

LAMPIRAN

Lampiran 1 Data Penempatan *Shuttlecock*

	P1	P2	P1	P2	P1	P2	P1	P2	P1	P2
1	H,1	F,2	F,1	C,1	H,1	A,2	A,1			
2	H,1	I,1	G,1	A,3	A,2	E,2				
3		H,1	H,1	I,1	F,1	F,2	E,3	B,2	C,1	
4	H,1	C,1	I,2							
5	H,1	F,1	B,2	F,1	B,1	F,1	H,2	H,1	C,2	D,3
6	H,2	H,1	B,1	C,1	C,1	H,1	C,2	C,1	I,2	D,1
7	H,1	A,1	G,1	A,2	G,2					
8	G,1	H,1	H,1	H,1	C,1	B,1	H,1	F,2	C,2	I,1
9		G,1	B,1	C,1						
10		H,1	D,1	E,1	E,1					
11	I,1	F,1	I,2	B,2	H,2	C,2	I,1	I,1		
12		G,1	F,1	B,1	C,1	C,1	H,1	C,2	H,2	C,1
13	H,1	F,1	B,1	F,2	B,2	B,1	H,1	C,1	C,1	I,2
14		I,1	C,1	C,1	C,1	B,2	E,2			
15	H,1	F,1	I,2	C,2	C,2	H,1				
16		G,1	C,1	C,1	G,2	B,1	C,2	C,2	A,1	
17	H,1	F,1	B,2							
18	H,1	D,2	E,2	E,2	D,3	G,1	B,2	B,1		
19	H,1	H,2	G,2	A,2	E,2	A,1	G,1	C,2	A,2	E,1
20		H,1	E,2	A,3	I,1	H,2				
21		G,1	F,2	C,2	C,1	B,4	I,2			
22	H,1	H,1	B,2	F,2	B,1	G,1	C,3	C,2	A,3	
23	H,1	G,1	F,1	B,1	B,2	H,2	G,2			
24	H,1	C,1	I,1	G,2	B,2	H,2	G,1	C,1	C,2	C,1
25		H,1	E,2	I,2	D,3	D,4	F,2	A,1	G,2	B,2
26		H,1	B,2	A,2	H,1	F,1	C,3	F,2	C,1	F,1
27	H,1	A,2	C,2	H,1	E,1	A,1				
28		H,1	C,1	C,1	H,1	C,1	C,2	H,1	C,2	E,2
29	H,1	I,2	G,1	I,2	H,3	D,2	C,3	H,2	A,2	F,1

30		H,1	E,1	B,1	G,2	C,2	C,1	B,3		
31	H,1	G,1	E,1	G,1	C,1	A,1	H,2	I,1	A,1	C,2
32		G,1	C,1	C,2	C,2	H,2	H,1			
33	H,1	F,1	C,2	G,2	B,1	C,1	E,1	G,2	A,2	
34	H,1	C,2	H,2	G,1						
35		H,2	I,1							
36	H,1	C,1								
37		H,1	B,2	H,2						
38		H,1	H,1	I,2	A,2	A,2	H,1	A,3	H,2	C,3
39		H,1	C,1	H,2	C,2					
40	H,1	F,1	B,1	A,1	A,2	A,1	I,1	I,2	G,2	G,2
41		I,1	D,1	C,2						
42		H,1	F,2							
43	H,2	E,1	F,1	A,2	G,2	G,2	I,1	A,2		
44		I,1	G,1	C,1	C,2	G,1	H,2	C,1	C,1	I,1
45	H,1	G,1	A,1	C,1	H,1	B,1	B,2	I,2	G,1	I,1
46	H,1	C,2	B,2	H,1	A,2	A,2	A,2	E,1		
47		H,1	C,2	E,3	I,2					
48		I,1	D,1	E,2	B,2	G,2	I,1			
49	G,2	E,1								
50		G,1	E,2	A,2	H,1	E,2	B,1	E,1	B,1	
51	H,1	E,1	B,1	I,2	H,1	E,3	C,2			
52	H,1	I,2	G,2	G,2	H,2	F,3				
53		H,1	I,2	A,1	A,2	H,1	G,2	I,3	A,2	G,2
53		G,1	D,2	E,3	C,2					
55	H,1	F,1	C,1	I,2	G,2	A,3				
56		H,1	D,1	A,1						
57		G,1	F,1	C,1	G,1	C,2	C,2	C,1		
58		H,1	I,1	G,1	A,1	G,2	A,1	I,1		
59		H,1	F,1	H,1						
60		H,1								
61		H,1	I,2							

