

CHAPTER V

IMPLEMENTATION AND TESTING

5. 1 Implementation

This application has five steps that must be done, for the first step you. Needs to input the file first. The input file here only serves to storing card data from kartu.txt.

5.1.1 Step 1 - Storing Card Data From kartu.txt.

```
1 As; Spade;1
2 2; Spade;2
3 3; Spade;3
4 4; Spade;4
5 5; Spade;5
6 6; Spade;6
7 7; Spade;7
8 8; Spade;8
9 9; Spade;9
10 10; Spade;10
11 J; Spade;10
12 Q; Spade;10
13 K; Spade;10
14 As; Heart;1
15 2; Heart;2
16 3; Heart;3
17 4; Heart;4
18 5; Heart;5
19 6; Heart;6
20 7; Heart;7
21 8; Heart;8
22 9; Heart;9
23 10; Heart;10
24 J; Heart;10
25 Q; Heart;10
26 K; Heart;10
27 As; Diamond;1
28 2; Diamond;2
29 3; Diamond;3
30 4; Diamond;4
31 5; Diamond;5
32 6; Diamond;6
```

```

10 As;Heart:10
11 9;Heart:9
12 10;Heart:10
13 J;Heart:10
14 Q;Heart:10
15 K;Heart:10
16 As;Diamond;1
17 2;Diamond;2
18 3;Diamond;3
19 4;Diamond;4
20 5;Diamond;5
21 6;Diamond;6
22 7;Diamond;7
23 8;Diamond;8
24 9;Diamond;9
25 10;Diamond;10
26 J;Diamond;10
27 Q;Diamond;10
28 K;Diamond;10
29 As;Club;1
30 2;Club;2
31 3;Club;3
32 4;Club;4
33 5;Club;5
34 6;Club;6
35 7;Club;7
36 8;Club;8
37 9;Club;9
38 10;Club;10
39 J;Club;10
40 Q;Club;10
41 K;Club;10

```

```

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

```

5.1.2. Step 2 – Random Card

The second step, in the second step we have to random cards in array .
Function to random cards is as follows.

```
5)
7
8 void randomize_cards() //random kartu
9 {
10     srand (time(NULL));
11     int r;
12     int angka_deck[52]; //untuk menampung index index kartu hasil random
13     bool kartu_terambil[52]; //untuk mengecek apakah kartu yang di random sudah ada di 'angka_deck'
14     for (i=0; i<52; i++) //looping 52/ada tempat kosong buat 52 kartu
15     {
16         kartu_terambil[i] = false; //menandi bahwa kartu belum masuk ke dalam 'angka_deck' kartu_terambil = [f.
17     }
18     for (i=0; i<52; i++) //looping 52 kartu untuk mengisi 'angka_deck'
19     {
20         r = rand()%52;
21         angka_deck[i] = r;
22         while (kartu_terambil[r] == true) //looping jika ternyata angka yang baru saja di random sudah masuk d
23         terambil sama dengan kartu yang di ambil maka akan di random
24         {
25             r = rand()%52;
26             angka_deck[i] = r;
27         }
28         kartu_terambil[r] = true;
29     }
30     for (i=0; i<52; i++)
31     {
32         deck[i] = record[angka_deck[i]]; //deck[i] menampung kartu yang di random
33     }
34 }
35
36 void urutkan_kartu(int index_nomal) //menurutkan id kartu pemain berdasarkan id kartu pada record
```

5.1.3 Step 3 – Game Started

The third step, the third step the game starting

```
}
randomize_cards();
while(p !=4)
{
    menu();
    printf("silahkan pilih menu:");
    scanf("%d",&p);
    switch(p)
    {
        case 1: printf("nama\t|jenis\t|nilai\t|\n\n");
                for(i=0; i<S2; i++)
                {
                    fprintf(stdout,"%s\t%s\t%d\n",record[i].nama,record[i].jenis,record[i].nilai); //
                }
                printf("\n");
                break;
        case 2: randomize_cards();
                for (i=0; i<S2; i++)
                {
                    printf("%s\t%s\t%d\n",deck[i].nama,deck[i].jenis,deck[i].nilai);
                }
                break;
        case 3: mulai_main();
                break;
        case 4: printf("Keluar\n");
                break;
        default:printf("Salah Masukan Menu\n");
                exit(0);
    }
}
return 0;
```

```
void mulai_main()
{
    jumlah_pemain();
    batas();
    bagi_kartu_semua(3); //tiap pemain mendapatkan 3 kartu awal
    lihat_kartu_main();
    while(q !=6)
    {
        menu_main();
        printf("silahkan pilih menu:");
        scanf("%d",&q);
        switch(q)
        {
            case 1: jawab_pemain();
                    break;
            case 2: bagi_kartu_semua(10);
                    lihat_kartu_main_semua();
                    break;
            case 3: urutkan_kartu_semua();
                    lihat_kartu_main();
                    break;
            case 4: printf("Lihat Minuman\n");
                    lihat_kartu_minuman();
                    break;
            case 5: reset_kartu();
                    randomize_cards();
                    bagi_kartu_semua(3);
                    lihat_kartu_main();
                    break;
            case 6: printf("Selesai main\n");
                    exit(0);
                    break;
        }
    }
}
```

