

# **LAPORAN FINAL PROJECT RSBP B**

## **Convert Classical Data to Quantum Data to implementation Machine Learning**



### **Penyusun**

Wendy Gata Ramadhani (502221162)

Ivan Fairuz Adinata (5025221167)

Gilang Kista Ramadhan (5025221173)

### **Pembimbing**

Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc Ph.D

Abdullah Faqih Septiyanto

**DEPARTEMEN TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS**

**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**

**SURABAYA**

**2024**

## DAFTAR ISI

<b>DAFTAR ISI.....</b>	<b>2</b>
<b>BAB I.....</b>	<b>3</b>
<b>Pendahuluan.....</b>	<b>3</b>
<b>BAB II.....</b>	<b>4</b>
<b>Solusi atau Metode yang diusulkan.....</b>	<b>4</b>
<b>BAB III.....</b>	<b>5</b>
<b>Eksperimen dan Hasil.....</b>	<b>5</b>
<b>Kesimpulan.....</b>	<b>19</b>

# BAB I

## Pendahuluan

Pembelajaran mesin telah mengalami perkembangan yang pesat sejak beberapa dekade terakhir dan telah terbukti memiliki banyak aplikasi di berbagai bidang kehidupan. Pembelajaran mesin sering dijelaskan sebagai cabang kecerdasan buatan (AI) yang berkonsentrasi pada merancang dan mengembangkan algoritma atau model yang memungkinkan komputer untuk belajar dari data dan membuat keputusan atau prediksi tanpa pemrograman yang eksplisit.

Pembelajaran mesin adalah bidang yang luas dengan aplikasi yang mencakup berbagai aktivitas seperti pengenalan pola dan analisis data hingga proses pengambilan keputusan otomatis. Di bidang ilmu komputer, contohnya mencakup sistem yang diperkuat pembelajaran mesin seperti sistem pengenalan wajah, asisten virtual, dan sistem rekomendasi konten. Di bidang kedokteran, teknologi ini membantu meningkatkan praktik di area seperti diagnosis penyakit, profil genetik, dan kustomisasi pengobatan untuk pasien. Bidang lainnya termasuk sektor keuangan di mana ada pemanfaatan aplikasi pembelajaran mesin untuk analisis risiko, deteksi penipuan, dan optimalisasi investasi.

Seiring dengan pertumbuhan jumlah data yang terus meningkat secara eksponensial bersama dengan kekuatan komputasi, ruang lingkup aplikasi pembelajaran mesin akan terus berkembang. Teknologi baru meningkatkan tidak hanya efisiensi dan akurasi proses tetapi juga memungkinkan peluang pengembangan baru di semua industri.

Namun, salah satu tantangan utama yang dihadapi saat ini adalah bagaimana memanfaatkan komputasi kuantum untuk lebih meningkatkan kinerja pembelajaran mesin, terutama dalam hal mengolah data klasik yang besar menjadi data kuantum yang dapat

diproses oleh algoritma kuantum. Komputasi kuantum menawarkan potensi besar untuk mempercepat algoritma tertentu dengan memanfaatkan prinsip-prinsip seperti superposisi dan entanglement.

Dalam laporan ini, kami akan membahas pendekatan untuk mengonversi data klasik ke data kuantum dan implementasi machine learning berbasis kuantum menggunakan teknologi seperti Qiskit. Fokus utama adalah untuk mengevaluasi efisiensi proses encoding data serta kinerja model kuantum dibandingkan model klasik.



## BAB II

### Solusi atau Metode yang diusulkan

Seiring dengan semakin berkembangnya penelitian di bidang machine learning, muncul pemikiran untuk menghubungkan konsep yang berasal dari mekanika kuantum ke dalam dunia machine learning dan disebut Quantum Machine Learning (QML). Quantum Machine Learning menjanjikan peluang baru yang dapat menyamai atau bahkan melebihi metode klasik dalam hal kecepatan dan efisiensi dalam aplikasi tertentu, terutama dataset besar dan kompleks.

Dalam hal ini, data klasikal akan ditransformasikan ke dalam data kuantum terlebih dahulu. Transformasi ini bertujuan agar data dapat dimanipulasi dengan memanfaatkan sifat-sifat tertentu dari mekanika kuantum seperti superposisi dan entanglement. Setelah sebagian besar dataset diubah dalam bentuk kuantum, machine learning dapat digunakan dengan memanfaatkan beberapa kelebihan quantum computing dalam proses pengolahan data yang tidak dapat dilakukan secara klasik. Namun, mengintegrasikan komputasi kuantum komputasi kuantum dengan pembelajaran mesin atau machine learning memiliki tantangan seperti stabilitas qubit, tingkat kesalahan, dan kebutuhan akan algoritma kuantum yang dapat sepenuhnya memanfaatkan perangkat keras kuantum. [1]

Di section selanjutnya, kami akan membahas proses yang berkaitan dengan transformasi dataset klasik menjadi dataset kuantum dan juga implementasi serta analisis hasil dari model Quantum Machine Learning yang digunakan. Adapun fokus penelitian ini adalah untuk menerapkan konsep *Quantum* pada metode-metode *machine learning*. Sebagai pembandingan, kami juga menerapkan beberapa metode machine learning pada data klasikal sebelum dikonversi, diantaranya RandomForestClassifier, K-Nearest Neighbor, Hyper Parameter Tuning. Pada data quantum, kami memakai SDK dari qiskit untuk menerapkan

machine learning. Beberapa metode milik qiskit machine learning yang kami gunakan di antara vqc variational quantum classifier VQC dan QSVC (Quantum Support Vector Classifier). VQC menggunakan qubits yang dapat berada dalam banyak keadaan secara simultan (prinsip superposisi) dan dapat menciptakan hubungan yang kompleks antar fitur menggunakan entanglement. [2]





## BAB III

### Eksperimen dan Hasil

Sebanyak dua dataset akan digunakan pada percobaan kali ini. Masing-masing metode machine learning pada klasikal data akan diterapkan terlebih dahulu pada setiap dataset. Selain model yang digunakan kita juga menggunakan *Feature Selection* dalam salah satu modelnya, dimana *Feature Selection* ini adalah proses untuk memilih hanya fitur fitur yang dianggap signifikan terhadap hasil yang akan diprediksi.

#### 3.1 Dataset breast-cancer-wisconsin-data

##### 3.1.1 Eksplorasi Data

Pada dataset pertama yaitu breast-cancer-wisconsin-data memiliki 569 baris data, dimana dataset tersebut memiliki 32 fitur, yaitu *diagnosis*, *radius\_mean*, *texture\_mean*, *perimeter\_mean*, *area\_mean*, *smoothness\_mean*, *compactness\_mean*, *concavity\_mean*, *concave points\_mean*, *symmetry\_mean*, *fractal\_dimension\_mean*, *radius\_se*, *texture\_se*, *perimeter\_se*, *area\_se*, *smoothness\_se*, *compactness\_se*, *concavity\_se*, *concave points\_se*, *symmetry\_se*, *fractal\_dimension\_se*, *radius\_worst*, *texture\_worst*, *perimeter\_worst*, *area\_worst*, *smoothness\_worst*, *compactness\_worst*, *concavity\_worst*, *concave points\_worst*, *symmetry\_worst*, *fractal\_dimension\_worst*.

Sementara, itu deskripsi dari dataset akan dipaparkan dalam tabel berikut

diagnosis	Diagnosis jaringan payudara dimana M melambangkan <i>Malignant</i> atau ganas, dan <i>benign</i> adalah jinak
-----------	---

radius_mean	Rata-rata dari titik tengah ke batas
texture_mean	seberapa banyak nilai intensitas piksel dalam gambar hitam-putih (grayscale) menyebar dari rata-rata intensitasnya
perimeter_mean	Rata-rata ukuran dari inti tumor
area_mean	luas area dari inti sel yang diukur dalam gambar mikroskopis jaringan payudara
smoothness_mean	rata-rata perubahan lokal dari panjang radius inti sel dalam gambar mikroskopis
compactness_mean	Rata-rata dipangkatkan 2 lalu dibagi luas dikurangi 1
concavity_mean	rata-rata tingkat keparahan cekungan pada kontur inti sel tumor
concave points_mean	rata-rata jumlah segmen cekung sepanjang kontur dalam sebuah gambar atau bentuk geometri
symmetry_mean	rata-rata simetri dari sel-sel kanker yang diamati
fractal_dimension_mean	rata-rata dimensi fraktal dari gambar sel atau tumor yang dianalisis