62	H,1	E,1	H,1	C,1	H,1	E,2	I,2	A,2	G,1	G,1
63		H,1	H,1	G,1	I,2	I,2	G,2	I,2	B,2	B,1
64		H,1	G,1							
65	I,1	A,2	A,2							
66	H,1	E,2	B,2	I,2	G,2	A,1	A,1	G,1	I,1	H,2
67		H,1	H,1	I,1	F,2	A,2	A,2	E,1	G,1	
68	H,1	D,1	F,1	F,2						
69		G,1	C,1	I,1	H,2	G,1				
70		H,1	F,1	E,2	H,2	E,2	B,3	H,2		
71		H,1	I,1	B,1	B,1	H,1	A,2	G,2	I,2	A,2
72		C,1	I,2	H,2	A,2					
73	H,1	G,2	C,2	A,2	I,2	H,1	C,1	H,3	G,2	
74	H,1	C,2	I,1	C,2	C,2	H,2	H,1	A,1	H,1	A,1
75		H,1	I,1	G,2	F,1	C,1				
76		G,1	C,2	C,1	C,1	C,1	H,2	D,3		
77		H,2	G,1	C,1	C,1	G,2	A,2	A,1	G,1	G,1
78		H,1	I,1							
79	H,1	F,2	C,2	H,3	F,2	H,3	H,2			
80	H,1	E,2	A,1	A,1	H,1	H,2	C,2	C,2	I,1	F,1
81		H,2	G,1	G,1	A,1	H,1	H,2			
82	H,1	C,2	G,2	H,1						
83		H,1	G,2	C,1	C,1	B,1	I,2	B,1	B,2	
84	H,1	F,1	B,1	H,1	A,1	F,2	H,2			
85	H,1	G,1	H,1	C,1	C,2	H,2	A,2	F,2	H,1	I,1
86		C,1	H,4	A,4	E,2					
87	I,2	H,1	G,1	H,1						
88		B,1	C,2	C,2						
89		C,1								
90		B,1	I,2	H,1						
91		B,1	I,1							
92	H,1	E,1	C,1	C,2						
93		B,1	C,2	I,1	F,2	H,1	G,1	E,1	D,1	A,2

94		H,1	C,1	G,1	I,2	A,1
95		H,1	F,2	H,2	D,3	
96	H,1	C,1	G,1	C,1	B,1	A,2

Lampiran 2 Kode Program Strategi Penempatan Shuttlecock

```

source('F:/AAAAA/Data/Skripsi/Koding/areapukulan.R')
#library("kamila", lib.loc="~/R/win-library/3.4")
MyData <-
data.frame(read.csv(file="F:/AAAAA/Data/Skripsi/Koding/areap
ukulan.csv", header=FALSE, sep=","))

pukulan = 3
if(is.numeric(pukulan) == FALSE){
  print("ERROR! Inputan tidak berupa integer")
}
if(is.numeric(pukulan) == TRUE && pukulan <= 0){
  print("ERROR! Masukan tidak boleh tidak boleh atau sama
dengan nol")
}else{
pattern=c("I,2","D,3","A,4")#Inputan user

jmlCluster = 2
tampHasil = kumpulanpattern(pattern, MyData, pukulan)
hasilprediksi = prediksi(tampHasil,pukulan,1)
hasilprediksi = lapply(hasilprediksi, function(x)
x[!is.na(x)])
hasilprediksi = lapply(hasilprediksi, function(x) x[!x==""])
hasilprediksi = cleaner(hasilprediksi, pukulan, pattern)
printprediksi(hasilprediksi, pukulan, pattern) #hasil kmp
hasil=predictWithKamila(tampHasil, pukulan,jmlCluster)
hasil=hasilkamila(hasil[[1]], tampHasil, jmlCluster)
print(hasil) #hasil kamila
print(Sys.time()- strt)

#buang spasi dari data
buangSpasi <- function(teks){
  for(i in 1:length(teks)){
    teks[i]=gsub(" ", "", teks[i], fixed = TRUE)
  }
  return(teks)
}

#fungsi cek pattern
cekpattern <- function(pattern){
  if(!is.null(pattern)){
    if(grepl(",", pattern[1])){
      #print("Ada Koma")
      return(0)#untuk ada koma
    }else{
      #print("Tanpa Koma")
      return(1)#tanpa koma
    }
  }
}

#fungsi membuang tanda koma