Still in the third step, the user can set the number of players and the minimum value of the card. Function to set the number of players and the minimum value of the card is as follows.

```

3         pemain[i].total = 0;
0         pemain[i].jumlahkartu = 0;
1         for (j=0; j<13; j++)
2             {
3                 pemain[i].kartutangan[j].nilai = 0;
4             }
5     }
6 }
7
8 void jumlah_pemain() //fungsi mengatur jumlah pemain
9 {
0     player = 0;
1     while(player > 7 || player < 2) //batas player 2 sampai 7 / jika player lebih dari 7 atau kurang dari 2 loopin
2     {
3         printf("Berapa Pemain? "); scanf("%d", &player);
4     }
5 }
6 }
7
8 void batas() //fungsi mengatur batas minimal nilai kartu
9 {
0     batas_nilai = 0;
1     while(batas_nilai > 25 || batas_nilai < 20) //batas nilai angka 20 sampai 45 / jika batas nilai lebih dari 45 .
2     {
3         printf("Masukkan Batas Minimal Nilai Kartu (20-25): ");
4         scanf("%d", &batas_nilai);
5     }
6 }
7
8 void randomize_cards() //random kartu
9 {
0

```

After that, each player gets three cards at random. The function of each player receives three cards at random. This program is as follows.

```

1     cek_peringkat();
2 }
3
4 void bagi_kartu(int index_pemain) //bagi kartu per index pemain
5 {
6     pemain[index_pemain].kartutangan[pemain[index_pemain].jumlahkartu]=deck[kartu_teratas]; //memasukan kartu dari
7     tangan pemain pada index terakhir yang masih kosong
8     pemain[index_pemain].jumlahkartu++;
9     kartu_teratas++;
10 }
11
12 void bagi_kartu_semua(int jumlah)
13 {
14     while(jumlah>0)
15     {
16         cek_total();
17         for (i=0; i<player; i++)
18         {
19             if (pemain[i].total < batas_nilai) //jika total dari pemain ber index i kurang dari batas nilai
20             index i) di bagikan sebanyak jumlah
21             {
22                 bagi_kartu(i);
23             }
24         }
25         jumlah--;
26     }
27     cek_total();
28 }
29
30 void lihat_kartu_main()
31 {

```

```

38 {
39     cek_total();
40     printf("\n");
41     for(i=0; i<player; i++)
42     {
43         if (i==0)
44         {
45             printf("Kartu Anda\\t\\t");
46         }
47         else
48         {
49             printf("Kartu pemain %d\\t\\t", i);
50         }
51     }
52     printf("\n");
53     for(j=0; j<13; j++)
54     {
55         for(i=0; i<1; i++)
56         {
57             if (j >= pemain[i].jumlahkartu)
58             {
59                 printf("\\t\\t\\t");
60             }
61             else if(pemain[i].kartutangan[j].nilai!=0)
62             {
63                 printf("%s\\t\\s\\t%d\\t",pemain[i].kartutangan[j].nama,pemain[i].kartutangan[j].jenis,pe
64             }
65         }
66         printf("\\n");
67     }
68     for(i=0; i<1; i++)
69     {

