 00:00:00	System	2022-09-19 00:00:00	1
4	1	4	Reg Status 4		System	2022-09-19 00:00:00	System	2022-09-19 00:00:00	1
5	1	5	Reg Status 5		System	2022-09-19 00:00:00	System	2022-09-19 00:00:00	1
6	1	6	Reg Status 6		System	2022-09-19 00:00:00	System	2022-09-19 00:00:00	1
7	1	7	Reg Status 7		System	2022-09-19 00:00:00	System	2022-09-19 00:00:00	1
8	1	8	Reg Status 8		System	2022-09-19 00:00:00	System	2022-09-19 00:00:00	1
9	1	9	Reg Status 9		System	2022-09-19 00:00:00	System	2022-09-19 00:00:00	1

Gambar 3. 19 Contoh isian pada tabel TB_LookupValue

Gambar 3.19 merupakan contoh *value* atau isi dari TB_LookupValue yang didapat melalui *screenshot* yang dilakukan pada aplikasi SQL Server Management Studio (SSMS), dimana data yang ada

didalam tabel TB_LookupValue biasanya akan di lakukan *filter* berdasarkan intLookupID, dan ditampilkan berdasarkan urutan yang ada di kolom intSequence menggunakan salah satu fungsi *Structured Query Language* (SQL) “ORDER BY [nama kolom] ASCENDING”

Column Name	Data Type	Allow Nulls
intTaxDocumentID	int	<input type="checkbox"/>
vcRefNumber	varchar(50)	<input checked="" type="checkbox"/>
chStatusCode	char(5)	<input type="checkbox"/>
vcDocumentType	varchar(50)	<input type="checkbox"/>
vcTitle	varchar(500)	<input checked="" type="checkbox"/>
vcShortDesc	varchar(MAX)	<input checked="" type="checkbox"/>
vcContent	varchar(MAX)	<input checked="" type="checkbox"/>
vcNumber	varchar(500)	<input checked="" type="checkbox"/>
vcRegulationType	varchar(500)	<input checked="" type="checkbox"/>
vcSourceLink	varchar(500)	<input checked="" type="checkbox"/>
vcIssuedBy	varchar(500)	<input checked="" type="checkbox"/>
sdIssuedDate	smalldatetime	<input checked="" type="checkbox"/>
vcRegulationStatus	varchar(100)	<input checked="" type="checkbox"/>
sdEffectiveDate	smalldatetime	<input checked="" type="checkbox"/>
vcPublishedBy	varchar(50)	<input checked="" type="checkbox"/>
sdPublishedDate	smalldatetime	<input checked="" type="checkbox"/>
vcCreateBy	varchar(50)	<input type="checkbox"/>
sdCreateDate	smalldatetime	<input type="checkbox"/>
vcLastModBy	varchar(50)	<input type="checkbox"/>
sdLastModDate	smalldatetime	<input type="checkbox"/>
btActive	bit	<input type="checkbox"/>
		<input type="checkbox"/>

Gambar 3. 20 Design tabel BB_TaxDocument

Gambar 3.20 merupakan *design table* dari tabel BB_TaxDocument dimana tabel tersebut menyimpan 21 data sebagai berikut:

- **intTaxDocumentID** dengan tipe data *integer* sebagai penanda ID dokumen, field ini akan di *generate* secara otomatis sehingga tidak ada data ID yang sama
- **vcRefNumber** dengan tipe data *varchar* sebagai nomor referensi yang di *generate* secara otomatis
- **chStatusCode** sebagai status dokumen
- **vcDocumentType** dengan tipe data *varchar* sebagai tipe dokumen yang diisi '*Tax Regulation*' atau '*Internal*' atau '*External*' dan lain sebagainya
- **vcTitle** dengan tipe data *varchar* sebagai judul dari dokumen
- **vcShortDesc** dengan tipe data *varchar* sebagai deskripsi singkat dokumen yang dapat ditampilkan pada *home page*
- **vcContent** dengan tipe data *varchar* sebagai konten dari dokumen tersebut yang ber-isi *script HTML (Hypertext Markup Language)* untuk tampilan dari dokumen
- **vcNumber** dengan tipe data *varchar* yang digunakan untuk menyimpan nomor dokumen
- **vcRegulationType** dengan tipe data *varchar* yang menyimpan data tipe regulasi yang dimiliki oleh dokumen
- **vcSourceLink** dengan tipe data *varchar* yang menyimpan link asal informasi atau sumber informasi dari dokumen tersebut
- **vcIssuedBy** dengan tipe data *varchar* yang menyimpan data instansi pembuat dari dokumen sumber atau asal, salah satu contohnya adalah kementerian keuangan
- **sdIssuedDate** dengan tipe data *smalldatetime* dan *format (YYY-MM-DD HH:MM:SS)* yang menyimpan tanggal pembuatan dokumen oleh instansi

- **vcRegulationStatus** dengan tipe data *varchar* yang menyimpan status apakah dokumen tersebut masih berlaku pada saat ini
- **sdEffectiveDate** dengan tipe data *smalldatetime* dan *format* (YYY-MM-DD HH:MM:SS) yang menyimpan tanggal efektif dokumen atau peraturan tersebut berlaku,
- **vcPublishedBy** dengan tipe data *varchar* yang menyimpan data nama *user* atau pengguna yang mengesahkan atau *publish* dokumen tersebut sehingga dapat dilihat oleh karyawan *internal*
- **sdPublishedDate** dengan tipe data *smalldatetime* dan *format* (YYY-MM-DD HH:MM:SS) sebagai kolom untuk menyimpan kapan pengguna melakukan pengesahan dokumen
- **vcCreateBy** dengan tipe data *varchar* yang ber-isi dari nama pembuat data tersebut
- **sdCreateDate** dengan tipe data *smalldatetime* dengan *format* (YYY-MM-DD HH:MM:SS) yang menyimpan data kapan data tersebut dibuat
- **vcLastModBy** dengan tipe data *varchar* yang menyimpan data siapa *user* atau pengguna terakhir yang mengubah data tersebut
- **sdLastModDate** dengan tipe data *smalldatetime* dengan *format* (YYY-MM-DD HH:MM:SS) yang menyimpan data tanggal dan jam terakhir data tersebut diubah
- **btActive** dengan tipe data *bit* atau yang biasanya dikenal dengan *Boolean* yang menyimpan data tentang apakah data tersebut aktif, dan biasanya digunakan untuk melakukan *soft delete*

```

10 /*****
11 AUTHOR : Philip budiono
12 DATE : 15 Sep 2022
13 PURPOSE : Submit into table BB_TaxDocument
14 *****/
15
16 ALTER PROCEDURE [dbo].[sp_BB_Document_Submit]
17 @intTaxDocumentID INT,
18 @vcDocumentType VARCHAR(50) = '', -- Tax Treaty / Tax Regulation
19 @vcTitle VARCHAR(500) = '',
20 @vcShortDesc VARCHAR(MAX) = '',
21 @vcContent VARCHAR(MAX) = '',
22 @vcNumber VARCHAR(500) = '',
23 @vcRegulationType VARCHAR(500) = '',
24 @vcSourceLink VARCHAR(500) = '',
25 @vcIssuedBy VARCHAR(500) = '',
26 @sdIssuedDate SMALLDATETIME,
27 @vcRegulationStatus VARCHAR(100) = '',
28 @sdEffectiveDate SMALLDATETIME,
29 @vcListCategory VARCHAR(MAX), -- Delimiter '#'
30 @vcCurrentUser VARCHAR(50),
31 @intRefID INT OUTPUT
32 AS
33 BEGIN TRANSACTION;
34
35 BEGIN TRY
36
37 -----
38
39 -- Prepare history detail
40 DECLARE @chRefType CHAR(3),
41 @chStatusCode CHAR(5),
42 @chStatusCodeOld CHAR(5),
43 @vcHistory VARCHAR(MAX),
44 @vcChanges VARCHAR(MAX),
45 @chMailType CHAR(7);
46
47 SET @chRefType = 'TRD';
48 SET @chStatusCodeOld = ISNULL(dbo.fn_getCurrentStatus(@chRefType, @intTaxDocumentID), '');
49 SET @chStatusCode = 'TB000'; -- submitted/Draft
50 SET @chMailType = '';
51
52 -----
53
54 -- save to BB_TaxDocument
55 IF ISNULL(@intTaxDocumentID, 0) = 0
56 BEGIN
57
58 INSERT INTO dbo.BB_TaxDocument
59 (
60 chStatusCode,
61 vcDocumentType,
62 vcTitle,
63 vcShortDesc,
64 vcContent,
65 vcNumber,
66 vcRegulationType,
67 vcSourceLink