```

Muh. Irfan Firmansyah, 2018

PENENTUAN STRATEGI PENEMPATAN SHUTTLECOCK DENGAN ALGORITMA KNUTH-MORRIS-PRATT

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

```

buangKoma <- function(teks){
  karakterBaru = c()
  for(i in 1:length(teks)){
    hasilSplit = unlist(strsplit(teks[i], split = ','))
    if(!is.na(hasilSplit[1])){
      karakterBaru = c(karakterBaru,hasilSplit[1])
    }else{
      karakterBaru = c(karakterBaru,"")
    }
  }
  return(karakterBaru)
}

#KMP_Prefix Function -----
#Fungsi untuk memberikan nilai prefix dalam algoritma utama
KMP
KMP_Prefix <- function(pattern){

  #inisiasi variabel
  n_pattern <- length(pattern)
  prefix <- c(0)
  a <- 0

  #pembentukan pattern
  for(b in 2:n_pattern){
    while(a > 0 && pattern[a+1] != pattern[b]){
      a <- prefix[a]
    }
    if(pattern[a+1] == pattern[b]){
      a <- a+1
    }
    prefix[b] <- a
  }

  #Melempar hasil prefix
  return(prefix)

}

KMP <- function(prefix, text, pattern, tampungan){
  i=1
  j=1
  k=1
  returnval=NULL #array untuk menyimpan tiap patern yang
sama di dalam teks
  n=length(text)
  m=length(pattern)
  ditemukan = 0

  potong=(text[k:(k+m-1)]) #cek pattern sebelumnya

  #mencari pattern yang paling awal

```

```

if(all(potong==pattern)){
  returnval=append(returnval,k)
}

while ((n-k)>=m && ditemukan == 0){
  while (j<m && text[i]==pattern[j]){
    i=i+1
    j=j+1
  }

  if (j>m) ditemukan = 0
  if (j==1) j=2

  if (prefix[j-1]>0){
    k=i-prefix[j-1]
  }else{
    if (i==k) i=i+1
    k=i
  }

  if (j>1) j=prefix[j-1]+1
  potong=(text[k:(k+m-1)]) #cek pattern sebelumnya

  if(all(potong==pattern)){
    returnval=append(returnval,k)
  }
}
return(returnval+m)
}

#fungsi tampung karakter setelah inputan pattern
kumpulanpattern <- function(pattern, MyData, jml.pukulan){
  prefix1=KMP_Prefix(pattern)
  hasilCekPattern = cekpattern(pattern)
  tampHasil = list()
  indeks = 1

  for(i in 1:nrow(MyData)){
    #Convert Integer to String and Put to variable teks
    teks = c()
    teksOri = c()
    if(hasilCekPattern == 0){
      for(j in 1:ncol(MyData)){
        teks = c(teks, toString(MyData[i,j]))
        teks = buangSpasi(teks)
      }
    }else{
      for(j in 1:ncol(MyData)){
        teks = c(teks, toString(MyData[i,j]))
        teksOri = c(teksOri, toString(MyData[i,j]))
        teks = buangSpasi(teks)
        teks = buangKoma(teks)
      }
    }
  }
}