radius_se	standard error untuk rata-rata jarak dari pusat ke titik-titik di keliling
texture_se	Standar error untuk standar deviasi dari <i>gray-scale value</i>
perimeter_se	Standar error untuk sel tumor yang diamati
area_se	Kesalahan untuk pengukuran luas area dari sel tumor
smoothness_se	Kesalahan untuk panjang radius
compactness_se	Kesalahan standar untuk $\text{perimeter}^2 / \text{area} - 1$
concavity_se	Kesalahan dalam pengukuran tingkat keparahan dari cekungan
concave points_se	Kesalahan dalam penghitungan jumlah cekungan yang ada
symmetry_se	Kesalahan dalam pengukuran simetri dari sel tumor
fractal_dimension_se	ketidakpastian dalam pengukuran kompleksitas atau kehalusan bentuk tepi tumor
radius_worst	Rata-rata paling besar untuk rata-rata radius

	dari inti ke pinggir
texture_worst	Rata-rata paling besar untuk rata-rata standar deviasi dari nilai gray-scale
perimeter_worst	Seberapa besar atau rumit bentuk tumor pada sampel yang paling ekstrem
area_worst	Ukuran tumor yang paling ekstrem dalam sampel
smoothness_worst	Ketidakteraturan bentuk dari tumor yang paling tinggi
compactness_worst	Nilai rata-rata terburuk atau terbesar untuk $\text{perimeter}^2 / \text{area} - 1.0$
concavity_worst	Tingkat keparahan cekungan terburuk dari tumor
concave points_worst	Jumlah titik cekung terburuk pada kontur tumor yang dianalisis
symmetry_worst	Nilai simetri terburuk atau terbesar dari tumor yang dianalisis
fractal_dimension_worst	dimensi fraktal terburuk atau terbesar dari tumor yang dianalisis

### 3.1.2 Pra-pemrosesan Data

Dari data tersebut akan dilakukan pra pemrosesan berupa menghapus kolom "Unnamed: 32" dan "id" dihapus karena kolom "Unnamed: 32" merupakan kolom kosong yang muncul akibat kesalahan format CSV dengan koma tambahan di akhirnya, sedangkan kolom "id" tidak memiliki kontribusi analitis pada model. Selanjutnya, variabel target "diagnosis" diubah menjadi format biner, dimana nilai "M" dikonversi menjadi 1 dan "B" menjadi 0 untuk mempermudah analisis. Selain itu, fitur numerik dinormalisasi menggunakan metode standardisasi untuk memastikan distribusi data yang lebih seragam dan membantu algoritma pembelajaran mesin dalam konvergensi selama pelatihan. Dengan langkah-langkah ini, data telah siap digunakan untuk analisis dan pembangunan model selanjutnya.

### 3.1.3 Random Forest Classifier

Random Forest Classifier adalah algoritma untuk machine learning dengan basis ensemble, dan ensemble sendiri adalah penggabungan beberapa model untuk menjadi model baru yang lebih kuat. Ensemble ini digunakan untuk mengurangi variance dari model decision tree. Ensemble ini bekerja dengan cara membuat banyak model decision tree, setelah semua decision tree selesai untuk memprediksi maka akan diambil hasil dengan mayoritas untuk ditentukan menjadi hasil akhirnya.

Sebelum data dimasukkan ke dalam model Random Forest Classifier, data akan dibagi menjadi test dan juga train dengan proporsi yaitu 20% untuk test dan 80% untuk train menggunakan `train_test_split` dari library `sklearn.model_selection`. Setelah itu akan dipanggil model Random Forest Classifier dari `sklearn.ensemble` dengan `random_state 42`. Lalu akan dilatih menggunakan model yang sudah dipanggil. Setelah selesai dilatih maka akan digunakan untuk memprediksi berdasarkan data

yang sudah di split. Ternyata setelah selesai proses tersebut didapatkan hasil yaitu akurasi sebesar 96,49%, hasil ini sudah terhitung cukup tinggi. Namun ini belum membuktikan bahwa model tersebut bekerja dengan baik karena masih bisa saja terdapat faktor lain yang membuat model tersebut menampilkan akurasi yang baik.

#### 3.1.4 K-Nearest Neighbor

K-Nearest Neighbors (K-NN) adalah salah satu algoritma *machine learning* yang termasuk dalam kategori lazy learning dan instance-based learning. Algoritma ini digunakan untuk tugas klasifikasi dan regresi, namun lebih sering diterapkan dalam kasus klasifikasi. Algoritma ini bekerja dengan cara mengurutkan jarak dari titik ke tetangga yang paling dekat ke jauh. Setelah diurutkan maka akan diambil berapa tetangga yang terdekat dengan mendefinisikan K sebagai suatu angka. K ini dapat membuat suatu hasil *overfitting* ataupun *underfitting* tergantung besar kecil dari K tersebut. Jadi K tersebut harus berada di titik yang pas agar tidak mengalami *overfitting* dan *underfitting*.

Sama seperti sebelumnya sebelum data dimasukkan ke dalam model K-NN, data akan dibagi menjadi test dan juga train dengan proporsi yaitu 20% untuk test dan 80% untuk train menggunakan `train_test_split` dari library `sklearn.model_selection`. Setelah itu akan dipanggil model K-NN dari `sklearn.neighbors` dengan K adalah 3 yang berarti 3 terdekat yang akan dipakai. Lalu akan dilatih menggunakan model yang sudah dipanggil tersebut. Setelah selesai dilatih maka akan digunakan untuk memprediksi berdasarkan data yang sudah di split. Ternyata setelah selesai proses tersebut didapatkan hasil yaitu akurasi sebesar 92,98245614035088%.

Selain itu, kami juga menggunakan hyperparameter tuning untuk mendapatkan nilai *K* yang optimal. Hyperparameter tuning dilakukan dengan menggunakan

GridSearchCV dari library `sklearn.model_selection`. Proses ini melibatkan pencarian nilai K terbaik dengan mencoba beberapa kemungkinan nilai, seperti  $K = 1, 3, 5, 7, 9$ , dan mengukur performa model pada data validasi.

Dengan menggunakan hyperparameter tuning dapat menghindarkan adanya pemilihan nilai K yang sembarangan, sehingga model K-NN kami tidak mengalami *overfitting* atau *underfitting* setelah dilakukan proses tersebut maka didapatkan bahwa K yang paling tepat adalah 14 dengan akurasi 93.5010091600683%

### 3.2. Dataset DNA

#### 3.2.1 Eksplorasi Data

Dataset ini berisi 3 fitur yaitu *sgRNA\_sequence*, *target\_DNA\_sequence*, *off\_target\_activity*. Dimana fitur-fitur tersebut akan dijelaskan di tabel di bawah

off_target_activity	potensi sgRNA untuk mengikat urutan DNA lain yang mirip, tetapi bukan target utama
sgRNA_sequence	molekul RNA pendek yang dirancang untuk mengarahkan protein Cas9 ke lokasi target di genom
target_DNA_sequence	urutan DNA target yang dimaksudkan untuk dipotong atau diedit oleh kompleks CRISPR-Cas9

dataset ini memiliki keseluruhan data 584949

### 3.2.2. Pra-pemrosesan Data

Pra-pemrosesan data DNA dilakukan untuk memastikan data siap digunakan dalam model pembelajaran mesin. Dataset yang digunakan terdiri dari sekian ribu entri, mencakup variabel target *off\_target\_activity* dan fitur sekuens DNA. Data awal dianalisis menggunakan `info()` untuk memahami struktur dataset, termasuk tipe data dan keberadaan nilai yang hilang. Nilai-nilai hilang diidentifikasi dengan `isnull().sum()` dan ditangani dengan metode penghapusan atau imputasi agar tidak memengaruhi hasil analisis. Untuk merepresentasikan data DNA yang berupa urutan basa nukleotida (A, T, G, C), dilakukan ekstraksi k-mer, yaitu subsekuensi DNA dengan panjang tertentu (k) yang menangkap pola lokal dalam data. Representasi numerik k-mer dilakukan menggunakan metode seperti *CountVectorizer* atau *TfidfVectorizer*. Selanjutnya, dataset diseimbangkan untuk mengatasi ketidakseimbangan kelas pada variabel target menggunakan teknik menyamakan jumlah data target 0 dan 1 masing-masing sebanyak 7371 baris. Data numerik kemudian dinormalisasi menggunakan *StandardScaler* untuk memastikan semua fitur berada pada skala yang seragam, yang sangat penting bagi algoritma sensitif terhadap skala seperti *Support Vector Machines*. Akhirnya, dataset dibagi menjadi data latih dan data uji menggunakan `train_test_split` untuk mengevaluasi performa model secara akurat. Dengan total entri sebanyak 14742 yang digunakan, hasil pra pemrosesan ini memastikan representasi data optimal untuk pengembangan model pembelajaran mesin.

### 3.2.3. Logistic Regression

Pada model Logistic Regression ini pertama akan di bagi menjadi 2 data yaitu `x_train` dan `y_train`, serta `x_test` dan `y_test` dengan rasio 80% untuk train dan 20%



untuk testing. Setelah datanya terbagi maka akan dipanggil model Logistic Regression dengan iterasi maksimal adalah 1000. Setelah terpanggil maka akan dilatih data untuk train tadi dan didapatkan akurasi yaitu sebesar 77%, sensitivity 80%, dan juga f1-score adalah 78%.

#### 3.2.4. Random Forest Classifier

Sebelum data dimasukkan ke dalam model Random Forest Classifier, data akan dibagi menjadi test dan juga train dengan proporsi yaitu 20% untuk test dan 80% untuk train menggunakan `train_test_split` dari library `sklearn.model_selection`. Setelah itu akan dipanggil model Random Forest Classifier dari `sklearn.ensemble` dengan `random_state` 42. Lalu akan dilatih menggunakan model yang sudah dipanggil. Setelah selesai dilatih maka akan digunakan untuk memprediksi berdasarkan data yang sudah di split. Ternyata setelah selesai proses tersebut didapatkan hasil yaitu akurasi sebesar 83% dan sensitivity 84% serta 84% pada f1-score.

### 3.3. Konversi Data Klasikal ke Data Kuantum

#### 3.3.1. Dataset Cancer

Dalam pendekatan quantum machine learning, data klasik perlu dikonversi ke format yang sesuai dengan komputasi kuantum. Proses ini dimulai dengan memilih empat fitur terbaik berdasarkan korelasi tertinggi dengan variabel target. Contoh fitur yang dipilih meliputi *concave points\_mean*, *radius\_worst*, *perimeter\_worst*, dan *concave points\_worst*. Pemilihan fitur ini bertujuan untuk memastikan data dapat diproses oleh empat qubit yang akan digunakan dalam model quantum.

Setelah fitur dipilih, langkah selanjutnya adalah normalisasi menggunakan metode standarisasi untuk memastikan distribusi data yang seragam. Data yang telah

dinormalisasi kemudian dilakukan encoding ke dalam quantum state menggunakan metode amplitude encoding, sehingga setiap fitur direpresentasikan sebagai state quantum yang siap diproses oleh sirkuit quantum. Untuk memastikan data kompatibel dengan jumlah qubit yang digunakan, dimensi data disesuaikan dengan kebutuhan empat qubit.

### 3.3.2. Konversi Dataset DNA

Dalam pendekatan ini, konversi data klasik ke kuantum dilakukan dengan menggunakan representasi berbasis k-mer untuk data sekuensial, seperti DNA, sebelum encoding ke dalam state quantum. Proses dimulai dengan ekstraksi k-mer dari setiap urutan DNA untuk mengubah data menjadi fitur numerik. K-mer merupakan subsekuensi dengan panjang k yang digunakan untuk merepresentasikan pola-pola pada urutan DNA. Setelah fitur k-mer diperoleh, data diubah menjadi representasi vektor menggunakan metode seperti TF-IDF atau representasi frekuensi sederhana. Langkah selanjutnya adalah normalisasi data menggunakan standarisasi untuk memastikan distribusi yang seragam. Setelah itu, data akan dilakukan normalisasi dan encoding ke dalam quantum state menggunakan amplitude encoding, yang memastikan fitur numerik dapat direpresentasikan sebagai state quantum. Proses ini dioptimalkan untuk jumlah qubit yang tersedia dengan memastikan dimensi fitur cocok untuk representasi quantum. Dengan pendekatan ini, data klasik berbasis DNA telah dikonversi ke format quantum yang siap digunakan dalam model quantum. Proses ini memungkinkan model quantum untuk memanfaatkan pola dalam data sekuensial untuk tugas klasifikasi atau regresi.

### 3.4. Penerapan Model Quantum Machine Learning

Setelah dilakukan konversi menjadi dataset quantum, penerapan model quantum machine learning dapat dilakukan. Dalam percobaan kali ini, kami memakai tiga model qiskit machine learning, di antaranya VQC (Variational Quantum Classifier), QSVC (Quantum Support Vector Classifier), dan SVC(Support Vector Classifier). Ketiga model akan diterapkan ke masing-masing dataset, baik Dataset Breast Cancer Winconsin dan Dataset DNA.

#### 3.4.1. Penerapan Model VQC

Dalam percobaan ini, kami menggunakan Variational Quantum Classifier (VQC) untuk mengklasifikasikan dataset yang telah dikodekan ke dalam ruang Hilbert. Dataset dikonversi ke vektor angka real dan kemudian dipetakan ke sirkuit kuantum menggunakan *ZZFeatureMap*, yang berfungsi untuk menangkap hubungan non-linear antar fitur. Peta fitur ini dikombinasikan dengan *ansatz* berbasis *RealAmplitudes*, yang dirancang untuk menghasilkan sirkuit kuantum parametrik dengan jumlah pengulangan yang dapat disesuaikan melalui parameter *ansatz\_reps*. Optimasi parameter kuantum dilakukan menggunakan Simultaneous Perturbation Stochastic Approximation (SPSA). Kami juga melakukan grid search untuk menguji kombinasi hyperparameter, yaitu *feature\_map\_reps*, *ansatz\_reps*, dan jumlah iterasi maksimum SPSA (*maxiter*). Selama pelatihan, kami melacak nilai loss di setiap iterasi menggunakan callback untuk mengevaluasi konvergensi model.

Selanjutnya, model dievaluasi berdasarkan akurasi pada data uji, dengan hasil terbaik diperoleh dari kombinasi *feature\_map\_reps*=2, *ansatz\_reps*=2, dan *maxiter*=100 dengan akurasi 74.56% untuk Dataset Breast Cancer Winconsin serta *feature\_map\_reps* = 1, *ansatz\_reps* = 2, dan *maxiter* = 100 dengan akurasi 59.27%

untuk Dataset DNA. Model terbaik kemudian digunakan untuk prediksi, dengan hasil yang mencerminkan kemampuan generalisasi yang baik.

#### 3.4.2. Penerapan Model QSVC

Dalam eksperimen ini, kami juga menggunakan Quantum Support Vector Classifier (QSVC). Model QSVC dilatih dengan kernel kuantum seperti model VQC sebelumnya, dan hyperparameter utamanya, yaitu nilai regulasi  $C$  dan jumlah pengulangan pada feature map, diuji secara manual melalui grid search. Evaluasi model dilakukan dengan mengukur akurasi pada data uji untuk setiap kombinasi parameter, dan hasil terbaik diperoleh pada kombinasi  $C=1$  dan *feature\_map\_reps*=1, dengan akurasi uji sebesar 91.2% untuk Dataset Breast Cancer Winconsin, serta kombinasi  $C=10$ , dan *feature\_map\_reps*=1 dengan akurasi 56.66% untuk Dataset DNA. Model terbaik kemudian digunakan untuk melakukan prediksi, dan hasil prediksinya mencerminkan performa model yang optimal.

#### 3.4.3. Penerapan Model SVC

Dalam eksperimen ini, kami menggunakan Support Vector Classifier (SVC) dengan kernel kuantum berbasis *FidelityQuantumKernel* untuk mengklasifikasikan dataset yang telah dikodekan ke dalam ruang Hilbert. Dataset diubah menjadi vektor angka real, kemudian dimetakan ke sirkuit kuantum menggunakan ZZFeatureMap, dengan parameter pengulangan (*reps*) sebesar 2 untuk menangkap hubungan non-linear antar fitur. Kernel kuantum dihitung menggunakan *FidelityQuantumKernel*, yang menghasilkan matriks kernel berdasarkan kesamaan antara vektor fitur kuantum. Matriks kernel ini digunakan sebagai input untuk model SVC, dengan parameter kernel disetel ke 'precomputed'.

Proses tuning hyperparameter dilakukan menggunakan GridSearchCV, dengan nilai  $C$  (parameter regulasi) yang diuji mencakup 0.1, 1, 10, dan 100. Evaluasi

dilakukan menggunakan validasi silang 5-lipat (5-fold cross-validation). Hasil eksperimen menunjukkan bahwa kombinasi parameter terbaik dicapai pada  $C=1$ , dengan akurasi rata-rata validasi silang sebesar 92.74% pada dataset Breast Cancer Winconsin dan parameter terbaik  $C=100$  dengan akurasi 72% untuk Dataset DNA.

### 3.5. Tabel Hasil Percobaan

#### 3.5.1. Tabel Komparasi Perlakuan pada Dataset Breast Cancer Winconsin

Model	Best Parameter	Akurasi	Sensitivity	F-1 score
Random Forest Classifier	-	0.9649	-	-
K-Nearest Neighbor	-	0.9298	0.93	0.96
VQC	- <i>feature_map_reps=2</i> - <i>ansatz_reps=2</i> - <i>maxiter=100</i>	0.745614	-	-
QSVC	- $C=1$ - <i>feature_map_reps=1</i>	0.91228	-	-
SVC	- $C=1$	0.92747	-	-

#### 3.5.2. Tabel Komparasi Perlakuan pada Dataset DNA

Model	Best Parameter	Akurasi	Sensitivity	F-1 score
Logistic Regression	-	0.77	0.80	0.78
Random Forest Classifier	-	0.83	0.83	0.84
VQC	- <i>feature_map_reps=1</i> - <i>ansatz_reps=2</i> - <i>maxiter=10</i>	0.592743	-	-

	0			
QSVC	- $C=10$ - $feature\_map\_reps=1$	0.566633	-	-
SVC	- $C=100$	0.72	-	-

## **Kesimpulan**

Solusi dari permasalahan dari kebutuhan komputasi yang tinggi adalah penerapan komputasi kuantum di dalamnya. Dari permasalahan yang kelompok kami angkat, kami menerapkan konsep kuantum dengan melakukan konversi dari dataset klasikal menjadi kuantum, dimana nantinya kita dapat menerapkan model-model quantum machine learning pada dataset yang telah dikonversi. Dari penerapan kuantum, didapatkan bahwa model quantum memiliki potensi besar dalam meningkatkan efisiensi pengolahan data, terutama pada dataset dengan fitur yang kompleks dan dimensi tinggi. Model quantum seperti QSVC dan SVC menunjukkan performa yang mendekati model klasikal pada dataset Breast Cancer Wisconsin, namun mengalami penurunan performa pada dataset DNA. Hal ini menunjukkan bahwa pendekatan quantum lebih cocok untuk dataset tertentu, tergantung pada karakteristik data dan kompleksitas fitur.

Dengan hasil yang kami dapatkan dari percobaan yang telah dilakukan, penerapan model quantum machine learning kami masih jauh dari kata sempurna, dibuktikan dengan hasil akurasi dari model machine learning klasikal cenderung lebih baik dari pada penerapan model quantum machine learning. Hal ini dikarenakan integrasi quantum dan machine learning masih menghadapi tantangan teknis, seperti stabilitas qubit dan kebutuhan akan algoritma quantum yang lebih matang. Studi lanjutan diperlukan untuk mengeksplorasi pengoptimalan model quantum dan penerapannya pada dataset yang lebih besar dan lebih kompleks. Selain itu kedepannya harapannya mesin quantum dapat diakses secara lebih mudah sehingga dapat diperoleh data yang lebih konkret daripada menggunakan simulator.

## Referensi

- [1]. Anurag Dubey, Thulsiram Gantala, Anupama Ray, Anil Prabhakar, Prabhu Rajagopal, Quantum machine learning for recognition of defects in ultrasonic imaging, NDT & E International, Volume 150, 2025, 103262, ISSN 0963-8695, <https://doi.org/10.1016/j.ndteint.2024.103262> (<https://www.sciencedirect.com/science/article/pii/S0963869524002275>)
- [2] Muhammad Aasim, Ramazan Katırcı, Alpaslan Şevket Acar, Seyid Amjad Ali, A comparative and practical approach using quantum machine learning (QML) and support vector classifier (SVC) for Light emitting diodes mediated in vitro micropropagation of black mulberry (*Morus nigra* L.), Industrial Crops and Products, Volume 213, 2024, 118397, ISSN 0926-6690, <https://doi.org/10.1016/j.indcrop.2024.118397> (<https://www.sciencedirect.com/science/article/pii/S0926669024003741>)