

1. PENDAHULUAN

Performa merupakan istilah yang digunakan untuk menunjukkan seberapa baik seseorang atau sesuatu dalam melakukan suatu pekerjaan atau aktivitas tertentu. Performa cabang dalam suatu perusahaan merupakan hal yang penting untuk diperhatikan. Pada project ini penulis akan menganalisis performa cabang dari perusahaan pembiayaan. Perusahaan pembiayaan adalah badan usaha di luar Bank dan Lembaga Keuangan Bukan Bank yang memberikan fasilitas pinjaman kepada nasabahnya. Berdasarkan total uang yang dicairkan, jumlah hari pencairan, dan lain sebagainya dapat dilihat performa cabang dari perusahaan pembiayaan.

2. METODE PENELITIAN

Data yang digunakan pada project ini yaitu data sekunder yang berasal dari salah satu perusahaan pembiayaan. Data tersebut berisikan ID pinjaman, tanggal cair, nama cabang, agen, dan total uang yang dicairkan. Data yang diambil adalah data bulan Januari 2020 sampai Mei 2020. Dalam project ini analisis data dilakukan dengan bantuan *software* R. Langkah-langkah yang dilakukan pada project ini yaitu:

1. Memisahkan data pada bulan terakhir dan menjumlahkan data per cabang.
2. Menentukan data dari lima cabang yang memiliki total uang yang dicairkan paling besar.
3. Menentukan data dari lima cabang yang memiliki total uang yang dicairkan paling kecil.
4. Menghitung umur masing-masing cabang dalam bulan.
5. Melihat performa cabang berdasarkan umur.
6. Membandingkan performa antara cabang dengan performa baik dan cabang dengan performa rendah.
7. Interpretasi hasil.

3. HASIL DAN PEMBAHASAN

1. Memisahkan Data pada Bulan Terakhir dan Menjumlahkan Data per Cabang

Sebelum memisahkan data bulan Mei 2020, terlebih dahulu akan dipanggil data yang berisikan ID pinjaman, tanggal cair, nama cabang, agen, dan total uang yang dicairkan dari perusahaan pembiayaan. Berikut adalah perintah yang digunakan:

```
#syntax untuk memanggil data  
  
>library(readr)  
  
>data1<-read.csv("C:/Users/Asus/Documents/R/data_cabang.csv",  
stringsAsFactors=F)
```

```
> data1
```

Setelah data berhasil muncul, selanjutnya pisahkan data pada bulan terakhir dan jumlahkan data per cabang dengan menggunakan perintah berikut:

```
#Memfilter data bulan Mei 2020 dan menjumlahkan data per cabang
#library(dplyr) digunakan untuk analisis dan manipulasi data
>library(dplyr)
>data_Mei<-data1%>%
>filter(tanggal_cair>='2020-05-01',tanggal_cair<='2020-05-31')%>%
#memfilter data
>group_by(cabang)%>%
>summarise(total_amount=sum(amount)) #mengubah beberapa nilai menjadi
satu ringkasan nilai
>data_Mei
```

Output yang dihasilkan dari perintah tersebut adalah:

```
# A tibble: 22 x 2
  cabang total_amount
  <chr>         <dbl>
1 AA          75710000
2 AB          81440000
3 AC          83990000
4 AD          76080000
5 AE          54200000
6 AF          68040000
7 AG          74080000
8 AH          73840000
9 AI          46640000
10 AJ          43580000
# ... with 12 more rows
```

2. Menentukan Data dari Lima Cabang yang Memiliki Total Uang yang Dicairkan Paling Besar.

Dalam melihat performa cabang, perlu diperhatikan total uang yang dicairkan dari cabang tersebut. Oleh karena itu, akan dicari lima cabang yang memiliki total uang yang dicairkan paling besar. Perintah yang digunakan dalam mencari lima cabang dengan total uang yang dicairkan paling besar adalah sebagai berikut:

```
#Menampilkan data dari lima cabang yang memiliki total amount paling
besar

>library(scales)

>data_Mei %>%

>arrange(desc(total_amount))%>% #desc digunakan untuk mengurutkan dari
yang terbesar

#untuk membuat variabel baru berdasarkan variabel yang sudah ada
menggunakan mutate

>mutate(total_amount= comma(total_amount))%>% #comma digunakan untuk
menampilkan total_amount dari kelima cabang tersebut

>head(5)
```

Output yang diberikan dari perintah tersebut adalah sebagai berikut:

```
# A tibble: 5 x 2
  cabang total_amount
  <chr>   <chr>
1 AC      83,990,000
2 AB      81,440,000
3 AD      76,080,000
4 AA      75,710,000
5 AG      74,080,000
```

Dapat dilihat bahwa cabang-cabang dengan total uang yang dicairkan paling besar yaitu cabang AC, AB, AD, AA, AG.