```

Gambar 3. 21 Stored procedure submit dokumen

Gambar 3.21 merupakan salah satu tangkapan layar atau *screenshot* yang didapat dari aplikasi SQL Server Management System, *stored procedure* ini dijalankan untuk memasukkan data kedalam tabel atau *database stored procedure* *sp_BB_Document_Submit* memiliki beberapa parameter diantaranya adalah sebagai berikut:

- @intTaxDocumentID sebagai ID dari dokumen
- @vcDocumentType sebagai pilihan antara dokumen tersebut termasuk dalam tipe *tax treaty* atau *tax regulation*

- @vcTitle sebagai judul dari dokumen tersebut, dan lain sebagainya.

Stored procedure ini juga digunakan dalam proses *update* data – data pada ID yang ada dimana jika ingin memasukkan data baru kedalam *database* dapat memasukkan parameter @intTaxDocumentID NULL atau tanpa data, tetapi jika ingin melakukan update data pada ID yang dipilih maka dapat memasukkan parameter @intTaxDocumentID dengan ID yang sesuai.

```

127
128
129
130 -- multi value save category
EXEC dbo.sp_TB_MULTI_VALUE_save @btIsReplace = 1,
131                                     @vcRefType = @chRefType,      -- varchar(10)
132                                     @intRefID = @intTaxDocumentID,  -- int
133                                     @vcAttribute = 'Category',      -- varchar(20)
134                                     @vcValueList = @vcListCategory, -- varchar(max)
135                                     @vcDelimiter = '#',            -- varchar(5)
136                                     @vcCurrentUser = @vcCurrentUser; -- varchar(50)
137
138 -----
139
140 -- delete on insert
141
142 -- Hyperlink
143 UPDATE dbo.BB_TD_Hyperlink
144 SET vcLastModBy = @vcCurrentUser,
145     sdLastModDate = GETDATE(),
146     btActive = 0
147 WHERE btActive = 1
148        AND intTaxDocumentID = @intTaxDocumentID;
149
150 -- Link
151 UPDATE dbo.BB_TD_Link
152 SET vcLastModBy = @vcCurrentUser,
153     sdLastModDate = GETDATE(),
154     btActive = 0
155 WHERE btActive = 1
156        AND intTaxDocumentID = @intTaxDocumentID;
157
158 -- Index
159 UPDATE dbo.BB_TD_Index
160 SET vcLastModBy = @vcCurrentUser,
161     sdLastModDate = GETDATE(),
162     btActive = 0
163 WHERE btActive = 1
164        AND intTaxDocumentID = @intTaxDocumentID;
165
166 -----
167
168
169 END TRY
170 BEGIN CATCH
171
172 IF @@TRANCOUNT > 0
173     ROLLBACK TRANSACTION;
174
175 -----
176 -- catch error details
177 DECLARE @vcErrProc VARCHAR(MAX),
178         @vcCurrProc VARCHAR(MAX);
179
180 SELECT @vcErrProc = ERROR_PROCEDURE(),
181        @vcCurrProc = OBJECT_NAME(@@PROCID);
182
183 EXEC dbo.sp_TB_LOG_ERROR_save @chRefType,
184                               @intTaxDocumentID

```

Gambar 3. 22 Stored procedure submit dokumen

Pada gambar 3.22 juga masih termasuk dalam *stored procedure* sp_BB_Document_Submit yang didapat dari tangkapan layar pada aplikasi SQL Server Management System, dimana pada baris ke 129 sampai ke baris 136 peserta magang memanggil *stored procedure* lain menggunakan fungsi EXECUTE untuk menjalankan *stored procedure* lain yaitu sp_TB_MULTI_VALUE_save dimana *stored procedure* ini memerlukan beberapa parameter untuk dijalankan yaitu @btIsReplace untuk memastikan apakah data ini akan me-replace data yang ada dengan @intRefID yang ditentukan, @vcRefType untuk menentukan tipe apakah dokumen tersebut, @intRefID untuk menentukan ID dokumen apa, @vcAttribute untuk menentukan hal yang disimpan merupakan data apa dan kali ini sudah ditentukan yaitu merupakan kategori dari dokumen, @vcValueList yang merupakan isi dari data yang nantinya akan disimpan, @vcDelimiter yang merupakan *delimiter* yang memisahkan data – data pada @vcValueList yaitu simbol #, dan yang terakhir adalah @vcCurrentUser yang merupakan *user* atau pengguna. Sp_TB_MULTI_VALUE_save akan menyimpan beberapa data dengan panduan ID yang telah ditentukan sebelumnya sehingga data yang ada dapat dicari dengan lebih mudah

Namun sampai pembuatan laporan magang MBKM *Track 2* ini selesai aplikasi web Kamus Besar Perpajakan masih belum dilakukan *launching* tetapi sudah dilakukan melewati tahap *beta testing* dimana pada tahap ini *staff* yang memiliki *job description* sebagai *quality assurance* melakukan *testing* atau pengujian terhadap aplikasi web Kamus Besar Perpajakan untuk menemukan *bug* ataupun *error*. Dan sampai pembuatan laporan magang MBKM *track 2* ini selesai sudah sampai ke tahap *User Acceptance Test* (UAT) dimana calon *user* akan melakukan pengujian dan menentukan apakah aplikasi sudah layak untuk digunakan atau di *launching*.

