Implement K-Nearest Neighbors algorithm on diabetes.csv dataset. Compute confusion matrix, accuracy, error rate, precision and recall on the given dataset. Dataset link : https://www.kaggle.com/datasets/abdallamahgoub/diabetes

In [17]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_score
import seaborn as sns
import matplotlib.pyplot as plt
```

In [18]:
```
# Load the dataset
df = pd.read_csv("C:/Users/samik/Downloads/diabetes (1).csv")
print(df.head(10))  # Inspect the first few rows
```

```
   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
0            6      148             72             35        0  33.6
1            1       85             66             29        0  26.6
2            8      183             64              0        0  23.3
3            1       89             66             23       94  28.1
4            0      137             40             35      168  43.1
5            5      116             74              0        0  25.6
6            3       78             50             32       88  31.0
7           10      115              0              0        0  35.3
8            2      197             70             45      543  30.5
9            8      125             96              0        0   0.0

   Pedigree  Age  Outcome
0     0.627   50        1
1     0.351   31        0
2     0.672   32        1
3     0.167   21        0
4     2.288   33        1
5     0.201   30        0
6     0.248   26        1
7     0.134   29        0
8     0.158   53        1
9     0.232   54        1
```

In [19]:
```
# Separate features and target variable
X = df.drop(columns=['Outcome'])
y = df['Outcome']
# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

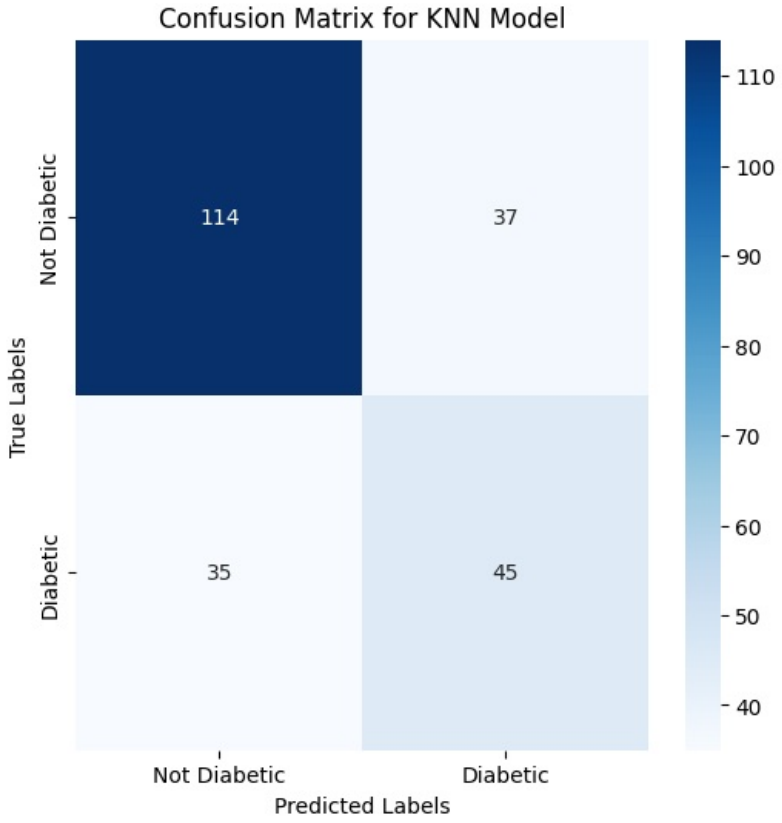
In [20]:
```
# Initialize and train the KNN model
knn = KNeighborsClassifier(n_neighbors=5)  # Default n_neighbors is 5; can be tuned
knn.fit(X_train, y_train)

# Make predictions on the test set
y_pred = knn.predict(X_test)
```

In [21]:
```
# Calculate performance metrics
conf_matrix = confusion_matrix(y_test, y_pred)
accuracy = accuracy_score(y_test, y_pred)
error_rate = 1 - accuracy
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
```

In [22]:
```
# Display results
print("Confusion Matrix:\n", conf_matrix)
print(f"Accuracy: {accuracy:.2f}")
print(f"Error Rate: {error_rate:.2f}")
print(f"Precision: {precision:.2f}")
print(f"Recall: {recall:.2f}")
```

```
Confusion Matrix:
 [[114  37]
 [ 35  45]]
Accuracy: 0.69
Error Rate: 0.31
Precision: 0.55
Recall: 0.56
```

In [23]:
```
# Plot confusion matrix as heatmap
plt.figure(figsize=(6, 6))
sns.heatmap(conf_matrix, annot=True, fmt="d", cmap="Blues", xticklabels=['Not Diabetic', 'Diabetic'], yticklabe
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")
plt.title("Confusion Matrix for KNN Model")
plt.show()
```



In [ ]: