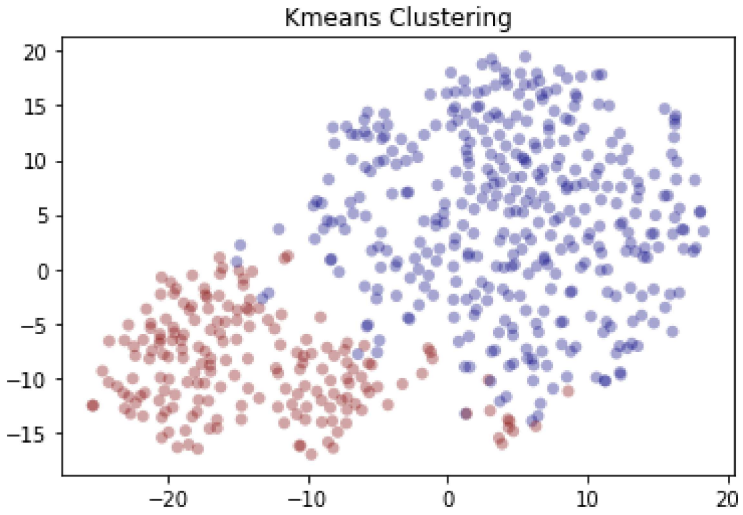
In [38]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import preprocessing
from sklearn.preprocessing import StandardScaler
from sklearn.manifold import TSNE
from sklearn.cluster import KMeans
df= pd.read_csv("C:/Users/shyma/OneDrive/Desktop/Machine learning/data.csv")
```

In [39]:

```
# Mapping Benign to 0 and Malignant to 1
df['diagnosis'] = df['diagnosis'].map({'M':1,'B':0})
# Scaling the dataset
datas = pd.DataFrame(preprocessing.scale(df.iloc[:,1:32]))
datas.columns = list(df.iloc[:,1:32].columns)
datas['diagnosis'] = df['diagnosis']
# Creating the high dimensional feature space X
data_drop = datas.drop('diagnosis',axis=1)
X = data_drop.values
#Creating a 2D visualization to visualize the clusters
tsne = TSNE(verbose=1, perplexity=40, n_iter= 4000)
Y = tsne.fit_transform(X)
kmns = KMeans(n_clusters=2, init='k-means++', n_init=50, max_iter=300,
tol=0.0001)
kY = kmns.fit_predict(X)
plt.scatter(Y[:,0],Y[:,1], c=kY, cmap = "jet", edgecolor = "None",
alpha=0.35)
plt.title("Kmeans Clustering")
plt.show()
```

```
[t-SNE] Computing 121 nearest neighbors...
[t-SNE] Indexed 569 samples in 0.000s...
[t-SNE] Computed neighbors for 569 samples in 0.010s...
[t-SNE] Computed conditional probabilities for sample 569 / 569
[t-SNE] Mean sigma: 1.522404
[t-SNE] KL divergence after 250 iterations with early exaggeration: 64.956306
[t-SNE] KL divergence after 2300 iterations: 0.868588
```



In [40]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import preprocessing
from sklearn.preprocessing import StandardScaler
from sklearn.manifold import TSNE
```

```
from sklearn_extra.cluster import KMedoids
df= pd.read_csv("C:/Users/shyma/OneDrive/Desktop/Machine learning/data.csv")
```

In [41]:

```
# Mapping Benign to 0 and Malignant to 1
df['diagnosis'] = df['diagnosis'].map({'M':1,'B':0})
# Scaling the dataset
datas = pd.DataFrame(preprocessing.scale(df.iloc[:,1:32]))
datas.columns = list(df.iloc[:,1:32].columns)
datas['diagnosis'] = df['diagnosis']
# Creating the high dimensional feature space X
data_drop = datas.drop('diagnosis',axis=1)
X = data_drop.values
#Creating a 2D visualization to visualize the clusters
tsne = TSNE(verbose=1, perplexity=40, n_iter= 4000)
Y = tsne.fit_transform(X)
kmedoids = KMedoids(n_clusters=2, random_state=0)
kY = kmedoids.fit_predict(X)
plt.scatter(Y[:,0],Y[:,1], c=kY, cmap = "jet", edgecolor = "None",
alpha=0.35)
plt.title("KMedoids Clustering")
plt.show()
```

```
[t-SNE] Computing 121 nearest neighbors...
[t-SNE] Indexed 569 samples in 0.000s...
[t-SNE] Computed neighbors for 569 samples in 0.034s...
[t-SNE] Computed conditional probabilities for sample 569 / 569
[t-SNE] Mean sigma: 1.522404
[t-SNE] KL divergence after 250 iterations with early exaggeration: 63.782005
[t-SNE] KL divergence after 1950 iterations: 0.878429
```