3.2.7 Membuat dan menambahkan *stored procedure* untuk aplikasi Aplikasi Perpajakan (Minggu 7 – continue)

Pada minggu ke – 7 peserta magang terlibat dalam aplikasi untuk divisi *Tax*, pada *project* ini dilakukan *re platform* dimana sebelumnya aplikasi ini hanya tersedia pada *desktop* saja, dan agar lebih mudah dalam *me-maintain* serta lebih *mobile*, aplikasi yang sebelumnya berbasis *desktop* ini dipindahkan menjadi berbasis *web*.

Name	Schema	Create Date	Policy Health State	Owner	Date Last Modified
System Stored Procedures					
sp_mek_FakturSelectByQueryForView	dbo	10/5/2022 8:41 AM		dbo	1/3/2023 11:56 AM
sp_mek_FakturInsert	dbo	8/9/2022 3:55 PM		dbo	12/13/2022 3:12 PM
sp_mek_FakturInsert	dbo	8/9/2022 7:47 PM		dbo	12/13/2022 10:49 AM
sp_mek_SignaturSelect	dbo	8/9/2022 10:48 AM		dbo	12/6/2022 7:39 PM
sp_mek_PDFInsert	dbo	8/9/2022 1:24 PM		dbo	12/7/2022 1:47 PM
sp_mek_KPPFakturSequencesInsert	dbo	8/9/2022 1:54 PM		dbo	12/6/2022 2:42 PM
sp_mek_VASOptimSelect	dbo	12/6/2022 2:36 PM		dbo	12/6/2022 2:36 PM
sp_mek_KMISet	dbo	12/1/2022 9:11 AM		dbo	12/1/2022 9:45 AM
sp_mek_EntryGrid	dbo	11/28/2022 9:17 AM		dbo	11/28/2022 11:48 AM
sp_mek_PDFSelect	dbo	11/28/2022 9:48 AM		dbo	11/28/2022 9:48 AM
sp_mek_PDFGrid	dbo	11/28/2022 9:29 AM		dbo	11/28/2022 9:29 AM
sp_mek_KPPFakturSequencesGrid	dbo	11/28/2022 9:52 AM		dbo	11/28/2022 9:52 AM
sp_mek_KPPFakturSequencesDelete	dbo	11/28/2022 9:47 AM		dbo	11/28/2022 9:47 AM
sp_mek_KPPFakturSequencesInsert	dbo	11/28/2022 9:48 AM		dbo	11/28/2022 9:48 AM
sp_mek_KPPFakturSequencesUpdate	dbo	8/9/2022 1:20 PM		dbo	11/28/2022 9:39 AM
sp_mek_EntrySelect	dbo	8/9/2022 1:13 PM		dbo	11/28/2022 9:30 AM
sp_mek_SignaturInsertSignature	dbo	11/29/2022 9:16 AM		dbo	11/29/2022 11:21 AM
sp_mek_CurrencyInsert	dbo	8/9/2022 3:53 PM		dbo	11/28/2022 9:50 AM
sp_mek_PDFDelete	dbo	8/9/2022 1:24 PM		dbo	11/22/2022 9:54 AM
sp_mek_PDFUpdate	dbo	8/9/2022 1:56 PM		dbo	11/23/2022 9:53 AM
sp_mek_ActiveCurrency	dbo	11/23/2022 9:56 AM		dbo	11/23/2022 9:56 AM
sp_mek_SelectCurrency	dbo	11/22/2022 9:34 AM		dbo	11/22/2022 9:34 AM
sp_mek_CurrencyInsert	dbo	8/9/2022 4:12 PM		dbo	11/23/2022 9:38 AM
sp_mek_CurrencyDelete	dbo	8/9/2022 4:09 PM		dbo	11/23/2022 9:29 AM
sp_mek_UpdateCurrency	dbo	8/9/2022 3:24 PM		dbo	11/23/2022 9:27 AM
sp_mek_SignaturDelete	dbo	8/9/2022 3:49 PM		dbo	11/23/2022 9:19 AM
sp_mek_SignaturUpdate	dbo	8/9/2022 3:19 PM		dbo	11/23/2022 9:07 AM
sp_mek_SignaturInsert	dbo	8/9/2022 3:04 PM		dbo	11/23/2022 8:59 AM
sp_mek_EntryUpdate	dbo	8/9/2022 4:16 PM		dbo	11/23/2022 8:55 AM
sp_mek_EntryDelete	dbo	8/9/2022 4:15 PM		dbo	11/23/2022 8:36 AM
sp_mek_PPHDataUpdateNormal	dbo	8/9/2022 2:48 PM		dbo	11/23/2022 8:33 AM
sp_mek_EntryInsert	dbo	8/9/2022 4:18 PM		dbo	11/22/2022 4:55 PM
sp_mek_PPHCalculateInsert	dbo	8/9/2022 2:27 PM		dbo	11/22/2022 1:44 AM
sp_mek_PPHDataUpdateChangeRate	dbo	8/9/2022 2:36 PM		dbo	11/22/2022 9:19 AM
sp_mek_PPHDataInsert	dbo	8/9/2022 2:09 PM		dbo	11/22/2022 8:48 AM
sp_mek_FakturSelectByPPHKey	dbo	11/9/2022 3:47 PM		dbo	11/15/2022 2:02 PM
sp_mek_PPHDataQueryForView	dbo	11/9/2022 1:57 PM		dbo	11/15/2022 8:21 AM

Gambar 3. 23 Stored procedure yang dibuat pada aplikasi web Aplikasi Perpajakan

Pada gambar 3.23 peserta magang membuat serta mengubah 32 *stored procedure* pada aplikasi Aplikasi Perpajakan yang memiliki fungsi untuk melakukan *create* data atau memasukkan data baru, *read* atau menampilkan data yang ada, *update* atau untuk mengubah data yang dipilih. *Stored procedure* yang dibuat oleh peserta magang adalah sebagai berikut.

