# Analisis Peluang Kemenangan Klub English Premiere League dengan metode GLMM (Generelized Linear Mixed Model)

Muhammad Naufal Irham Ramdhani

13/5/2021

# Latar belakang

Pertandingan olahraga merupakan salah satu rekreasi yang disenanangi oleh banyak orang, baik tua maupun muda. Salah satu cabang olahraga dengan peminat terbanyak adalah sepak bola. Hampir setiap negara, memiliki liga sepak bola masing-masing. Namun, salah satu liga yang paling ditunggu dan paling kompetitif adalah liga sepakbola inggris yaitu *English Premiere League*.

Dalam sepakbola, banyak sekali pihak yang terlibat. Mulai dari para pendukung, pelatih, pemain, bahkan pebisnis pun dapat terlibat dalam dunia sepakbola. Dan sudah pasti, mereka menginginkan tim yang mereka dukung menang. Banyak faktor yang dapat menentukan apakah suatu tim akan menang atau kalah. Faktor-faktor tersebut dapat kita analisis untuk memprediksi hasil kemenangan. Disinilah letak pentingnya analisis data, tidak hanya untuk sepakbola namun seluruh cabang olahraga. Dengan menganalisis data olahraga, kita dapat mengetahui bagaimana performa tim saat di lapangan.

Pada sepakbola, apapun bisa terjadi. Tim yang sejak awal unggul dapat mempertahankan keunggalan dan memenangi pertandingan, namun bisa juga tim yang pada mulanya kalah ataupun sama kuat pada babak pertama, bisa jadi ketika pertandingan usai, tim tersebut dapat meraih kemenangan. Hal inilah yang akan dianalisis. Karena data hasil pertandingan sudah pasti bukanlah berdistribusi normal melainkan distribusi bernoulli. Maka akan diggunakan *Generelized Linear Model* untuk memprediksi hasil akhir suatu pertandingan dengan diketahui bagaimana keadaan pada babak pertama.

Data yang akan dianalisis adalah data pertandingan *English Premiere League* dari musim 2010-2011 sampai 2020-2021. Data ini bersumber dari: https://www.kaggle.com/pablohfreitas/all-premier-league-matches-20102021 (https://www.kaggle.com/pablohfreitas/all-premier-league-matches-20102021) Data tersebut, diambil dengan metode *web scraping* atau mengekstrak langsung data dari website resmi *English Premiere League*. Data berisi 4070 pertandingan, dengan kolom sebanyak 114 kolom. Kolom-kolom tersebut diantaranya berisi tentang hal-hal berikut, yaitu skor akhir, skor babak pertama, Tim yang bermain sebagai tuan rumah, dst. Data tersebut memiliki size yang tidak terlalu besar, sehingga tidak perlu waktu komputasi yang lama dalam menganalisis data tersebut.

Dengan analisis data ini, kiranya dapat pula memilih tim mana yang lebih memungkinkan untuk menang. Hal ini tidak hanya terbatas pada olahraga saja. Tapi juga diharapkan dengan analisis data yang serupa juga bisa digunakan dalam bidang lain, seperti mencari agensi periklanan mana yang memiliki kemungkinan sukses terbaik. Hal ini karena analisis data ini tidak memerlukan peluang sukses dari variabel yang bersangkutan untuk memilih variabel terbaik

# Metodologi

## Import Library

*Library* yang akan digunakan dalam analisis data ini, yakni:

```
library(lme4)
library(dplyr)
library(ggplot2)
library(pROC)
library(caret)
```

Adapun fungsi dari masing-masing *library* adalah sebagai berikut:

- lme 4 : Untuk melakukan analisis GLMM
- dplyr : Untuk mentransformasi *data set*
- ggplot : Untuk membuat berbagai macam plot
- pROC : Untuk membuat kurva ROC
- caret : Untuk membuat tabel kontingensi

# Import Dataset

```
Data = read.csv('D:/Memento/Project/df_full_premierleague.csv')
head(Data)[,2:5]
```

```
##                                link_match season       date         home_team
## 1 https://www.premierleague.com/match/7186  10/11 2010-11-01         Blackpool
## 2 https://www.premierleague.com/match/7404  10/11 2011-04-11         Liverpool
## 3 https://www.premierleague.com/match/7255  10/11 2010-12-13 Manchester United
## 4 https://www.premierleague.com/match/7126  10/11 2010-09-13        Stoke City
## 5 https://www.premierleague.com/match/7350  10/11 2011-02-14            Fulham
## 6 https://www.premierleague.com/match/7096  10/11 2010-08-16 Manchester United
```

Dari cuplikan *data set* diatas, kita bisa lihat sumber darimana data hasil pertandingan tersebut berasal. Ada banyak sekali kolom pada data set tersebut, namun hanya sebagian data saja yang akan digunakan. Maka dari itu, data set tersebut perlu dibersihkan

# Bersihkan dataset

Dari sekian banyak kolom yang ada, hanya beberapa saja yang digunakan, yaitu:

1. Nama tim yang bertanding (Kategorikal, dengan 37 level)
2. Menang atau tidak (Kategorikal, dengan 2 level)
3. Tuan rumah atau tamu (Kategorikal, dengan 2 level)
4. Lawan tanding (Kategorikal, dengan 37 level (klub))
5. Perbedaan goal dengan lawan (Diskrit, Bilangan bulat)