```



```

    }
  }

  #KMP Process
  hasil = KMP(prefix1, teks, pattern, tampungan = c())

  if(length(hasil) != 0){
    nextChar = c()
    if(hasilCekPattern==0){
      for(k in hasil:(hasil+(pukulan-1))){
        nextChar = c(nextChar,teks[k])
      }
    }else{
      for(k in hasil:(hasil+(pukulan-1))){
        nextChar = c(nextChar,teksOri[k])
      }
    }
  }
}
}
#print(tampHasil)
#print(cbind(length(tampHasil), indeks))
return(tampHasil)
}

#CONVERSI MATRIKS JADI LIST
convertToList <- function(matrixtampHasil = matrix()){
  daftarhasil = list()
  indekslist = 1

  for(i in 1: ncol(matrixtampHasil)){
    tampstring <- c()
    counterstr = 0
    for(j in 1: nrow(matrixtampHasil)){
      if(is.null(matrixtampHasil[j,i])==FALSE ||
matrixtampHasil[j,i]!=""){ #string kosong atau isi
        tampstring = c(tampstring, matrixtampHasil[j,i])
      }else{
        counterstr = counterstr + 1
      }
    }
    if(counterstr != nrow(matrixtampHasil)){ #cek semua
string kosong
      daftarhasil[[indekslist]] =
tampstring[!is.na(tampstring)] #for nul
      daftarhasil[[indekslist]] = tampstring[tampstring!=""]
#for karakter kosong
      indekslist = indekslist + 1
    }
  }
  return(daftarhasil)
}
}

```

```

#fungsi output tanpa (,)
outputtanpakoma <- function(tanpakoma){
  hasil = list()
  for(i in 1:length(tanpakoma)){
    setKarakter = tanpakoma[[i]]
    karakterBaru = c()
    for(j in 1:length(setKarakter)){
      hasilSplit = unlist(strsplit(setKarakter[j], split =
', '))
      karakterBaru = c(karakterBaru, hasilSplit[1])
    }
    hasil[[i]]=karakterBaru
  }

  return(hasil)
}

#FUNGSI MEMBUANG SET KARAKTER YANG TIDAK MEMENUHI SYARAT
SERING MUNCUL
hapusyggapenting <- function(matrixtampHasil, indexBaris){
  pembandingan = unique(matrixtampHasil[indexBaris,])#ambil
string untuk dibandingkan
  jumlah = array(0, length(pembandingan))#tampung jumlah
  daftarIndex = c()#tampung indeks yg akan dibuang

  #dapatkan jumlah masing masing karakter
  for(i in 1:length(pembandingan)){
    jumlah[i]=length(grep(pembandingan[i],
matrixtampHasil[indexBaris,]))
  }

  #ubah jumlah yg string kosong menjadi 0
  indeksKKosong = which(pembandingan=="")
  jumlah[indeksKKosong] = 1

  #ambil karakter yang sering muncul
  karakter = pembandingan[which.max(jumlah)]

  #ambil indeks yang akan dihapus
  for(i in 1:ncol(matrixtampHasil)){
    if(matrixtampHasil[indexBaris,i]!=karakter){
      daftarIndex = c(daftarIndex, i)
    }
  }

  #cek keseluruhan berjumlah sama
  jumlahstringsama = length(grep(jumlah[1], jumlah))

  #hapus kolom yang tidak memenuhi syarat sering muncul
karakter
  if(jumlahstringsama != length(jumlah)){
    matrixtampHasil = matrixtampHasil[,-daftarIndex]
  }
}

```

```

    return(matrixtampHasil)
}

#hapus yang kurang dari jumlah next pukulan
deletayangnull <- function(daftarH, pukulan){
  if(length(daftarH) != 0){
    hasill = list()
    ind = 1
    for(i in 1:length(daftarH)){
      if(length(daftarH[[i]]) == pukulan){
        hasill[[ind]]=daftarH[[i]]
        ind = ind + 1
      }
    }
    return(hasill)
  }
}

#Fungsi hasil prediksi
prediksi <- function(tampHasil,pukulan,indekskolom){
  if(indekskolom < pukulan && length(tampHasil) > 2){
    #ambil huruf sesuai indekskolom
    krt = c()
    for (i in 1:length(tampHasil)) {
      krt = c(krt,tampHasil[[i]][indekskolom])
    }
    krt = unique(krt)

    #Jumlah kemunculan huruf
    jmlhuruf = array(0,dim = length(krt))
    for (i in 1:length(krt)){
      #jika karakter yg dibanding bukan string kosong atau
null
      if(krt[i]!=""){
        for (j in 1:length(tampHasil)){
          if(krt[i] == tampHasil[[j]][indekskolom]){
            jmlhuruf[i] = jmlhuruf[i] + 1
          }
        }
      }else{
        jmlhuruf[i]=1
      }
    }

    #ambil huruf yg sering muncul
    hrf=krt[which.max(jmlhuruf)]
    jmlhrf=jmlhuruf[which(krt==hrf)]
    krtYGSama=krt[which(jmlhuruf==jmlhrf)]

    #print(c(hrf,jmlhrf))
    #print(krtYGSama)
  }
}