Tabel 3. 4 *Stored procedure* yang di ubah atau di buat pada aplikasi web Aplikasi Perpajakan

No	Nama <i>stored procedure</i>	Penjelasan
1	sp_Web_FakturInsert	<i>Stored procedure</i> ini digunakan untuk memasukkan data ke tabel <i>tbIFakturPajak</i>

No	Nama <i>stored procedure</i>	Penjelasan
		dengan status sesuai dengan <i>input</i> dari <i>user</i> atau pengguna
2	sp_Web_FakturInsertDraft	<i>Stored procedure</i> ini digunakan untuk memasukkan data ke tabel tblFakturPajak dengan status TAP00 atau <i>draft</i>
3	sp_Web_FakturInsertCombine	<i>Stored procedure</i> ini digunakan untuk memasukkan data ke tabel tblFakturPajak dengan status mengikuti <i>input</i> dari <i>user</i> dan juga melakukan <i>input</i> data ke dalam tabel tblFakturPajakDtl
4	sp_Web_FakturInsertFinal	<i>Stored procedure</i> ini digunakan untuk memasukkan data ke tabel tblFakturPajak dengan status TAP10 atau <i>final</i>
5	sp_web_FakturSelectByQueryForView3	<p><i>Stored procedure</i> ini digunakan untuk melakukan pencarian dari tabel tblFakturPajak dimana <i>stored procedure</i> ini memiliki 10 parameter yaitu:</p> <ul style="list-style-type: none"> • @vcNPWP dengan tipe data <i>varchar</i> • @vcClientName dengan tipe data <i>varchar</i>

No	Nama <i>stored procedure</i>	Penjelasan
		<ul style="list-style-type: none"> • @vcRealInvoice dengan tipe data <i>varchar</i> • @vcEntityCode dengan tipe data <i>varchar</i> • @btOpenStatus dengan tipe data <i>integer</i> • @intRegardStatus dengan tipe data <i>integer</i> • @intReferenceNumber dengan tipe data <i>integer</i> • @intFakturType dengan tipe data <i>integer</i> • @intFakturStatus dengan tipe data <i>integer</i> • @intOldFaktur dengan tipe data <i>integer</i>
6	sp_Web_FakturUpdate	<i>Stored procedure</i> ini digunakan untuk melakukan update data pada tabel tblFakturPajak
7	sp_Web_FakturUpdateDraft	<i>Stored procedure</i> ini digunakan untuk melakukan

No	Nama <i>stored procedure</i>	Penjelasan
		update data pada tabel tblFakturPajak dimana status dari data yang di <i>update</i> menjadi <i>draft</i>
8	sp_Web_FakturUpdateFinal	<i>Stored procedure</i> ini digunakan untuk melakukan update data pada tabel tblFakturPajak dimana status dari data yang di <i>update</i> menjadi <i>final</i>
9	sp_Web_FakturDelete	<i>Stored procedure</i> ini digunakan untuk melakukan <i>soft delete</i> data pada tabel tblFakturPajak dimana btActive dari data yang di pilih akan diganti menjadi 0
10	sp_web_PKPInsert	<i>Stored procedure</i> ini digunakan untuk memasukkan data ke tabel tblPKP
11	sp_web_PKPSelect	<i>Stored procedure</i> ini digunakan untuk menampilkan data dari tabel tblPKP berdasarkan nomor ID yang dimasukkan
12	sp_web_PKPGrid	<i>Stored procedure</i> ini digunakan untuk menampilkan data dari tabel tblPKP berdasarkan kata kunci yang dimasukkan oleh user pada

No	Nama <i>stored procedure</i>	Penjelasan
		<p>@vcKey, yang nantinya kata kunci tersebut akan digunakan untuk melakukan <i>filter</i> pada data vcEntityCode atau vcPKPName</p>
13	sp_web_PKPUpdate	<p><i>Stored procedure</i> ini digunakan untuk melakukan penggantian atau <i>update</i> data dari tabel tblPKP berdasarkan nomor ID yang dimasukkan</p>
14	sp_web_PKPDelete	<p><i>Stored procedure</i> ini digunakan untuk melakukan <i>soft delete</i> data pada tabel tblPKP dimana btActive dari data yang di pilih akan diganti atau <i>update</i> menjadi 0</p>
15	sp_web_ClientInsert	<p><i>Stored procedure</i> ini digunakan untuk memasukkan data ke tabel tblClient</p>
16	sp_web_ClientSearch	<p><i>Stored procedure</i> ini digunakan untuk melakukan pencarian dari tabel tblClient dimana <i>stored procedure</i> ini memiliki 4 parameter yaitu:</p> <ul style="list-style-type: none"> • @vcClientNameKey dengan tipe data <i>varchar</i>

No	Nama <i>stored procedure</i>	Penjelasan
		<ul style="list-style-type: none"> • @vcNPWPKey dengan tipe data <i>varchar</i> • @intClientTypeKey dengan tipe data <i>integer</i> • @intActiveStatusKey dengan tipe data <i>integer</i>
17	sp_web_ClientUpdate	<p><i>Stored procedure</i> ini digunakan untuk melakukan penggantian atau <i>update</i> data dari tabel <i>tblClient</i> berdasarkan nomor ID yang dimasukkan</p>
18	sp_web_EntityInsert	<p><i>Stored procedure</i> ini digunakan untuk memasukkan data ke tabel <i>tblEntity</i></p>
19	sp_web_EntityGrid	<p><i>Stored procedure</i> ini digunakan untuk menampilkan data dari tabel <i>tblEntity</i> berdasarkan kata kunci yang dimasukkan oleh user pada @vcKey, yang nantinya kata kunci tersebut akan digunakan untuk melakukan <i>filter</i> pada data <i>vcEntityCode</i> atau <i>vcEntityDisplay</i> atau <i>vcEntityName</i></p>

No	Nama <i>stored procedure</i>	Penjelasan
21	sp_web_EntityUpdate	<i>Stored procedure</i> ini digunakan untuk melakukan penggantian atau <i>update</i> data dari tabel tblEntity berdasarkan nomor ID yang dimasukkan
22	sp_web_EntityDelete	<i>Stored procedure</i> ini digunakan untuk melakukan <i>soft delete</i> data pada tabel tblEntity dimana btActive dari data yang di pilih akan diganti atau <i>update</i> menjadi 0
23	sp_Web_KPPFakturSequenceInsert	<i>Stored procedure</i> ini digunakan untuk memasukkan data ke tabel tblKPPFakturSequence
24	sp_web_KPPFakturSequenceGrid	<i>Stored procedure</i> ini digunakan untuk menampilkan data dari tabel tblKPPFakturSequence berdasarkan kata kunci yang dimasukkan oleh user pada @vcKey, yang nantinya kata kunci tersebut akan digunakan untuk melakukan <i>filter</i> pada data vcEntityCode atau chYear
25	sp_web_KPPFakturSequenceSelect	<i>Stored procedure</i> ini digunakan untuk menampilkan data dari tabel

No	Nama <i>stored procedure</i>	Penjelasan
		tblKPPFakturSequence berdasarkan nomor ID yang dimasukkan
26	sp_web_KPPFakturSequenceUpdate	<i>Stored procedure</i> ini digunakan untuk melakukan penggantian atau <i>update</i> data dari tabel tblKPPFakturSequence berdasarkan nomor ID yang dimasukkan
27	sp_web_KPPFakturSequenceDelete	<i>Stored procedure</i> ini digunakan untuk melakukan <i>soft delete</i> data pada tabel tblKPPFakturSequence dimana btActive dari data yang di pilih akan diganti atau <i>update</i> menjadi 0
28	sp_web_PKPInsert	<i>Stored procedure</i> ini digunakan untuk memasukkan data ke tabel tblPKP
29	sp_web_PKPGrid	<i>Stored procedure</i> ini digunakan untuk menampilkan data dari tabel tblPKP berdasarkan kata kunci yang dimasukkan oleh user pada @vcKey, yang nantinya kata kunci tersebut akan digunakan untuk melakukan <i>filter</i> pada