Jika diperhatikan, data set di atas sebenarnya menunjukkan data per pertandingan. Sedangkan, data yang akan dianalisis adalah data per tim. Dengan kata lain, terdapat data implisit yang perlu dibangun. Dibawah ini adalah proses untuk membangun data implisit tersebut:

```
##   club_name H_or_A Win Goal_diff               Lawan
## 2 Blackpool      H   1         1 West Bromwich Albion
## 3 Blackpool      H   0        -1             Chelsea
## 4 Blackpool      A   0        -1     Manchester City
## 5 Blackpool      A   1         0          Stoke City
## 6 Blackpool      A   1         1    Newcastle United
## 7 Blackpool      H   0         0          Aston Villa
```

Struktur dari data yang dihasilkan di atas adalah:

```
str(df)
```

```
## 'data.frame':    7682 obs. of  5 variables:
##  $ club_name: chr  "Blackpool" "Blackpool" "Blackpool" "Blackpool" ...
##  $ H_or_A   : chr  "H" "H" "A" "A" ...
##  $ Win      : num  1 0 0 1 1 0 0 1 0 0 ...
##  $ Goal_diff: num  1 -1 -1 0 1 0 0 1 0 -2 ...
##  $ Lawan    : chr  "West Bromwich Albion" "Chelsea" "Manchester City" "Stoke City" ...
```

Dari hasil keluaran R diatas, kolom “club_name”, “H_or_A”, “Win”, dan “Lawan” masih bertipe “character”. Agar dapat dianalisis, maka tipe data dari masing-masing kolom diubah menjadi “factor”

```
df$club_name = as.factor(df$club_name)
df$H_or_A    = as.factor(df$H_or_A)
df$Win       = as.factor(df$Win)
df$Lawan     = as.factor(df$Lawan)
```

Selanjutnya cek *summary* data tersebut

```
summary(df)
```

```
##              club_name    H_or_A   Win        Goal_diff
##  Manchester United: 386   A:3841   0:4785   Min.   :-5
##  Arsenal          : 385   H:3841   1:2897   1st Qu.:-1
##  Manchester City  : 385                     Median : 0
##  Everton          : 384                     Mean   : 0
##  Liverpool        : 384                     3rd Qu.: 1
##  Chelsea          : 383                     Max.   : 5
##  (Other)          :5375
##                Lawan
##  Manchester United: 386
##  Arsenal          : 385
##  Manchester City  : 385
##  Everton          : 384
##  Liverpool        : 384
##  Chelsea          : 383
##  (Other)          :5375
```

Akan kita ubah level dari variabel-variabel yang berupa data kategorikal

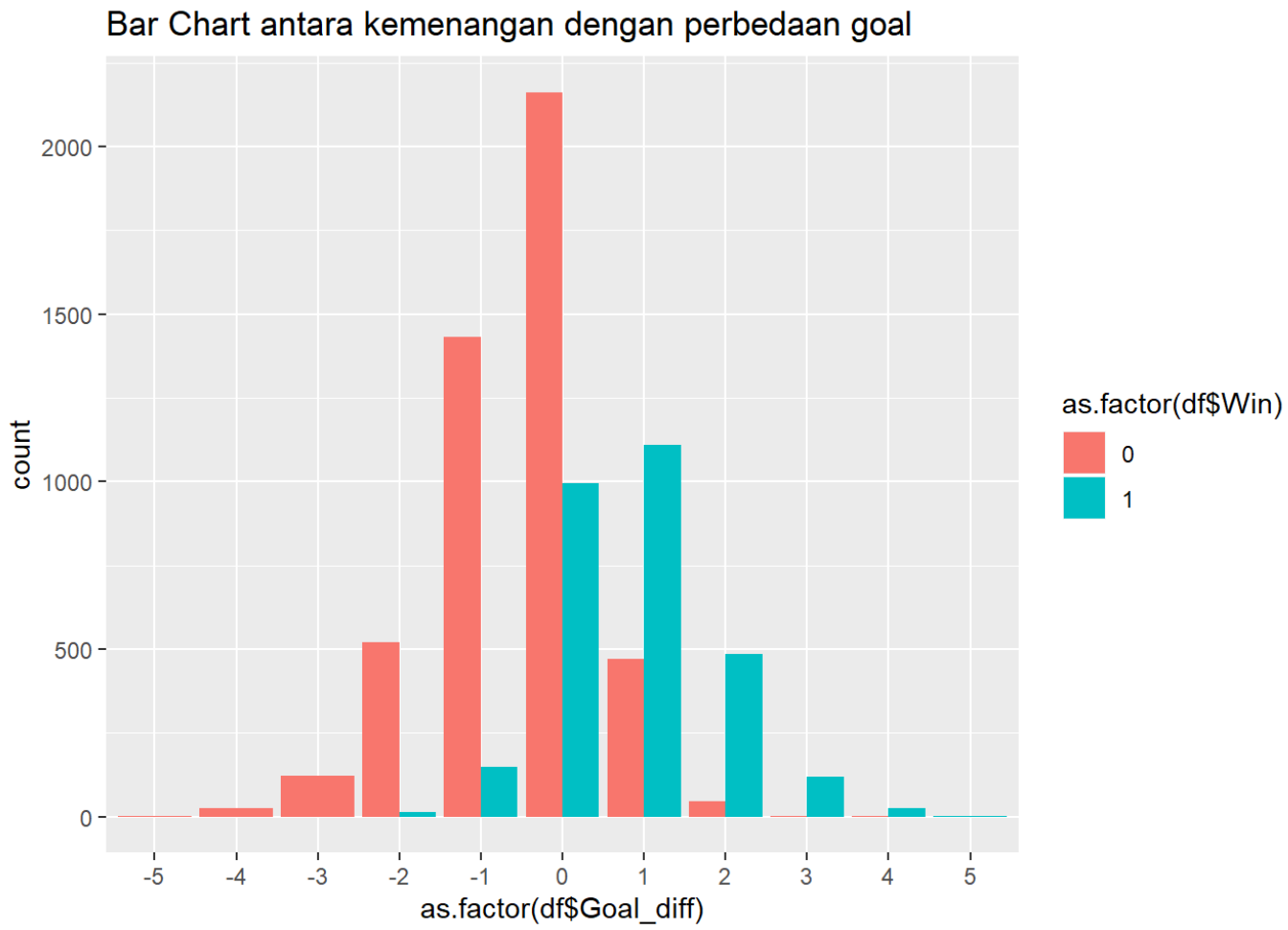
```
df = df %>% mutate(club_name = relevel(club_name, ref = "Manchester United"))
df = df %>% mutate(Win = relevel(Win, ref = "0"))
df = df %>% mutate(Lawan = relevel(Lawan, ref = "Manchester United"))
df = df %>% mutate(H_or_A = relevel(H_or_A, ref = "H"))

str(df)
```

```
## 'data.frame':    7682 obs. of  5 variables:
##  $ club_name: Factor w/ 37 levels "Manchester United",..: 7 7 7 7 7 7 7 7 7 7 ...
##  $ H_or_A   : Factor w/ 2 levels "H","A": 1 1 2 2 2 1 2 1 2 1 ...
##  $ Win      : Factor w/ 2 levels "0","1": 2 1 1 2 2 1 1 2 1 1 ...
##  $ Goal_diff: num  1 -1 -1 0 1 0 0 1 0 -2 ...
##  $ Lawan    : Factor w/ 37 levels "Manchester United",..: 34 12 21 29 23 4 35 8 34 36 ...
```