```

5.1.4 Step 4 - Adding cards for user(kartu anda)

Once the user reaches a minimum value of the card, the user has the right to add the card if the card has not reached the target obtained. function is as follows.

```
78 }
79
80 void jawab_pemain()
81 {
82     char jawab;
83     printf("Anda ambil kartu? tekan Y or N. ");
84     scanf("%c", &jawab);
85     while (jawab == 'Y' || jawab == 'y')
86     {
87         bagi_kartu(0);
88         lihat_kartu_main();
89     }
90     printf("Ambil kartu lagi? Y or N. ");
91     scanf("%c", &jawab);
92 }
93
94 void lihat_kartu_mai();
95 }
96
97 void lihat_kartu_minuman()
98 {
99     for (i=kartu_teratas; i<52; i++)
100     {
101         printf("%s\t%s\t%s\t\n",deck[i].nama,deck[i].jenis,deck[i].nilai);
102     }
103 }
104
105 void mulai_main()
106 {
107     jumlah_pemain();
108     while(jumlah_pemain() < 2)
109     {
110         for (i=0; i<jumlah_pemain(); i++)
111         {
112             if (i == pemain[i].jumlahkartu)
113             {
114                 break;
115             }
116             total = total + pemain[i].kartutangan[j].nilai;
117         }
118         pemain[i].total = total;
119         if (pemain[i].total > 30)
120         {
121             pemain[i].kalah = true;
122         }
123     }
124     cek_peringkat();
125 }
126
127 void bagi_kartu(int index_pemain) //bagi kartu per index pemain
128 {
129     pemain[index_pemain].kartutangan[pemain[index_pemain].jumlahkartu]=deck[kartu_teratas]; /
130     tangan pemain pada index terakhir yang masih kosong
131     pemain[index_pemain].jumlahkartu++;
132     kartu_teratas++;
133 }
134
135 void bagi_kartu_senua(int jumlah)
136 {
137     while(jumlah>0)
138     {
139         bagi_kartu(jumlah-1);
140     }
141 }
```

5.1.5 Step 5 – Game Over

Once the user reaches the maximum value or the user is not willing to add another card user can view the results. function is as follows.

```
211 )
212
213 void bagi_kartu(int index_pemain; //bagi kartu per index pemain
214 {
215     pemain[index_pemain].kartutangan[pemain[index_pemain].jumlahkartu]-deck[kartu_teratas]; //memasukan kartu dari
    tangan pemain pada index terakhir yang masih kosong
216     pemain[index_pemain].jumlahkartu++;
217     kartu_teratas++;
218 }
219
220 void bagi_kartu_serua(int jumlah;
221 {
222     while(jumlah>0)
223     {
224         cek_total();
225         for (i=0; i<player; i++)
226         {
227             if (pemain[i].total < batas_nilai) //jika total dari pemain ber index i kurang dari batas nilai
    index i) di bagikan sebanyak jumlah
228             {
229                 bagi_kartu(i);
230             }
231         }
232         jumlah--;
233     }
234     cek_total();
235 }
236
237 void lihat_kartu_pain()
238 {
239     cek_total();
240     printf("\n");
```

```
281 {
282     cek_total();
283     printf("\n");
284     for(i=0; i<player; i++)
285     {
286         if (i==0)
287         {
288             printf("Kartu Anda\n");
289         }
290         else
291         {
292             printf("Kartu pemain %d\n", i);
293         }
294     }
295     printf("\n");
296     for(j=0; j<13; j++)
297     {
298         for(i=0; i<player; i++)
299         {
300             if (j >= pemain[i].jumlahkartu)
301             {
302                 printf("\n");
303             }
304             else if(pemain[i].kartutangan[j].nilai==0)
305             {
306                 printf("\s\t\s\t\s\t\s\t",pemain[i].kartutangan[j].nama,pemain[i].kartutangan[j].jenis,i
307             }
308         }
309     }
310     printf("\n");
311     for(i=0; i<player; i++)
```


5.2 Interface

5.2.1 Main Menu Window

Step by step for playing card samgong

Step 1 : Compile and run samgong.c

```
File Edit View Search Terminal Help
andri@andri-PC:~$ su
Password:
root@andri-PC:~/home/andri# cd desktop/
root@andri-PC:~/home/andri/Desktop# gcc -o samgong samgong.c
root@andri-PC:~/home/andri/Desktop# ./samgong
Menu:
1. Menampilkan Data Kartu
2. Random Kartu
3. Mulai Main
4. Keluar
silahkan pilih menu: [1]
```

Figure 5.2.1 First menu

Step 2 : Select menu one to see the card data

```
Menu:
1. Menampilkan Data Kartu
2. Random Kartu
3. Mulai Main
4. Keluar
silahkan pilih menu: [1]
nama [1-10][1-10]: [10][10]
A1 Spade 1
2 Spade 2
3 Spade 3
4 Spade 4
5 Spade 5
6 Spade 6
7 Spade 7
8 Spade 8
9 Spade 9
10 Spade 10
J Spade 10
Q Spade 10
K Spade 10
A2 Heart 1
2 Heart 2
3 Heart 3
4 Heart 4
5 Heart 5
6 Heart 6
7 Heart 7
8 Heart 8
9 Heart 9
10 Heart 10
```