No	Nama <i>stored procedure</i>	Penjelasan
		data vcEntityCode atau vcPKPName
30	sp_web_PKPSelect	<i>Stored procedure</i> ini digunakan untuk menampilkan data dari tabel tblPKP berdasarkan nomor ID yang dimasukkan
31	sp_web_PKPUpdate	<i>Stored procedure</i> ini digunakan untuk melakukan penggantian atau <i>update</i> data dari tabel tblPKP berdasarkan nomor ID yang dimasukkan
32	sp_web_PKPDelete	<i>Stored procedure</i> ini digunakan untuk melakukan <i>soft delete</i> data pada tabel tblPKP dimana btActive dari data yang di pilih akan diganti atau <i>update</i> menjadi 0

Tabel 3.4 merupakan 32 *stored procedure* yang dibuat atau diubah pada aplikasi web *Aplikasi Perpajakan*. Terdapat fungsi *create*, *read*, dan *update* saja karena untuk melakukan penghapusan data pada KAP XYZ menggunakan metode *soft delete* atau hanya penandaan bahwa data tersebut tidak akan di tampilkan pada saat melakukan “SELECT”. Dan berikut merupakan salah satu contoh dari *stored procedure* yang telah dibuat atau diubah oleh peserta magang

```

10 /*****
11 AUTHOR : Philip budiono
12 DATE : 8 Aug 2022
13 PURPOSE : Insert to table dbo.tblFakturPajak
14 *****/
15
16 ALTER PROCEDURE [dbo].[sp_web_FakturInsert]
17 @vcNPNP_1 [varchar](50),
18 @vcNPNPori_2 [varchar](50),
19 @intFakturType_3 [int],
20 @vcEntityCode_4 [varchar](20),
21 @intEntityNumber_5 [int],
22 @vcEntityName_6 [varchar](200),
23 @vcFakturNumber_7 [varchar](200),
24 @vcSignatureName_8 [varchar](200),
25 @vcSignatureLevel_9 [varchar](50),
26 @intReferenceNumber_10 [int],
27 @vcClientName_11 [varchar](200),
28 @vcAddress1_12 [varchar](200),
29 @vcAddress2_13 [varchar](200),
30 @vcVATStatus_14 [varchar](50),
31 @dtFakturDate_15 [datetime],
32 @vcCurrency_16 [varchar](50),
33 @dcAmount_17 [decimal] (18,3),
34 @dcMKRate_18 [decimal] (18,3),
35 @vcVATOption_19 [varchar](50),
36 @vcVATDesc_20 [varchar](200),
37 @vcPPN_21 [varchar](50),
38 @dcPPNPercentage_22 [decimal] (18,3),
39 @btFakturPaidStatus_23 [bit],
40 @btSSPstatus_24 [bit],
41 --@vcCreatedBy_25 [varchar](50),
42 --@dtCreatedDate_26 [datetime],
43 @btOldstatus_27 [bit],
44 @intRegardStatus_28 [int],
45 @vcRegardsDescription_29 [varchar](200),
46 @dcIDRAmount_30 [decimal] (18,3),
47 @dcDPPori_31 [decimal] (18,3),
48 @dcDPPIDR_32 [decimal] (18,3),
49 @dcPPNori_33 [decimal] (18,3),
50 @dcPPNIDR_34 [decimal] (18,3),
51 --@vcLastModifiedBy_35 [varchar](50),
52 --@dtLastModifiedDate_36 [datetime],
53 @vcLastPrintedBy_37 [varchar](50),
54 @dtLastPrintedDate_38 [datetime],
55 @intInvoicesType_39 [int],
56 @intFakturStatus_40 [int],
57 @vcRealInvoice_41 [varchar](50),
58 @vcRealVariant_42 [varchar](50),
59 @btOpenStatus_43 [bit],
60 @vcPKPName_44 [varchar](100),
61 @vcPKPAddress1_45 [varchar](50),
62 @vcPKPAddress2_46 [varchar](50),
63 @vcPKP_NPNP_47 [varchar](50),
64 @dtPKPDate_48 [datetime],
65 @ArrInvoice_49 [varchar] (3000),
66 @ArrIvcVariant_50 [varchar] (3000),
67 @intPrinted_51 [int]

```

Gambar 3. 24 Stored procedure sp_web_fakturInsert

```

226 | @chStatusCode)
227 |
228 | SET @intrefid = SCOPE_IDENTITY();
229 |
230 |
231 | DECLARE @vcVariant_2 [varchar](50),
232 | @vcInvoice_3 [varchar](50),
233 | @intFakturKey_1 [int]
234 |
235 | --@ArrInvoice_49 [varchar] (3000),
236 | --@ArrIvcVariant_50 [varchar] (3000)
237 |
238 | --get primary key of pph detail
239 | SET @intFakturKey_1= SCOPE_IDENTITY()
240 |
241 | --should have parsed the @ArrayFakturKey_2 [varchar 3000] here and loop
242 |
243 | DECLARE @Invoice varchar(50), @Variant varchar(50), @Pos1 int, @Pos2 int
244 |
245 | SET @ArrInvoice_49 = LTRIM(RTRIM(@ArrInvoice_49))+ ','
246 | SET @ArrIvcVariant_50 = LTRIM(RTRIM(@ArrIvcVariant_50))+ ','
247 | SET @Pos1 = CHARINDEX(',', @ArrInvoice_49, 1)
248 | SET @Pos2 = CHARINDEX(',', @ArrIvcVariant_50, 1)
249 |
250 | IF REPLACE(@ArrInvoice_49, ',', '') <> ''
251 | BEGIN
252 | WHILE (@Pos1 > 0 AND @Pos2 > 0)
253 | BEGIN
254 | PRINT 'Process Looping'
255 | SET @vcInvoice_3 = LTRIM(RTRIM(LEFT(@ArrInvoice_49, @Pos1 - 1)))
256 | SET @vcVariant_2 = LTRIM(RTRIM(LEFT(@ArrIvcVariant_50, @Pos2 - 1)))
257 | IF LTRIM(RTRIM(@vcVariant_2)) = '#'
258 | BEGIN
259 | SET @vcVariant_2 = NULL
260 | END
261 | IF @vcInvoice_3 <> ''
262 | BEGIN
263 | INSERT INTO [tbFakturPajakDt1]
264 | ([intFakturKey],
265 | [vcInvoice],
266 | [vcVariant])
267 |
268 | VALUES
269 | (@intFakturKey_1,
270 | @vcVariant_2,
271 | @vcInvoice_3)
272 | END
273 | SET @ArrInvoice_49 = RIGHT(@ArrInvoice_49, LEN(@ArrInvoice_49) - @Pos1)
274 | SET @ArrIvcVariant_50 = RIGHT(@ArrIvcVariant_50, LEN(@ArrIvcVariant_50) - @Pos2)
275 | SET @Pos1 = CHARINDEX(',', @ArrInvoice_49, 1)
276 | SET @Pos2 = CHARINDEX(',', @ArrIvcVariant_50, 1)
277 | END
278 | END
279 | --PRINT 'Process 2'
280 | ----after done looping
281 |
282 | DECLARE @intCountSeq [int]
283 | DECLARE @intFakturSeq [int]