```

```

#cek apabila seluruh karakter berjumlah sama
jmlYGSama = length(grep(jmlhuruf[1],jmlhuruf))

#print(rbind(krt,jmlhuruf))

#tampung prediksi
if(length(krtYGSama)>1 && jmlYGSama <
length(jmlhuruf)){#ada beberapa yg sama karakternya
  lhasil = list()
  indeklhasil = 1
  for(k in 1:length(krtYGSama)){
    hrf = krtYGSama[k]
    for (i in 1:length(tampHasil)){
      if(tampHasil[[i]][indekskolom] == hrf){
        lhasil[[indeklhasil]]=tampHasil[[i]]
        indeklhasil = indeklhasil + 1
      }
    }
  }
  return(prediksi(lhasil,pukulan,indekskolom =
indekskolom+1))
}else if(length(krtYGSama)==1 && jmlYGSama !=
length(jmlhuruf)){ #jika tidak sama semua jumlahnya
  lhasil = list()
  indeklhasil = 1
  for (i in 1:length(tampHasil)){
    if(tampHasil[[i]][indekskolom] == hrf){
      lhasil[[indeklhasil]]=tampHasil[[i]]
      indeklhasil = indeklhasil + 1
    }
  }
  return(prediksi(lhasil,pukulan,indekskolom =
indekskolom+1))
}else{ #jika sama semua jumlahnya
  return(prediksi(tampHasil,pukulan,indekskolom =
indekskolom+1))
}

}
return(tampHasil)
}

#fungsi print hasil prediksi
cleaner <- function(hasilprediksi, pukulan, pattern){
  if(length(hasilprediksi) == 2){
    x = round(runif(1,1,2))
    hasilprediksi = list(hasilprediksi[[x]])
  }else if(length(hasilprediksi) > 2){
    hasilprediksi = konversi(hasilprediksi)
    hasilprediksi = prediksi(hasilprediksi,pukulan*2,1)
    return(cleaner(hasilprediksi, pukulan, pattern))
  }
  return(hasilprediksi)
}

```

```

}

printprediksi <- function(hasilprediksi, pukulan, pattern){
  if(length(hasilprediksi)==1){
    if(grepl(",",pattern[1])){#output jika input dengan
pukulan
      print(hasilprediksi)
    }else{#output jika input hanya pattern
      print(hasilprediksi)
      print(outputtanpakoma(hasilprediksi))
    }
  }else{
    print("Input pattern tidak sesuai. Coba lagi")
  }
}

#fungsi konversi string jadi 1 string 1 karakter
konversi <- function(temp){
  return(lapply(temp, function(x) unlist(strsplit(x,
"[,]"))))
}

#fungsi karakter ke ascii
konversikeascii <- function (temp){
  for(i in 1:length(temp)){
    temp[[i]]=unlist(lapply(temp[[i]], function (x)
utf8ToInt(x)))
  }
  return(temp)
}

#fungsi konversi ke string asli
convertToStringPattern <- function(pattern){
  hasil = array()
  indeksHasil = 1
  tamp = ""
  for(i in 1:length(pattern)){
    if(tamp == ""){
      tamp = pattern[i]
    }else{
      hasil[indeksHasil] = paste(tamp, pattern[i], sep = "")
      tamp = hasil[indeksHasil]
      if((i %% 3) == 0){
        tamp = ""
        indeksHasil=indeksHasil+1
      }
    }
  }
  return(hasil)
}

#konversi list ke matrix
konversilistmatrix <- function(temp, pukulan){

```

```

jmlkolom = pukulan*2
mhasil = matrix(0,ncol = jmlkolom, nrow = length(temp))
for(i in 1:length(temp)){
  for(j in 1:jmlkolom){
    if(is.na(temp[[i]][j])){
      mhasil[i,j] = 0
    }else{
      mhasil[i,j] = temp[[i]][j]
    }
  }
}
return(mhasil)
}

#using kamila
predictWithKamila <- function(temp, pukulan,jmlCluster){
  hasilX = konversi(temp) #1 string 1 karakter
  hasilY = konversikeascii(hasilX) #string ke ascii
  hasilZ = konversilistmatrix(hasilY, pukulan) #list ke
matrix
  dat <- genMixedData(length(temp), nConVar=2, nCatVar=2,
nCatLevels=4, nConWithErr=2,
nCatWithErr=2,
popProportions=c(.5,.5), conErrLev=0.3, catErrLev=0.8)
  msRes = gmsClust(hasilZ, hasilZ, jmlCluster)

return(list(msRes$results[[1]],table(msRes$results$cluster,
dat$trueID)))
  ## End
}

#output dari kamila
hasilkamila <- function(indexcluster, tampHasil,
jmlCluster){
  tampind = c()
  for(i in 1:jmlCluster){
    ketemu = 0
    indeks = 1
    while (ketemu == 0 && indeks <= length(indexcluster)){
      if(i == indexcluster[indeks]){
        ketemu=1
        tampind= c(tampind, indeks)
      }
      indeks = indeks+1
    }
  }

  kesimpulan = list()
  for(i in 1:length(tampind)){
    kesimpulan[[i]]=tampHasil[[tampind[i]]]
  }
  return(kesimpulan)
}

```

Lampiran 3 Lembar Bimbingan



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN

UNIVERSITAS PENDIDIKAN INDONESIA
 FAKULTAS PENDIDIKAN MATEMATIKA DAN ILMU PENGETAHUAN ALAM
 DEPARTEMEN PENDIDIKAN ILMU KOMPUTER
 PRODI ILMU KOMPUTER

Jalan Dr. Setiabudi No. 229 Bandung 40154 Telp. 022-2007031 Inter 5308 e-mail:cs@upi.edu

FORM BIMBINGAN SKRIPSI

Nama : Muh. Irfan Firmansyah NIM : 1301643
 Program studi : Ilmu Komputer
 Dosen Pembimbing : 1. Dr. Lala Septon Riza, M.T.
 2. Herbert Siregar, M.T.

JUDUL SKRIPSI :

Strategi Penempatan Shuttlecock dengan algoritma
~~sh~~ Knuth-Morris Pratt

Pertemuan/ tgl	Uraian	Paraf Pembimbing

Catatan : _____

Menyatakan bahwa dokumen skripsi dengan judul :

Telah siap untuk di Prasadangkan.

Pembimbing I

Pembimbing II



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN

UNIVERSITAS PENDIDIKAN INDONESIA
FAKULTAS PENDIDIKAN MATEMATIKA DAN ILMU PENGETAHUAN ALAM
DEPARTEMEN PENDIDIKAN ILMU KOMPUTER
PRODI ILMU KOMPUTER

Jalan Dr. Setiabudi No. 229 Bandung 40154 Telp. 022-2007031 Inter 5308 e-mail:cs@upi.edu

FORM BIMBINGAN SKRIPSI

Nama : Muh. Irfan Firmansyah NIM : 1301643
 Program studi : Ilmu Komputer
 Dosen Pembimbing : 1. Dr. Lala Septen Riza, M.T.
 2. Herbert Siregar, M.T.

JUDUL SKRIPSI :

Strategi Penempatan shuttlecock dengan algoritma Knuth Morris Pratt.

Pertemuan/ tgl	Uraian	Paraf Pembimbing

Catatan : _____

Menyatakan bahwa dokumen skripsi dengan judul :

Telah siap untuk di Prasadangkan.

Pembimbing I

Pembimbing II

Herbert S.