Figure 5.2.2 Showing the cards

Step 3: Select menu two for random cards

```
3      Club  3
As     Heart 1
3      Spade 3
8      Heart 8
10     Heart 10
As     Diamond 1
5      Spade 5
9      Diamond 10
10     Diamond 10
K      Heart 10
2      Heart 2
A      Club  10
3      Diamond 5
9      Diamond 9
3      Club  2
8      Diamond 8
9      Heart 9
10     Spade 10
J      Spade 10
A      Diamond 3
8      Spade 8
7      Diamond 7
As     Club  1
K      Diamond 10
9      Spade 9
2      Diamond 2
10     Club  10
10     Club  10
As     Spade 1
3      Diamond 10
K      Spade 10
```

Figure 5.2.3 Random Cards

Step 4: Select menu three for starting game (mulai main)

```
7      Club  7
Menu:
1. Menampilkan Data Kartu
2. Random Kartu
3. Mulai Main
4. Keluar

Silahkan pilih menu:
Berapa Pemain? > 3 Pemain: 4
Masukan Data Awal Awal Kartu: (29-25) 29

Kartu Anda      Kartu pemain 1      Kartu pemain 2      Kartu pemain 3
3      Club  3
10     Heart 10
10     Diamond 10

Total Awal: 29
Peringkat: 2
Menu:
1. Menampilkan Data Kartu Anda
2. Lihat Hasil Akhir
3. Uraikan Kartu
```

Figure 5.2.4 Game Started

The fourth step, the fourth step the game starts the player can set the number of players and the minimum value of the card. Function to set the number of players and the minimum value. after that each player gets three cards at random.

Step 5: Take the card decks for user

```
Total nilai: 22
Peringkat: 2
Menu:
1. Mengambil Minusai kartu Anda
2. Lihat Hasil Akhir
3. Urutkan Kartu
4. Lihat Minusai
5. Ulang
6. Selesai
Silahkan pilih menu:
Anda ambil kartu? tekan Y or N: y

Kartu Anda          Kartu pemain 1      Kartu pemain 2      Kartu pemain 3
J          Club      3
10         Heart     10
10         Diamond    10
5          Diamond    5

Total nilai: 20
Peringkat: 1
Ambil kartu lagi? y or n: n
```

Figure 5.2.5 Add the Cards for user

Users are required to add a minimum limit of the value of the card if the card has not been reached. provided that the addition of the card until the card at least a minimum threshold value is met.

Step 6: See the end result

```

Total Nilai: 28
Peringkat: 1
Menu:
1. Pengasbil Nilai dan Kartu Anda
2. Lihat Hasil Akhir
3. Buatkan Kartu
4. Lihat Nilai
5. Ulangi
6. Selesai
silahkan pilih menu:2

Kartu Anda          Kartu pemain 1      Kartu pemain 2      Kartu pemain 3
3      Club  3      A5      heart  1      3      Spade  3      8      heart  8
10     heart  10     A5      Diamond 1      5      Spade  5      9      Diamond 10
10     Diamond 10     K      heart  10     7      heart  7      K      Club  10
5      Diamond 5      5      Diamond 5      2      Club  2
8      Diamond 8      9      heart  9
9      Spade  10

Total Nilai: 28      Total Nilai: 29      Total Nilai: 31      Total Nilai: 28
Peringkat: 2        Peringkat: 1        Peringkat: 4        Peringkat: 2
Menu:
1. Pengasbil Nilai dan Kartu Anda
2. Lihat Hasil Akhir
  
```

Figure 5.3.6 End Result

If the user feels the value of the card has reached the target or the minimum value of the card are met the user can see the final result. the final result is known players the winning and losing

Step 7: Repeated game

```
Total Nilai: 28      Total Nilai: 29      Total Nilai: 31      Total Nilai: 28
Peringkat: 2        Peringkat: 1        Peringkat: 4        Peringkat: 2
Menu:
1. Mengambil Minus Kartu Anda
2. Lihat Hasil Akhir
3. Lihat Kartu
4. Lihat Minus
5. Ulangi
6. Selesai
silahkan pilih nomor:

Kartu Anda      Kartu pemain 1      Kartu pemain 2      Kartu pemain 3
10  Diamond 10
6   Club     6
0   Spade   10

Total Nilai: 26
Peringkat: 2
Menu:
1. Mengambil Minus Kartu Anda
2. Lihat Hasil Akhir
```

Figure 5.3.7 Repeated Game Samgong

The game is repeated, starting a new game again each player gets three cards at random.