```

Gambar 3. 25 stored procedure *sp_web_faktur_Insert* bagian looping

Gambar 3.24 dan 3.25 di atas merupakan contoh *stored procedure* yang tangkapan layar didapat dari aplikasi SQL Server Management System (SSMS). Dimana *stored procedure* ini digunakan untuk memasukkan data faktur ke dalam *database* yang memiliki cukup banyak *parameter* dengan jumlah 51 *parameter* untuk menjalankannya diantaranya:

1. @vcNPWP_1 dengan tipe data dengan tipe data *varchar*
2. @vcNPWPori_2 dengan tipe data dengan tipe data *varchar*
3. @intFakturType_3 dengan tipe data *integer*
4. @vcEntityCode_4 dengan tipe data dengan tipe data *varchar*
5. @intEntityNumber_5 dengan tipe data *integer*
6. @vcEntityName_6 dengan tipe data dengan tipe data *varchar*

7. @vcFakturNumber_7 dengan tipe data dengan tipe data *varchar*
8. @vcSignatureName_8 dengan tipe data dengan tipe data *varchar*
9. @vcSignatureLevel_9 dengan tipe data dengan tipe data *varchar*
10. @intReferenceNumber_10 dengan tipe data *integer*
11. @vcClientName_11 dengan tipe data dengan tipe data *varchar*
12. @vcAddress1_12 dengan tipe data dengan tipe data *varchar*
13. @vcAddress2_13 dengan tipe data dengan tipe data *varchar*
14. @vcVATStatus_14 dengan tipe data dengan tipe data *varchar*
15. @dtFakturDate_15 dengan tipe data tanggal *datetime*
16. @vcCurrency_16 dengan tipe data dengan tipe data *varchar*
17. @dcAmount_17 dengan tipe data *decimal*
18. @dcKMRate_18 dengan tipe data *decimal*
19. @vcVATOption_19 dengan tipe data dengan tipe data *varchar*
20. @vcVATDesc_20 dengan tipe data dengan tipe data *varchar*
21. @vcPPN_21 dengan tipe data dengan tipe data *varchar*
22. @dcPPNPercentage_22 dengan tipe data *decimal*
23. @btFakturPaidStatus_23 dengan tipe data *bit* atau *boolean*
24. @btSSPStatus_24 dengan tipe data *bit* atau *boolean*
25. @btOldStatus_27 dengan tipe data *bit* atau *boolean*
26. @intRegardStatus_28 dengan tipe data *integer*
27. @vcRegardsDescription_29 dengan tipe data dengan tipe data *varchar*
28. @dcIDRAmount_30 dengan tipe data *decimal*
29. @dcDPPOri_31 dengan tipe data *decimal*
30. @dcDPPIDR_32 dengan tipe data *decimal*
31. @dcPPNOri_33 dengan tipe data *decimal*
32. @dcPPNIDR_34 dengan tipe data *decimal*
33. @vcLastPrintedBy_37 dengan tipe data dengan tipe data *varchar*

34. @dtLastPrintedDate_38 dengan tipe data tanggal *datetime*
35. @intInvoicesType_39 dengan tipe data *integer*
36. @intFakturStatus_40 dengan tipe data *integer*
37. @vcRealInvoice_41 dengan tipe data dengan tipe data *varchar*
38. @vcRealVariant_42 dengan tipe data dengan tipe data *varchar*
39. @btOpenStatus_43 dengan tipe data *bit* atau *boolean*
40. @vcPKPName_44 dengan tipe data dengan tipe data *varchar*
41. @vcPKPAddress1_45 dengan tipe data dengan tipe data *varchar*
42. @vcPKPAddress2_46 dengan tipe data dengan tipe data *varchar*
43. @vcPKP_NPWP_47 dengan tipe data dengan tipe data *varchar*
44. @dtPKPDate_48 dengan tipe data tanggal *datetime*
45. @ArrInvoice_49 dengan tipe data dengan tipe data *varchar*
46. @ArrIvcVariant_50 dengan tipe data dengan tipe data *varchar*
47. @intIsPrinted_51 dengan tipe data *integer*
48. @intNPWPType_52 dengan tipe data *integer*
49. @vcBranchCode dengan tipe data *varchar*
50. @vcCurrentUser dengan tipe data *varchar*
51. @intRefId dengan tipe data *integer* yang sekaligus menjadi *output*

Namun sampai pembuatan laporan magang MBKM *Track 2* ini selesai aplikasi web Aplikasi Perpajakan masih belum dilakukan *launching* dan masih dalam tahap *development*, aplikasi web Aplikasi Perpajakan masih harus melewati tahap *beta testing* dimana pada tahap ini *staff* yang memiliki *job description* sebagai *quality assurance* melakukan *testing* atau pengujian terhadap aplikasi web Kamus Besar Perpajakan untuk menemukan *bug* ataupun *error*. Lalu juga harus melewati tahap *User Acceptance Test (UAT)* dimana calon *user* akan melakukan pengujian dan menentukan apakah aplikasi sudah layak untuk digunakan atau di *launching*, dan peserta magang masih melanjutkan magang di KAP XYZ.

3.2.8 Melakukan *update* data pada Enhancement NTPM (Nilai Transaksi Pasar Modal) (Minggu 10-12)

Pada minggu ke – 10 peserta magang mendapat tugas untuk melakukan perubahan data pada aplikasi NTPM (Nilai Transaksi Pasar Modal) yaitu pada pertanyaan – pertanyaan yang ada pada *section* tertentu, peserta magang juga melakukan penggantian data terhadap nomor – nomor yang ada. Namun sayangnya peserta magang tidak dapat menampilkan perubahan apa saja yang dilakukan karena data tersebut bersifat *confidential*. Dan perubahan perubahan tersebut dapat terselesaikan pada minggu ke – 12.

3.2.9 Membuat visualisasi report *stock card* pada IM (Minggu 10 – 12)

Pada minggu ke 10 – 12 peserta magang kembali mendapat tugas untuk membuat visualisasi dari data IM yang merupakan data untuk penyimpanan *hardware* yang nantinya akan digunakan untuk keperluan tim *Client Service* agar lebih mudah untuk melihat data dari inventory yang ada.