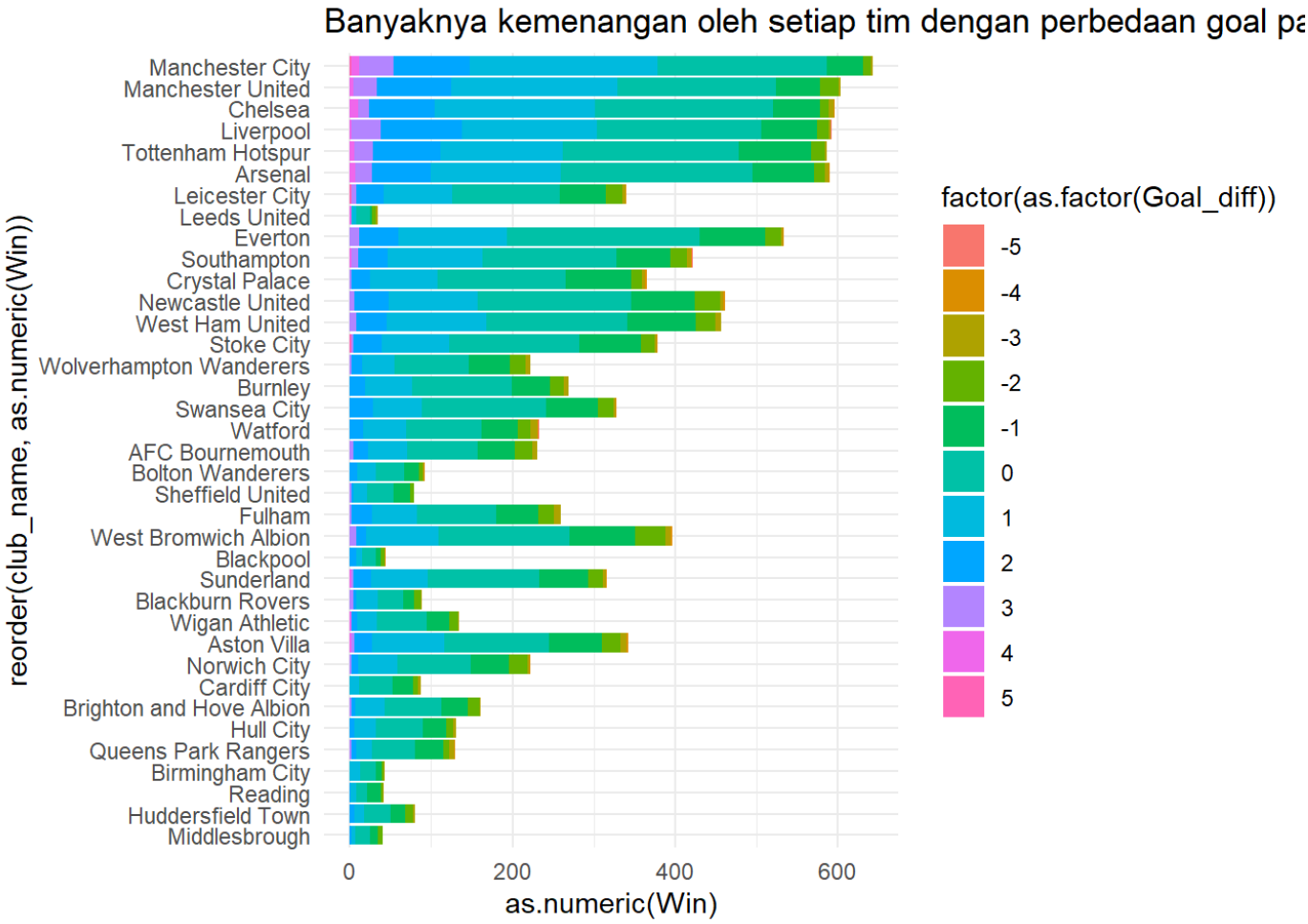
Data set tersebut dapat dijelaskan lebih jelas dengan menggunakan bar plot

```
ggplot(df, aes(x = as.factor(df$Goal_diff),
               fill = as.factor(df$Win))) +
  geom_bar(position = "dodge")+
  ggtitle("Bar Chart antara kemenangan dengan perbedaan goal")
```