3. Menentukan Data dari Lima Cabang yang Memiliki Total Uang yang Dicairkan Paling Kecil.

Cabang-cabang dengan total uang yang dicairkan paling besar perlu dibandingkan dengan cabang-cabang yang memiliki total uang yang dicairkan paling kecil. Hal tersebut perlu dilakukan agar terlihat perbedaan yang terjadi cukup signifikan atau tidak, sehingga dapat dilakukan analisis lebih lanjut terkait penyebab perbedaan tersebut. Berikut adalah perintah yang digunakan untuk menampilkan lima cabang dengan total uang yang dicairkan paling kecil:

```
#Menampilkan data dari lima cabang yang memiliki total amount paling kecil

>library(scales)

>data_Mei %>%

>arrange(total_amount)%>%

>mutate(total_amount=  comma(total_amount))%>%    #untuk    menampilkan
total_amount dari kelima cabang tersebut

>head(5)
```

Output yang dihasilkan yaitu sebagai berikut:

```
# A tibble: 5 x 2
  cabang total_amount
  <chr>   <chr>
1 AV      30,280,000
2 AS      31,740,000
3 AT      34,840,000
4 AU      35,610,000
5 AO      39,120,000
```

Dapat terlihat bahwa cabang-cabang dengan total uang yang dicairkan paling kecil yaitu cabang AV, AS, AT, AU, AO. Terlihat pula bahwa terjadi perbedaan yang signifikan antara lima cabang yang memiliki total uang yang dicairkan paling besar dengan lima cabang yang memiliki total uang yang dicairkan paling kecil. Perbedaan tersebut mungkin disebabkan oleh umur cabang dikarenakan cabang bertambah setiap bulannya. Oleh karena itu, dalam menentukan performa cabang juga perlu diperhatikan umur cabang.

4. Menghitung umur masing-masing cabang dalam bulan.

Umur cabang dapat ditentukan dengan menghitung sudah berapa lama sejak tanggal cair pertama sampai dengan bulan Mei. Untuk memudahkan cara perhitungan umur dengan membagi jumlah selisih hari dengan 30, karena itu tanggal batas nya menggunakan tanggal tengah bulan (2020-05-15), agar tidak terlalu mempengaruhi presisi perhitungan. Perintah yang digunakan untuk menghitung umur cabang yaitu sebagai berikut:

```
#Menghitung umur cabang dalam bulan

>umur_cabang<- data1%>%

>group_by(cabang)%>%
```

```
>summarise(pertama_cair=min(tanggal_cair))%>%
>mutate(umur=as.numeric(as.Date('2020-05-15')-
as.Date(pertama_cair))%/% 30)
>umur_cabang
```

Diperoleh umur setiap cabang dalam bulan yaitu sebagai berikut:

```
# A tibble: 22 x 3
  cabang pertama_cair umur
  <chr>    <chr>      <dbl>
1 AA      2020-01-06      4
2 AB      2020-01-06      4
3 AC      2020-01-06      4
4 AD      2020-01-06      4
5 AE      2020-02-03      3
6 AF      2020-02-03      3
7 AG      2020-02-03      3
8 AH      2020-02-03      3
9 AI      2020-03-02      2
10 AJ      2020-03-02      2
# ... with 12 more rows
```

5. Melihat performa cabang berdasarkan umur.

Langkah berikutnya yaitu melihat performa cabang berdasarkan umur pada bulan terakhir yaitu Mei 2020. Sebelum dapat melihat performa cabang berdasarkan umur, terlebih dahulu akan digabungkan antara data bulan Mei dengan data umur cabang agar lebih mudah dalam menganalisis, dengan menggunakan perintah berikut:

```
#Menggabungkan data bulan Mei dan umur cabang
>data_umur_Mei <- umur_cabang %>%
```

```
>inner_join(data_Mei, by='cabang') #untuk menggabungkan data
>data_umur_Mei
```

Output yang diberikan yaitu:

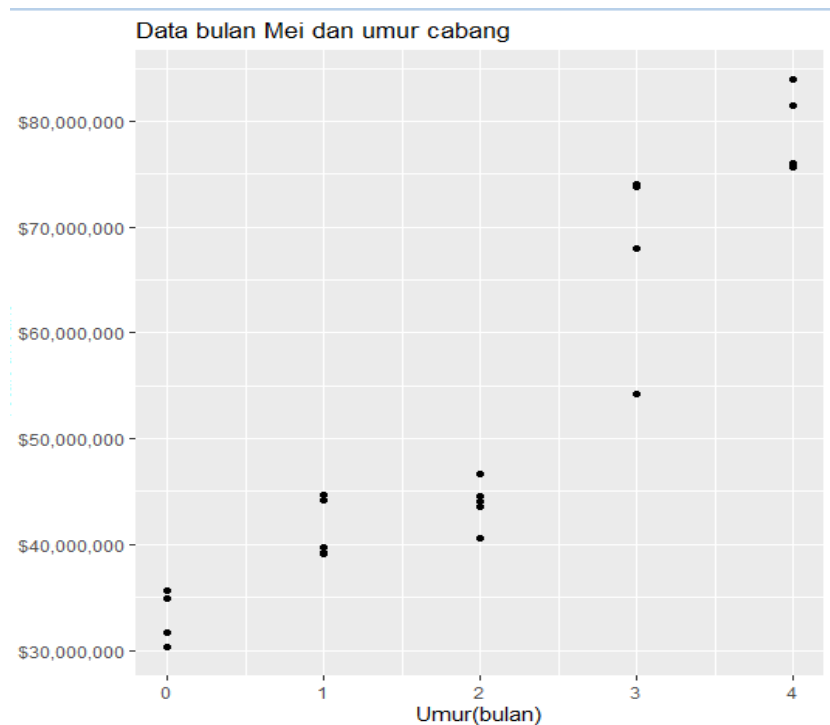
```
# A tibble: 22 x 4
  cabang pertama_cair umur total_amount
  <chr>   <chr>         <dbl>         <dbl>
1 AA     2020-01-06         4      75710000
2 AB     2020-01-06         4      81440000
3 AC     2020-01-06         4      83990000
4 AD     2020-01-06         4      76080000
5 AE     2020-02-03         3      54200000
6 AF     2020-02-03         3      68040000
7 AG     2020-02-03         3      74080000
8 AH     2020-02-03         3      73840000
9 AI     2020-03-02         2      46640000
10 AJ     2020-03-02         2      43580000
# ... with 12 more rows
```

Data kemudian akan divisualisasikan agar lebih mudah untuk dibaca dengan perintah berikut:

```
#Plot data gabungan dari umur cabang dan data bulan Mei
>library(ggplot2)
>ggplot(data_umur_Mei, aes(x= umur, y= total_amount)) +
>geom_point() + #untuk membuat scatterplot
>scale_y_continuous(labels=scales::dollar) + #agar format angkanya koma
>labs(title="Data bulan Mei dan umur cabang", #untuk menambahkan judul
```

```
>x="Umur(bulan)", #menambahkan caption pada sb x
>y="Total Amount") #menambahkan caption pada sb y
```

Plot yang dihasilkan adalah sebagai berikut:



Terlihat bahwa semakin tua cabang maka total uang yang dicairkan semakin besar, yang artinya performa cabang semakin baik. Cabang dengan umur yang lebih muda tentunya belum berkembang dengan maksimal dibandingkan dengan cabang dengan umur yang lebih tua. Oleh karena itu, yang perlu dianalisis adalah performa cabang dengan umur yang sama.

Performa cabang untuk masing-masing umur dapat dilihat dengan menggunakan perintah berikut:

```
#Melihat performa cabang berdasarkan umur
>data_Mei_performa<-data_umur_Mei%>%
>group_by(umur)%>%
>mutate(Q1=quantile(total_amount,0.25), #untuk menentukan kuartil 1 dari
total amount
>Q3=quantile(total_amount,0.75), #untuk menentukan kuartil 3 dari total
amount
>IQR=(Q3-Q1))%>% #untuk menentukan Inter Quartile Range
```

```
>mutate(performa=ifelse(total_amount<(Q1-IQR),'rendah','baik'))
>data_Mei_performa
```