S&C			S&C 2			S&C 3		
Rack	Tag/Num	Total	Rack	Tag/Num	Total	Rack	Tag/Num	Total
EI_4E	105245	1	EI_4D	103960	1	EI_2D	104499	1
	105333	1		104247	1		104693	1
	105341	1		104327	1		104717	1
	105385	1		104373	1		105050	1
	105418	1		104427	1		300925	1
	105436	1		104447	1		Total	5
	105477	1		104449	1	EI_2C	103362	1
	105504	1		104466	1		104444	1
	105679	1		104492	1		104455	1
	106436	1		104510	1		104522	1
	400919	1		104535	1		104548	1
	Total	11		104559	1		104600	1
EI_4D	105517	1		104561	1		104601	1
	105290	1		104608	1		104736	1
	201451	1		104901	1		105508	1
	300470	1		105016	1		105022	1
	300475	1		105166	1		105045	1
	300476	1		106089	1		201062	1
	Total	6		201142	1		500117	1
EI_4C	104638	1		104608	1		500318	1
	104645	1		104901	1		Total	14
	104653	1	EI_4C	104715	1	EI_2B	104443	1
	104656	1		201051	1		104560	1
	104660	1		201059	1		104682	1
	104677	1		201070	1		104827	1
	104709	1		201197	1		104915	1
	104723	1		201210	1		201143	1
	104845	1		201305	1		201194	1
	104862	1		201318	1		201304	1
	104884	1		300928	1		300948	1
	104923	1		300952	1		300963	1

Gambar 3. 26 Stock card inventory lemari S&C

Gambar diatas didapat melalui *screenshot* dari aplikasi Microsoft powerBI. Pada gambar diatasterdapat 3 tabel yang terdiri dari S&C, S&C 2, dan S&C 3 dimana setiap tabel memiliki 3 kolom yaitu rack, tagNum, dan total. Tabel – tabel ini nantinya akan digunakan untuk melihat untuk *item* atau barang dengan tagnum sekian berada pada *rack* mana, dan ada berapa *unit*, serta pada tabel dapat melihat total item yang berada pada setiap *rack*.

Available 1	Available 2	Available 3	Available 4	Available 5	Available 6	Available 7	Available 8	
Rack	TagNum	Total	Rack	TagNum	Total	Rack	TagNum	Total
2D 104255	4E 300931	1	4A 106366	4A 300299	1	1A 106639	3C 107586	1
104298	300956	1	106388	300430	1	306852	2A 201732	1
104490	300958	1	106380	Total 2	106940	3B 107451	1	
104532	300960	1	106389	3C 301019	1	106942	201785	1
104543	300964	1	106609	301036	1	106946	Total 3	3
104573	300871	1	106623	301048	1	107091	1A 402073	1
104617	300974	1	106632	301181	1	107201	4A 301303	1
104647	300979	1	106871	Total 4	107219	1	402075	1
104685	300984	1	106872	3A 201511	1	107239	402080	1
104702	300989	1	106877	Total 1	107448	1	402081	1
104704	301000	1	106879	2B 105541	1	107519	402082	1
104919	301133	1	106888	105639	1	107651	402086	1
104955	Total 12	1	106890	Total 2	107822	1	402087	1
104976	4D 300926	1	106899	1B 105753	1	107897	402088	1
105134	300930	1	Total 14	105319	1	Total 14	402091	1
105138	300932	1	3A 201329	105909	1	3A 201727	402093	1
Total 16	300935	1	201331	105995	1	201826	Total 12	1
3C 103959	300941	1	201349	Total 6	105999	2	Total 15	1
103967	300942	1	201353	1A 105752	1	4A 201334	3A 107984	1
104413	300946	1	201359	105768	1	Total 1	107985	1
104418	300947	1	201371	105923	1	4C 300513	107988	1
104461	300950	1	201378	105936	1	Total 1	108019	1
104726	300951	1	201397	105968	1	Total 18	108023	1
104815	300961	1	201408	105989	1	108024	1	
104837	300972	1	201432	105990	1	108025	1	
104986	300980	1	201437	106013	1	108026	1	
105002	300988	1	Total 11	106014	1	108027	1	
105020	300999	1	2D 400747	106061	1	108028	1	
105021	301001	1	Total 8	106072	1	108029	1	
105038	301003	1	2B 105238	106116	1	108030	1	
105031	301118	1	105246	106352	1	108031	1	

Gambar 3. 27 Stock card inventory lemari available

Pada Gambar diatas yang didapat melalui *screenshot* dari aplikasi Microsoft powerBI merupakan lemari available, dapat dilihat jumlah lemari berbeda dengan jumlah 8 lemari available yang terdiri dari available 1, available 2, available 3, available 4, available 5, available 6, available 7, dan available 8. Setiap tabel memiliki 3 kolom yaitu rack, tagNum, dan total. Tabel – tabel ini nantinya akan digunakan untuk melihat untuk *item* atau barang dengan tagnum sekian berada pada *rack* mana, dan ada berapa *unit*, serta pada tabel dapat melihat total item yang berada pada setiap *rack*.

Disposal 1			Disposal 2			Disposal 3			Disposal 4			Disposal Box			
Rack	TagNum	Total	Rack	TagNum	Total	Rack	TagNum	Total	Rack	TagNum	Total	Rack	TagNum	Total	
4D	103145	1	4D	102737	1	2D	103260	1	4D	103881	1		102722	1	
	103331	1		102782	1		103278	1		103893	1		102727	1	
	103352	1		103280	1		103284	1		104011	1		102728	1	
	103359	1		103362	1		103312	1		104018	1		102732	1	
	103424	1		103619	1		103234	1		105041	1		102734	1	
	103413	1		104069	1		103336	1		104116	1		102742	1	
	103454	1		300671	1		103342	1		104117	1		102746	1	
	103458	1		300674	1		103399	1		104134	1		102755	1	
	103470	1		300695	1		103422	1		104168	1		102756	1	
	103472	1		300698	1		103426	1		104200	1		102759	1	
	102761	1		300870	1		102770	1		106083	1		102760	1	
	102758	1		300882	1		105801	1		200984	1		102761	1	
	103774	1		400523	1		Total	12		201058	1		102764	1	
	103906	1		400559	1		103287	1		300869	1		102765	1	
	102815	1		400804	1		103289	1		300912	1		102766	1	
	103829	1		500095	1		103305	1		300915	1		102770	1	
	Total	16		500171	1		103343	1		400846	1		102771	1	
4C	103363	1		Total	17		103376	1		400850	1		102772	1	
	103387	1		4C	103206	1		103388	1		500525	1		102773	1
	103433	1		201015	1		103406	1		Total	19		102775	1	
	200704	1		300676	1		103427	1		4C	103384	1		102776	1
	200760	1		300694	1		103771	1		103887	1		102779	1	
	200794	1		300896	1		103820	1		103889	1		102781	1	
	200848	1		500598	1		103850	1		103896	1		102786	1	
	200929	1		Total	6		105743	1		103912	1		102792	1	
	200962	1		4B	100800	1		Total	12		104044	1		102794	1
	300517	1		101369	1		2B	103321	1		104045	1		102796	1
	300733	1		103746	1		103356	1		104053	1		102797	1	
	300865	1		103866	1		103361	1		104073	1		102798	1	
	300886	1		103878	1		103366	1		104091	1		102799	1	
	300897	1		104128	1		103368	1		104121	1		102801	1	

Gambar 3. 28 Stock card inventory lemari disposal

Pada Gambar 3.28 yang didapat melalui *screenshot* dari aplikasi Microsoft powerBI merupakan dashboard lemari disposal, dimana terdapat 5 lemari yang terdiri dari disposal 1, disposal 2, disposal 3, disposal 4, dan disposal box. Setiap tabel memiliki 3 kolom yaitu rack, tagNum, dan total. Tabel – tabel ini nantinya akan digunakan untuk melihat untuk *item* atau barang dengan tagnum sekian berada pada *rack* mana, dan ada berapa *unit*, serta pada tabel dapat melihat total item yang berada pada setiap *rack*.

3.2.10 Membuat dan menambahkan *stored procedure* pada aplikasi *Morning News* (Minggu 16 – continue)

Tugas atau *project* terakhir yang dilakukan dimana peserta magang terlibat langsung adalah aplikasi berbasis web *Morning News* dimana aplikasi ini merupakan aplikasi berbasis web yang menyediakan berita – berita ataupun artikel terkait dengan topik – topik tertentu seperti *financial services, energy, utilities, and resources, consumer industrial product and services* dan lain sebagainya yang dapat digunakan untuk karyawan dari

perusahaan sebagai sumber menambah wawasan dengan sumber yang dapat di percaya.

```
73
74
75 IF(@intArticleID = 0)
76 BEGIN
77     --Insert new data
78     SET @vcHistory = 'Article has been submitted by ' + @vcCreatedBy
79     --Insert to S_Article
80     INSERT INTO S_ARTICLE
81     (
82         intCategoryID ,
83         intSourceID ,
84         dtSource ,
85         intStatusID ,
86         intTypeID ,
87         vcArtTitle ,
88         vcArtTitleEnglish,
89         vcArtKeyword ,
90         vcArtHighlight ,
91         btArtLocalGlobalFlag ,
92         btArtShareableToExtFlag ,
93         vcRemarks ,
94         vcCreatedBy ,
95         dtCreated ,
96         vcLastMod ,
97         dtLastMod ,
98         btActive
99     )
100     VALUES (
101         @intCategoryID ,
102         @intSourceID ,
103         @dtSource , -- convert(datetime,@dtSource),
104         @intStatusCode ,
105         @intTypeID ,
106         @vcArtTitle ,
107         @vcArtTitleEnglish,
108         @vcArtKeyword ,
109         @vcArtHighlight ,
110         @btArtLocalGlobalFlag ,
111         @btArtShareableToExtFlag ,
112         @vcRemarks ,
113         @vcCreatedBy ,
114         GETDATE() ,
115         @vcCreatedBy ,
116         GETDATE() ,
117         1
118     )
119
120 SET @intArticleID = SCOPE_IDENTITY();
121 SET @intRefID = @intArticleID
122
123 --insert Article Content
124 EXEC sp_S_ARTICLE_CONTENT_insert @intArticleID, @vbArtContent, @vcCreatedBy
125 --insert to Article_Lock if status = DRAFT
126 IF (@intStatusCode = 1)
127 BEGIN
128     EXEC [sp_S_Lock_Insert_V2] @intLockID = @intArticleID, @vcCurrentUser = @vcCreatedBy
129 END;
130
131 --Looping for TAG-----
132 DECLARE @intCounter INT = 1,
133         @intDelimiter INT,
134         @intStart INT =0,
135         @intEnd INT,
```

Gambar 3. 29 Stored procedure Morning News

Pada aplikasi ini peserta magang membuat *stored procedure* baru dengan standar perusahaan dimana terdapat *history* serta *log error* untuk memantau pergerakan dari data yang ada. Pada gambar 3.29 merupakan salah satu contoh dari *stored procedure* yang digunakan untuk memasukkan

artikel yang ada kedalam *database* dimana jika @intArticleID diisi 0 akan membuat data baru dengan ID baru, dan juga *stored procedure* ini akan menjalankan *stored procedure* lain yaitu sp_S_Article_Content_Insert yang digunakan untuk memasukkan konten artikel dengan tipe data *varbinary*, tipe data *varbinary* sendiri digunakan untuk menyimpan data biner dengan panjang maksimum yang ditentukan (dalam byte).

Namun sampai pembuatan laporan magang MBKM *Track 2* ini selesai aplikasi web Kamus Besar Perpajakan masih belum dilakukan *launching* dan masih dalam tahap *User Acceptance Test (UAT)* dimana calon *user* akan melakukan pengujian dan menentukan apakah aplikasi sudah layak untuk digunakan atau di *launching*, dan peserta magang masih melanjutkan magang di KAP XYZ

3.3 Kendala yang Ditemukan

- Terdapat beberapa informasi terkait aplikasi yang digunakan seperti salah satunya Power BI yang tidak didapat selama kegiatan belajar di universitas, sehingga peserta magang harus mencari solusi atas masalah yang di hadapi.
- Terdapat bahasa pemrograman yang tidak didapat selama proses belajar di universitas seperti DAX (*Data Analysis Expressions*), sehingga peserta magang mendapati kendala untuk menyelesaikan tugas yang diberikan dan harus mencari solusi atas masalah yang di hadapi.
- Terdapat bahasa pemrograman yang tidak didapat secara mendalam selama proses belajar di universitas seperti SQL (*Structured Query Language*), sehingga peserta magang mendapati kendala untuk menyelesaikan tugas yang diberikan dan harus mencari solusi atas masalah yang di hadapi.

3.4 Solusi atas Kendala yang Ditemukan

- Mencari solusi untuk masalah yang di hadapi melalui internet ataupun bertanya kepada senior, pembimbing, dan supervisi magang. Supaya masalah yang dihadapi dalam kegiatan magang dapat terselesaikan dengan baik.

- Mencari solusi yang di hadapi melalui internet ataupun bertanya kepada senior, dan supervisi magang seputar bahasa pemrograman DAX. Supaya masalah yang dihadapi dalam kegiatan magang dapat terselesaikan dengan baik.
- Mencari solusi yang di hadapi melalui internet ataupun bertanya kepada senior, dan supervisi magang seputar bahasa pemrograman *Structured Query Language* (SQL). Supaya masalah yang dihadapi dalam kegiatan magang dapat terselesaikan dengan baik.