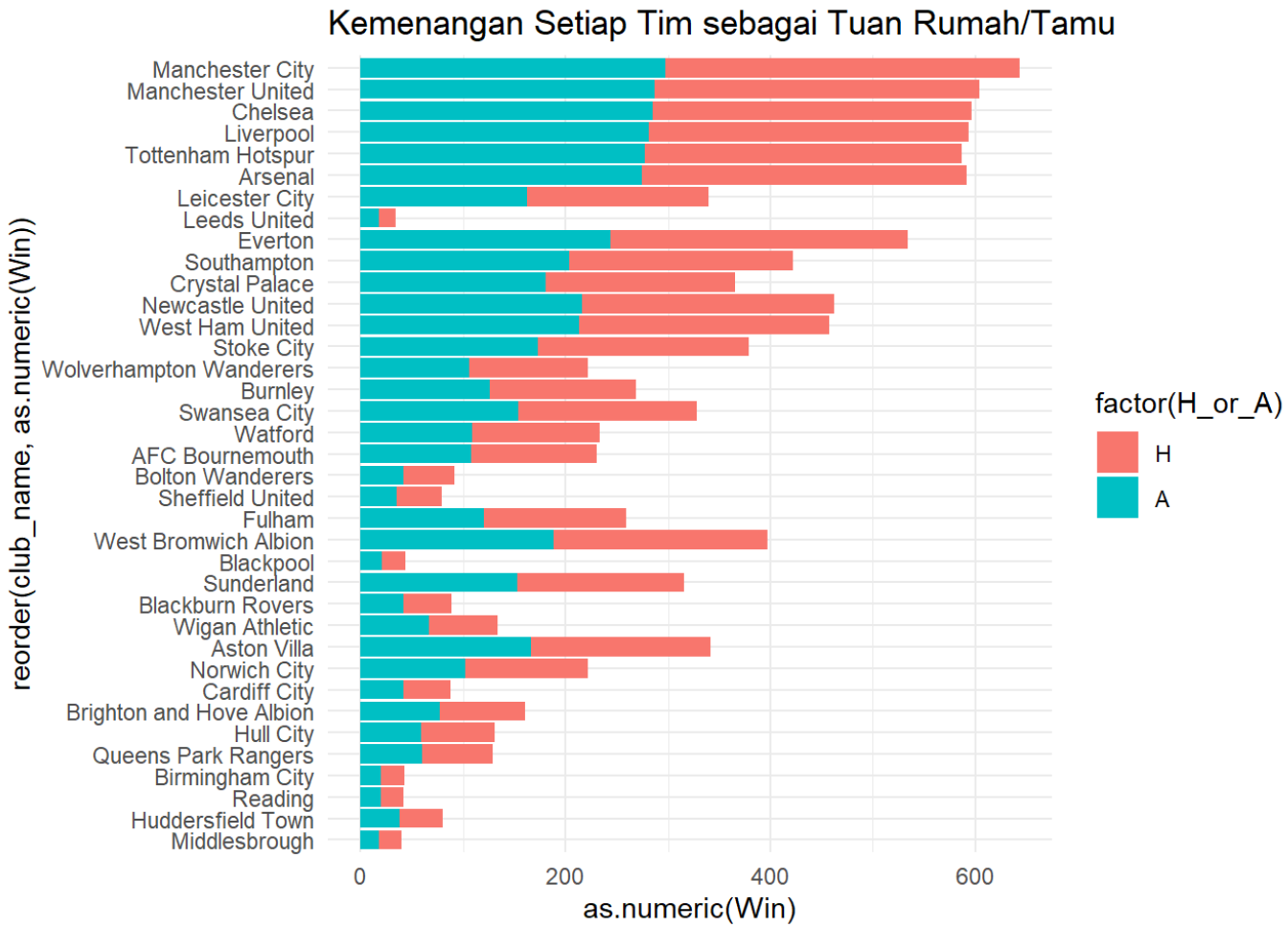



Dari plot diatas, sekilas terlihat bahwa lebih banyak klub yang unggul di babak pertama lalu memenangi pertandingan dibandingkan dengan tim yang bangkit dari defisit gol lalu menang. Lebih dalam lagi, akan ditunjukkan barplot kemenangan tiap tim saat berlaga sebagai tamu ataupun tuan rumah dan juga barplot kemenangan tiap tim untuk tiap perbedaan goal pada babak pertama yang pernah dialami

```
ggplot(df, aes(x = reorder(club_name, as.numeric(Win)),
               y = as.numeric(Win),
               fill = factor(as.factor(Goal_diff)))) +
  geom_bar(stat = "identity") +
  coord_flip() +
  theme_minimal() +
  ggtitle("Banyaknya kemenangan oleh setiap tim dengan perbedaan goal pada babak pertama")
```




```
ggplot(df, aes(x = reorder(club_name, as.numeric(Win)),
               y = as.numeric(Win),
               fill = factor(H_or_A))) +
  geom_bar(stat = "identity") +
  coord_flip() +
  theme_minimal() +
  ggtitle("Kemenangan Setiap Tim sebagai Tuan Rumah/Tamu")
```




Dari plot diatas, hasil suatu pertandingan dapat berkorelasi dengan klub yang bermain. Misal, klub yang kuat memiliki kemungkinan kemenangan yang lebih besar. Sedangkan klub yang lemah, memiliki kemungkinan kemenangan yang kecil. Dengan kata lain, setiap klub memiliki kemungkinan kemenangan yang berbeda-beda. Begitu pun juga hasil suatu kemenangan bergantung oleh lawan yang dihadapi. Lawan yang lebih kuat akan lebih sulit untuk dikalahkan, dan begitu juga sebaliknya. Maka dari itu, kita dapat membuat suatu *cluster*, dimana *cluster* tersebut berisikan klub-klub yang bermain di liga premier inggris. Atas dasar hal tersebut, maka model GLMM (Generelized Linear Mixed Models) adalah model yang tepat untuk menganalisis data ini karena variabel response memiliki korelasi.