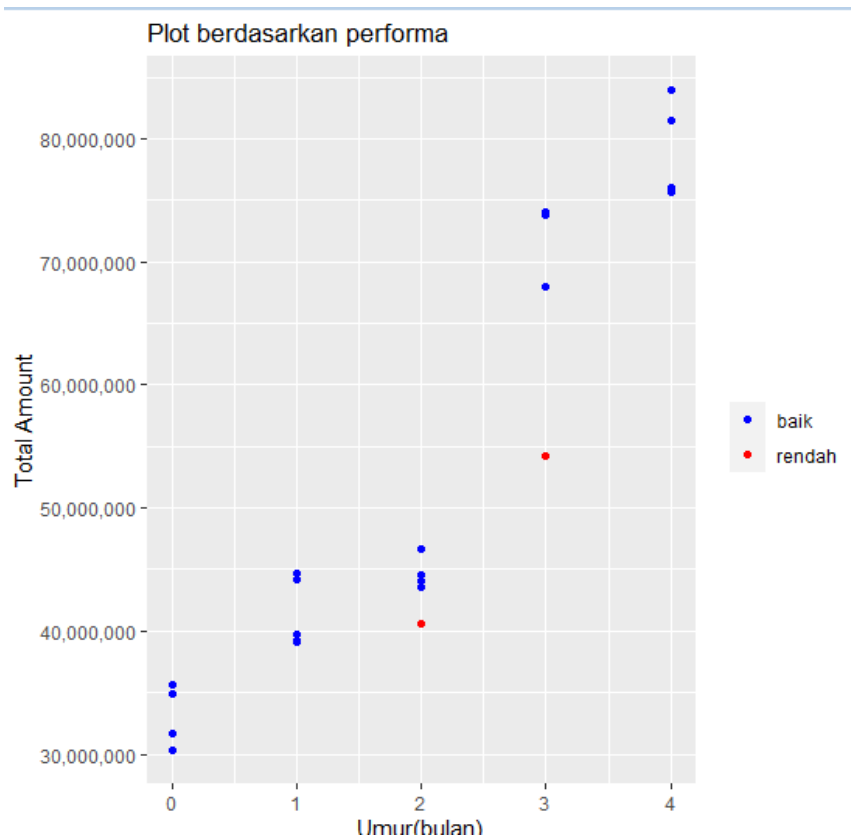
Output yang diberikan yaitu:

```
# A tibble: 22 x 8
# Groups:   umur [5]
  cabang pertama_cair umur total_amount      Q1      Q3      IQR performa
  <chr>   <chr>      <dbl>      <dbl>      <dbl>      <dbl>      <dbl> <chr>
1 AA     2020-01-06      4    75710000 75987500 82077500 6090000 baik
2 AB     2020-01-06      4    81440000 75987500 82077500 6090000 baik
3 AC     2020-01-06      4    83990000 75987500 82077500 6090000 baik
4 AD     2020-01-06      4    76080000 75987500 82077500 6090000 baik
5 AE     2020-02-03      3    54200000 64580000 73900000 9320000 rendah
6 AF     2020-02-03      3    68040000 64580000 73900000 9320000 baik
7 AG     2020-02-03      3    74080000 64580000 73900000 9320000 baik
8 AH     2020-02-03      3    73840000 64580000 73900000 9320000 baik
9 AI     2020-03-02      2    46640000 43580000 44590000 1010000 baik
10 AJ    2020-03-02      2    43580000 43580000 44590000 1010000 baik
# ... with 12 more rows
```

Dapat diperhatikan bahwa cabang dengan umur yang sama belum tentu memiliki performa yang sama, ini yang akan dianalisis yaitu alasan cabang tersebut performanya lebih rendah dibandingkan dengan cabang yang lain meskipun memiliki umur yang sama. Visualisasi data akan dilakukan untuk lebih memudahkan melihat performa cabang yaitu menggunakan perintah berikut:

```
#Plot data berdasarkan Perfoma cabang
>library(ggplot2)
>ggplot(data_Mei_performa, aes(x= umur, y= total_amount))+
>geom_point(aes(color=performa))+
>scale_color_manual(breaks=c("baik","rendah"),values=c("blue","red"))+
>scale_y_continuous(labels=scales::comma)+
>labs(title="Plot berdasarkan performa", color="",
>x="Umur(bulan)",
>y="Total Amount")
```

Plot yang dihasilkan yaitu:



Pada cabang dengan umur 2 bulan dan 3 bulan terdapat cabang dengan performa rendah. Akan dilakukan analisis terhadap cabang dengan performa rendah pada umur 2 bulan. Hal yang dilakukan adalah melihat performa masing-masing cabang pada umur 2 bulan terlebih dahulu dengan perintah berikut:

```
#Performa cabang di umur yang sama berdasarkan umur yang memiliki cabang
berperforma rendah

#Umur 2 bulan

>data_Mei_performa%>%
>filter(umur==2)%>%
>inner_join(data1,by='cabang')%>%
>filter(tanggal_cair>='2020-05-01',tanggal_cair<='2020-05-31')%>%
>group_by(cabang,performa)%>%
>summarise(jumlah_hari=n_distinct(tanggal_cair),
>agen_aktif=n_distinct(agen),
>total_loan_cair=n_distinct(loan_id),
>avg_amount=mean(amount),
>total_amount=sum(amount))%>%
```

```
>arrange(total_amount)%>%
>mutate_if(is.numeric,funs(comma))
```

Berdasarkan perintah tersebut dapat diperhatikan performa masing-masing cabang yang berumur 2 bulan yaitu sebagai berikut:

```
# A tibble: 5 x 7
# Groups:   cabang [5]
  cabang performa jumlah_hari agen_aktif total_loan_cair avg_amount total_amount
  <chr>   <chr>      <chr>      <chr>      <chr>      <chr>      <chr>
1 AL     rendah    21         2         134      303,358    40,650,000
2 AJ     baik     21         2         144      302,639    43,580,000
3 AM     baik     21         2         144      305,625    44,010,000
4 AK     baik     21         2         143      311,818    44,590,000
5 AI     baik     21         2         152      306,842    46,640,000
```

Terlihat bahwa yang berbeda jauh adalah total loan (pinjaman) yang cair, untuk melihat penyebab dari hal tersebut perlu dicari performa dari agen dengan cabang berperforma rendah dengan menggunakan perintah berikut:

```
# A tibble: 2 x 7
# Groups:   cabang [1]
  cabang agen jumlah_hari agen_aktif total_loan_cair avg_amount total_amount
  <chr>   <chr> <chr>      <chr>      <chr>      <chr>      <chr>
1 AL     AL-2  21         1         67      299,403    20,060,000
2 AL     AL-1  21         1         67      307,313    20,590,000
```

Langkah berikutnya adalah membandingkan antara performa agen pada cabang berperforma rendah dengan cabang berperforma paling baik. Cabang berumur 2 bulan yang berperforma paling baik adalah cabang AI. Perintah yang digunakan untuk mengetahui performa agen pada cabang AI adalah sebagai berikut:

```
#Performa agen pada cabang yang paling baik (2 BULAN)
>data_Mei_performa%>%
>filter(cabang=='AI')%>%
>inner_join(data1,by='cabang')%>%
>filter(tanggal_cair>='2020-05-01',tanggal_cair<='2020-05-31')%>%
>group_by(cabang,agen)%>%
>summarise(jumlah_hari=n_distinct(tanggal_cair),
```

```

>agen_aktif=n_distinct(agen),
>total_loan_cair=n_distinct(loan_id),
>avg_amount=mean(amount),
>total_amount=sum(amount))%>%
>arrange(total_amount)%>%
>mutate_if(is.numeric,funs(comma))

```

Output yang diberikan oleh perintah tersebut adalah:

```

# A tibble: 2 x 7
# Groups:   cabang [1]
  cabang agen jumlah_hari agen_aktif total_loan_cair avg_amount total_amount
<chr>  <chr> <chr>          <chr>          <chr>          <chr>          <chr>
1 AI    AI-1  21             1             75.0          292,933       21,970,000
2 AI    AI-2  21             1             77.0          320,390       24,670,000

```

Dapat disimpulkan bahwa yang jauh berbeda dari cabang berperforma rendah dan baik pada cabang yang berumur 2 bulan adalah total loan (pinjaman) yang cair.

Langkah berikutnya adalah analisis penyebab ada cabang yang berperforma rendah pada cabang dengan umur 3 bulan. Hal awal yang dilakukan adalah melihat performa masing-masing cabang dengan perintah sebagai berikut:

```

#Performa cabang di umur yang sama berdasarkan umur yang memiliki cabang
berperforma rendah

#Umur 3 bulan

>data_Mei_performa%>%
>filter(umur==3)%>%
>inner_join(data1,by='cabang')%>%
>filter(tanggal_cair>='2020-05-01',tanggal_cair<='2020-05-31')%>%
>group_by(cabang,performa)%>%
>summarise(jumlah_hari=n_distinct(tanggal_cair),
>agen_aktif=n_distinct(agen),
>total_loan_cair=n_distinct(loan_id),
>avg_amount=mean(amount),
>total_amount=sum(amount))%>%

```

```
>arrange(total_amount)%>%
>mutate_if(is.numeric,funs(comma))
```

Output yang diberikan adalah sebagai berikut:

```
# A tibble: 4 x 7
# Groups:   cabang [4]
  cabang performa jumlah_hari agen_aktif total_loan_cair avg_amount total_amount
  <chr>   <chr>      <chr>      <chr>      <chr>      <chr>      <chr>
1 AE     rendah    21         3         175      309,714    54,200,000
2 AF     baik     21         3         225      302,400    68,040,000
3 AH     baik     21         3         241      306,390    73,840,000
4 AG     baik     21         3         241      307,386    74,080,000
```

Performa agen dari cabang berperforma rendah ditentukan dengan perintah berikut:

```
#Performa agen pada cabang yang rendah
#umur 3 bulan
>data_Mei_performa%>%
>filter(umur==3,performa=='rendah')%>%
>inner_join(data1,by='cabang')%>%
>filter(tanggal_cair>='2020-05-01',tanggal_cair<='2020-05-31')%>%
>group_by(cabang,agen)%>%
>summarise(jumlah_hari=n_distinct(tanggal_cair),
>agen_aktif=n_distinct(agen),
>total_loan_cair=n_distinct(loan_id),
>avg_amount=mean(amount),
>total_amount=sum(amount))%>%
>arrange(total_amount)%>%
>mutate_if(is.numeric,funs(comma))
```

Performa agen cabang berperforma rendah adalah sebagai berikut:

```
# A tibble: 3 x 7
# Groups:   cabang [1]
  cabang agen jumlah_hari agen_aktif total_loan_cair avg_amount total_amount
  <chr> <chr> <chr>      <chr>      <chr>      <chr>      <chr>
```

1	AE	AE-3	4.0	1	16	310,625	4,970,000
2	AE	AE-2	18.0	1	73	320,274	23,380,000
3	AE	AE-1	21.0	1	86	300,581	25,850,000

Performa agen pada cabang berperforma paling baik ditentukan dengan perintah berikut:

```
#Performa agen pada cabang yang paling baik (3 BULAN)

>data_Mei_performa%>%
>filter(cabang=='AG')%>%
>inner_join(data1,by='cabang')%>%
>filter(tanggal_cair>='2020-05-01',tanggal_cair<='2020-05-31')%>%
>group_by(cabang,agen)%>%
>summarise(jumlah_hari=n_distinct(tanggal_cair),
>agen_aktif=n_distinct(agen),
>total_loan_cair=n_distinct(loan_id),
>avg_amount=mean(amount),
>total_amount=sum(amount))%>%
>arrange(total_amount)%>%
>mutate_if(is.numeric,funs(comma))
```

Diperoleh performa agen pada cabang berperforma paling baik sebagai berikut:

```
# A tibble: 3 x 7
# Groups:   cabang [1]
cabang agen jumlah_hari agen_aktif total_loan_cair avg_amount total_amount
<chr> <chr> <chr> <chr> <chr> <chr> <chr>
1 AG AG-3 19.0 1 74.0 308,243 22,810,000
2 AG AG-2 21.0 1 84.0 301,548 25,330,000
3 AG AG-1 21.0 1 83.0 312,530 25,940,000
```

Terlihat bahwa rendahnya performa dari cabang AE adalah karena salah satu agen yang melakukan pencairan hanya 4 hari dalam 1 bulan, padahal agen lain bisa aktif 21 hari.

4. KESIMPULAN

Berdasarkan analisis tersebut, dapat disimpulkan bahwa jumlah hari pencairan dan total pinjaman (loan) yang cair mempengaruhi performa cabang. Rendahnya performa dari cabang AL disebabkan karena total loan (pinjaman) yang cair jauh lebih rendah dibandingkan dengan cabang yang lain. Rendahnya performa dari cabang AE adalah karena salah satu agen yang melakukan pencairan hanya 4 hari dalam 1 bulan, padahal agen lain bisa aktif 21 hari. Hal ini membuat total amount dari agen tersebut hanya 20% dibandingkan agen yang lainnya. Sedangkan pada cabang AG, performanya sangat baik karena ketiga agen melakukan pencairan hampir / selalu setiap hari kerja. Dua orang full 21 hari dan satu orang 19 hari. Sehingga performanya terjaga dengan baik. Oleh karena itu, disarankan untuk selanjutnya semua agen melakukan pencairan hampir atau setiap hari dengan jumlah pencairan semaksimal mungkin.