Ditambah lagi, dari plot diatas, tiap klub memiliki jumlah kemenangan yang bervariasi untuk perbedaan goal tertentu. Selain itu, terlihat juga bahwa banyaknya kemenangan ketika bertanding sebagai tim tuan rumah ataupun tamu relatif sama. Namun, hal ini akan diteliti lebih lanjut.

Dalam anaslis data ini, Variabel respon dari data yang akan diolah adalah kemenangan suatu tim. Tim yang menang bernilai “1”, sedangkan tim yang kalah atau seri bernilai “0”. Maka dari itu, variabel respon memiliki distribusi bernoulli. *Link function* yang akan dipilih dalam menganalisis data ini adalah *logit link*. Seperti yang sudah dijelaskan sebelumnya, variabel penjelas yang akan digunakan dalam model adalah lawan, bermain sebagai tuan rumah atau tamu, dan lawan yang dihadapi

Akan dicoba memodelkan GLMM dengan *Random intercept model* yang secara umum memiliki bentuk sebagai berikut:

$$ln(\frac{\pi}{1-\pi}) = \alpha + X'\beta$$

dengan

$$y \sim B(1, \pi)$$

dan

$$\alpha \sim N(0, v^2)$$

# Analisis Data

Dalam menganalisis data, saya bagi terlebih dahulu membagi data menjadi dua, yang pertama untuk melatih model dan yang kedua untuk menguji model dan melakukan prediksi

```
n_train  = nrow(df)*0.75

train_df = sample_n(df, n_train)
test_df  = anti_join(df, train_df)
```

```
## Joining, by = c("club_name", "H_or_A", "Win", "Goal_diff", "Lawan")
```

Pada kode R diatas, data yang digunakan sebagai data untuk melatih model sebanyak 75% dari data keseluruhan, sedangkan 25% dari data sisanya dijadikan data untuk menguji prediksi.

Dibawah ini, akan dilakukan fitting model terhadap data *train*. Ada 3 model yang akan dicoba untuk fitting dengan data, yaitu:

1. Model1 <- Win ~ Goal_diff + H_or_A + (1 | club_name)
2. Model2 <- Win ~ Goal_diff + H_or_A + Lawan + (1 | club_name)
3. Model3 <- Win ~ Goal_diff + H_or_A + (1| Lawan) + (1 | club_name)
4. Model4 <- Win ~ Goal_diff + H_or_A + Lawan

Disini, ada 3 hal akan dibandingkan, yaitu:

1. Apakah model dengan *random effect* yang lebih simpel lebih baik daripada model dengan *random effect* yang lebih kompleks (membandingkan model 1 dan 2).
2. Apakah model dengan *random effect* lebih baik daripada model tanpa *random effect* (Memandngkan model 2 dan 4)
3. Apakah model variabel penjelas “Lawan” lebih baik dianggap sebagai *random effect* atau *fixed effect* (Memandngkan model 2 dan 3)

```
Model1 <- glmer(as.numeric(paste(Win)) ~ Goal_diff + H_or_A + (1 | club_name),
        data = train_df, family = binomial,
        control = glmerControl(),
        start = NULL)

Model2 <- glmer(as.numeric(paste(Win)) ~ Goal_diff + H_or_A + Lawan + (1 | club_name),
        data = train_df, family = binomial,
        control = glmerControl(),
        start = NULL)

Model3 <- glmer(as.numeric(paste(Win)) ~ Goal_diff + H_or_A + (1 | Lawan) + (1 | club_name),
        data = train_df, family = binomial,
        control = glmerControl(),
        start = NULL)

Model4 <- glm(as.numeric(paste(Win)) ~ Goal_diff + H_or_A + Lawan,
        data = train_df,
        family = binomial(link = "logit"))
```

Hasil AIC dari ketiga model tersebut adalah:

```
modelsummary1 = summary(Model1)
modelsummary2 = summary(Model2)
modelsummary3 = summary(Model3)
modelsummary4 = summary(Model4)

cat("AIC untuk Model 1:", modelsummary1$AICtab[1])
```

```
## AIC untuk Model 1: 5252.455
```

```
cat("AIC untuk Model 2:", modelsummary2$AICtab[1])
```

```
## AIC untuk Model 2: 5151.167
```

```
cat("AIC untuk Model 3:", modelsummary3$AICtab[1])
```

```
## AIC untuk Model 3: 5171.824
```

```
cat("AIC untuk Model 4:", modelsummary4$aic)
```

```
## AIC untuk Model 4: 5264.775
```

Dari hasil keluaran R diatas, didapat bahwa model kedua memiliki AIC terkecil, namun model kedua dan ketiga memiliki AIC yang tidak begitu berbeda jauh. Sedangkan model tanpa random effect memiliki AIC terbesar dibandingkan yang lain. Lebih lanjut lagi, kita dapat cari apakah random effect pada model signifikan atau tidak terhadap model tanpa random effect.

```
anova(Model2, Model4)
```

```
## Data: train_df
## Models:
## Model4: as.numeric(paste(Win)) ~ Goal_diff + H_or_A + Lawan
## Model2: as.numeric(paste(Win)) ~ Goal_diff + H_or_A + Lawan + (1 | club_name)
##        npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## Model4   39 5264.8 5524.5 -2593.4   5186.8
## Model2   40 5151.2 5417.5 -2535.6   5071.2 115.61  1  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Dari hasil keluaran R diatas, didapat bahwa p-value yang didapat kurang dari $\alpha = 0.01$ bahkan mendekati nol, sehingga dapat dikatakan bahwa model dengan *random effect* berbeda secara signifikan dengan model tanpa *random effect*. Dengan kata lain, *random effect* pada model pertama tidak dapat diabaikan.

Dibawah ini akan dihasilkan kurva ROC untuk masing-masing model sebagai perbandingan mana model yang terbaik dalam memprediksi

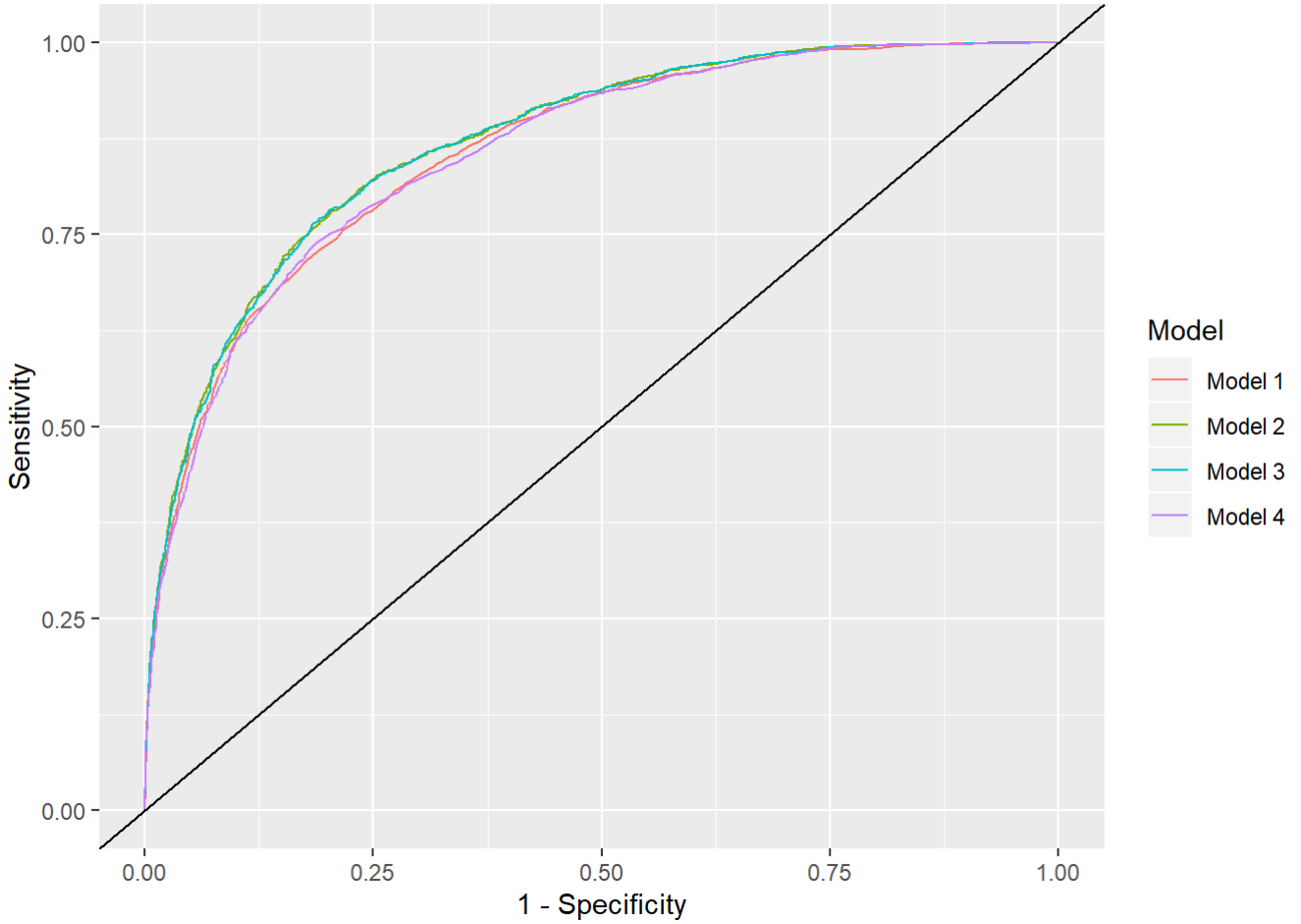
```
pred1 = predict(Model1, type = "response")
roccurve1 = roc(train_df$Win ~ pred1)

pred2 = predict(Model2, type = "response")
roccurve2 = roc(train_df$Win ~ pred2)

pred3 = predict(Model3, type = "response")
roccurve3 = roc(train_df$Win ~ pred3)

pred4 = predict(Model4, type = "response")
roccurve4 = roc(train_df$Win ~ pred4)
```

```
roclist = list("Model 1" = roccurve1,
               "Model 2" = roccurve2,
               "Model 3" = roccurve3,
               "Model 4" = roccurve4)
ggroc(roclist, aes = "colour", legacy.axes = T)+
  geom_abline(intercept = 0, slope = 1)+
  labs(x = "1 - Specificity",
       y = "Sensitivity",
       colour = "Model")
```




Secara sekilas, dapat dilihat bahwa kurva ROC model 2 dan 3 merupakan kurva yang terbaik, sedangkan kurva untuk model 1 dan model 4 terlihat tidak jauh berbeda. Secara lebih detail, akan dihitung luas dibawah kurva ROC untuk masing-masing model.

```
cat("Luas dibawah kurva ROC untuk Model 1 :", auc(roccurve1))
```

```
## Luas dibawah kurva ROC untuk Model 1 : 0.8586951
```

```
cat("Luas dibawah kurva ROC untuk Model 2 :", auc(roccurve2))
```

```
## Luas dibawah kurva ROC untuk Model 2 : 0.8698394
```

```
cat("Luas dibawah kurva ROC untuk Model 3 :", auc(roccurve3))
```

```
## Luas dibawah kurva ROC untuk Model 3 : 0.8694571
```

```
cat("Luas dibawah kurva ROC untuk Model 4 :", auc(roccurve4))
```

```
## Luas dibawah kurva ROC untuk Model 4 : 0.857007
```

Dari hasil luas kurva ROC diatas, terlihat bahwa model 2 dan 3 memiliki luas dibawah kurva ROC yang tidak berbeda jauh.Dari sini, Karena model 3 membutuhkan parameter yang lebih sedikit dibanding model 2 (lebih simpel), maka model 3 lah yang akan dipilih.

```
modelsummary3
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: as.numeric(paste(Win)) ~ Goal_diff + H_or_A + (1 | Lawan) + (1 |
##     club_name)
##    Data: train_df
## Control: glmerControl()
##
##      AIC      BIC   logLik deviance df.resid
##   5171.8   5205.1  -2580.9   5161.8     5756
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -20.4365  -0.5561  -0.2024   0.5028   8.7401
##
## Random effects:
##  Groups    Name        Variance Std.Dev.
##  Lawan     (Intercept) 0.1807   0.4251
##  club_name (Intercept) 0.2129   0.4614
## Number of obs: 5761, groups:  Lawan, 37; club_name, 37
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.48492    0.12111  -4.004 6.22e-05 ***
## Goal_diff    1.54202    0.04882  31.583  < 2e-16 ***
## H_or_AA     -0.59910    0.07024  -8.530  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##           (Intr) Gl_dff
## Goal_diff -0.080
## H_or_AA   -0.267 -0.017
```

Contoh interpretasi yang dapat diambil dari hasil fitting diatas adalah setiap kenaikan 1 goal pada perbedaan goal pada babak pertama akan meningkatkan *odds* tim tersebut untuk menang sebesar $352,96\%$ karena $e^{1.51064} = 4.5296$. Jika tim bertanding sebagai tim tamu, maka *odds* tim tersebut akan menang menurun sebesar $45,32\%$ karena $e^{-0.60368} = 0.5467957$ dibandingkan jika tim tersebut bermain sebagai tuan rumah.

Selanjutnya, akan dibangun tabel kontingensi dari model ketiga

```
pred = factor(ifelse(predict(Model3) < 0.1, 0, 1))
mat  = confusionMatrix(pred, as.factor(train_df$Win))
mat
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 3178  789
##          1  383 1411
##
##                Accuracy : 0.7966
##                  95% CI : (0.7859, 0.8069)
##     No Information Rate : 0.6181
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5533
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.8924
##             Specificity : 0.6414
##          Pos Pred Value : 0.8011
##          Neg Pred Value : 0.7865
##              Prevalence : 0.6181
##          Detection Rate : 0.5516
##    Detection Prevalence : 0.6886
##       Balanced Accuracy : 0.7669
##
##        'Positive' Class : 0
##
```

Dari tabel kontingensi diatas, ada beberapa hal dapat diambil, yaitu:

1. Tingkat akurasi model terhadap *train data* adalah 79.74% (Accuracy)
2. 89.24% Pertandingan yang berakhir seri atau kalah dapat diprediksi dengan benar (Sensitivity)
3. 64.36% Pertandingan yang berakhir kemenangana dapat diprediksi dengan (Specificity)
4. 80.21% Hasil prediksi kemenangan sesuai dengan data sesungguhnya
5. 78.71% Hasil prediksi kekalahan atau seri sesuai dengan data sesungguhnya
6. 61.81% data yang ada adalah pertandingan yang berakhir seri atau kalah(Prevalance)
7. 55.16% data yang ada dideteksi sebagai seri atau kekalahan (Detection Rate)
8. 68.77% data yang ada dideteksi sebagai seri atau kekalahan baik prediksinya salah ataupun benar (Detection Prevelance)
9. Rata-rata kebenaran model dalam meprediksi adalah 76.80%

# Prediksi

Cuplikan hasil prediksi model dengan data test adalah sebagai berikut

```
prob=round(predict(Model1, test_df, type='response'),2)

predtest=factor(ifelse(prob < 0.1 , 0, 1))

print(sample_n(data.frame(prob, FittedValue=predtest, test_df), 10))
```

```
##     prob FittedValue         club_name H_or_A Win Goal_diff
## 224 0.97           1   Manchester City      H   1         2
## 501 0.04           0       Aston Villa      A   0        -1
## 716 0.58           1      Swansea City      A   1         1
## 803 0.35           1       Southampton      H   0         0
## 5   0.61           1         Blackpool      H   1         1
## 6   0.01           0         Blackpool      H   0        -2
## 332 0.68           1   Birmingham City      H   1         1
## 379 0.39           1 Tottenham Hotspur      A   1         0
## 950 0.07           0           Burnley      A   0        -1
## 456 0.98           1   West Ham United      H   1         3
##                        Lawan
## 224 Brighton and Hove Albion
## 501                Liverpool
## 716           Crystal Palace
## 803                  Everton
## 5            Bolton Wanderers
## 6              Wigan Athletic
## 332                Sunderland
## 379           AFC Bournemouth
## 950                   Fulham
## 456              Southampton
```

```
confusionMatrix(predtest, as.factor(test_df$Win))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 243  22
##          1 430 430
##
##                Accuracy : 0.5982
##                  95% CI : (0.5689, 0.627)
##     No Information Rate : 0.5982
##     P-Value [Acc > NIR] : 0.5129
##
##                   Kappa : 0.2721
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.3611
##             Specificity : 0.9513
##          Pos Pred Value : 0.9170
##          Neg Pred Value : 0.5000
##              Prevalence : 0.5982
##          Detection Rate : 0.2160
##    Detection Prevalence : 0.2356
##       Balanced Accuracy : 0.6562
##
##        'Positive' Class : 0
##
```

Dari tabel kontingensi diatas, ada beberapa hal dapat diambil, yaitu:

1. Tingkat akurasi model terhadap *test data* adalah 69.05% (Accuracy)

2. 52.82% Pertandingan yang berakhir seri atau kalah dapat diprediksi dengan benar (Sensitivity)
3. 91.85% Pertandingan yang berakhir kemenangana dapat diprediksi dengan (Specificity)
4. 90.10% Hasil prediksi kemenangan sesuai dengan data sesungguhnya (PPV)
5. 58.07% Hasil prediksi kekalahan atau seri sesuai dengan data sesungguhnya (NPV)
6. 58.43% data yang ada adalah pertandingan yang berakhir seri atau kalah(Prevalance)
7. 30.87% data yang ada dideteksi sebagai seri atau kekalahan (Detection Rate)
8. 34.26% data yang ada dideteksi sebagai seri atau kekalahan baik prediksinya salah ataupun benar (Detection Prevelance)
9. Rata-rata kebenaran model dalam meprediksi adalah 72.33%

# Kesimpulan

1. Prediksi Kemungkinan suatu tim akan menang dapat diketahui dengan menggunakan model 1 + Tingkat akurasi model dengan *train data* adalah 79.31%
    - Tingkat akurasi model dengan *test data* adalah 69.05%
    - Cuplikan hasil predikisi ada pada tabel bagian prediksi

# Referensi

De jong, piet, dan Gillian Z. Heller. 2008. *Generalized linear models for insurance data*. Cambridge: Cambridge University Press.