

BAB IV

PENGEMBANGAN PERANGKAT LUNAK

4.1 Pendahuluan

Pada bab sebelumnya telah dijelaskan tahapan-tahapan penelitian yang akan dilakukan dan alat yang digunakan pada penelitian yaitu berupa perangkat lunak yang digunakan untuk mengetahui kinerja algoritma *k-Nearest Neighbor* dan *Modified k-Nearest Neighbor* pada klasifikasi *website phishing*. Oleh karena itu akan dikembangkan sebuah perangkat lunak dengan menggunakan konsep object oriented, metode yang digunakan pada pengembangan perangkat lunak yaitu metode *Rational Unified Proses* (RUP). Dalam metode RUP empat fase pemodelan yaitu *inception, elaboration, construction dan transition*.

4.2 Rational Unified Process (RUP)

4.2.1 Fase Inception

Dalam tahapan Rational Unified Process (RUP) fase *inception* merupakan fase pertama pada proses pengembangan preangkat lunak, pada fase ini akan dilakukan pemodelan bisnis dan menentukan scope proyek dari perangkat lunak yang akan dibangun dalam fase ini dilakukan proses pembuatan *use-case* diagram, menentukan *user, requirement*, menentukan kebutuhan fungsional dan non fungsional perangkat lunak, serta memastikan fungsionalitas pengujian pada perangkat lunak.

4.2.1.1 Pemodelan Bisnis

Phishing merupakan kejahatan internet yang relative umum dijumpai dengan memiliki tujuan untuk mencuri informasi pribadi dari pengguna atau hak akses pengguna. Pengguna internet dapat dengan mudah tertipu dikarenakan sebuah *website phishing* dibuat dengan replika yang sangat menyerupai dan terlihat sangat meyakinkan. Sehingga pada umumnya secara tidak disadari pengguna internet sangat sering menjumpai website yang terindikasi *phishing*, namun hal tersebut tidak mudah disadari oleh pengguna internet.

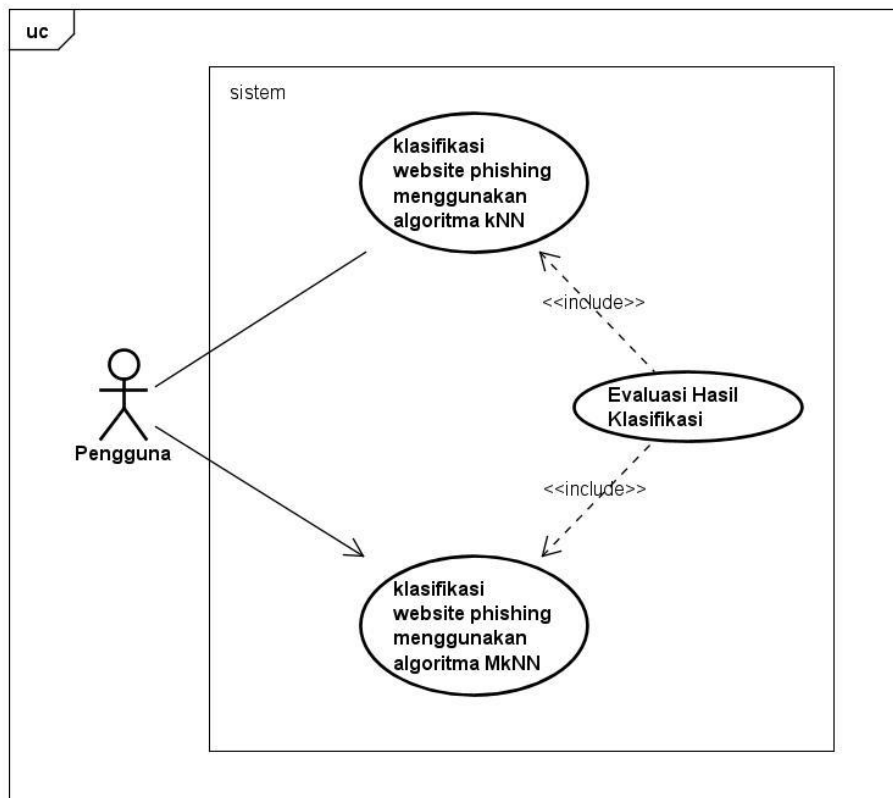
Dengan menggunakan teknik klasifikasi dapat dilakukan pengklasifikasian apakah sebuah website teridentifikasi memiliki aktivitas phishing didalamnya atau tidak. Klasifikasi sendiri merupakan teknik menemukan sebuah pola yang dapat menggambarkan konsep/kategori pada sebuah data. Sehingga dalam penelitian ini akan diterapkan teknik klasifikasi untuk mengklasifikasikan dataset *website phishing* dengan menggunakan algoritma *k-Nearest Neighbor*. Namun algoritma *k-Nearest Neighbor* sendiri terdapat kelemahan salah satunya yaitu pada tingkat akurasi, berdasarkan penelitian sebelumnya telah dilakukan pengembangan dari *k-Nearest Neighbor* yaitu *Modified k-Nearest Neighbor* yang dikembangkan untuk mengatasi kelemahan yang terdapat pada algoritma *k-Nearest Neighbor*.

Oleh karena itu perangkat lunak yang akan dikembangkan merupakan perangkat lunak berbasis desktop yang dapat melakukan klasifikasi data menggunakan dataset *website phishing* dengan algoritma *k-Nearest Neighbor* dan *Modified k-Nearest Neighbor*, serta dapat menghasilkan output berupa hasil kinerja dari algoritma tersebut untuk mengetahui kinerja terbaik antara kedua

lagoritma yang digunakan. Untuk mengetahui fungsionallitas dari perangkat lunak yang akan dibangun akan dijelaskan dalam use case diagram dibawah ini.

a. Diagram *use case*

Diagram *use case* merupakan diagram yang menggambarkan aktivitas-aktivitas dari *actor* terhadap sistem, digambarkan pada gambar IV-1.



Gambar IV-1. Use Case Diagram

b. Definisi *actor*

Actor merupakan salah satu komponen dari sebuah use case diagram, Penjelasan dari actor yang berperan pada perangkat lunak yang akan dikembangkan dapat dilihat pada table IV-1.

Table IV-1. Definisi actor

Nomor	Actor	Definisi
1	Pengguna	Pengguna merupakan actor yang memiliki akses terhadap perangkat lunak, dimana pengguna dapat memasukkan dataset serta melakukan pengklasifikasian terhadap data masukan dengan menggunakan algoritma k-Nearest Neighbor dan Modified k-Nearest Neighbor.

c. Table definisi *use case*

Use case dibuat berdasarkan keperluan actor yang merupakan interaksi yang dapat dilakukan oleh *actor* terhadap sistem. Penjelasan dari *use case* diagram yang gambarkan diatas dapat dilihat pada table IV-2.

Table IV-2. Definisi Use Case

Use Case	Deskripsi
Klasifikasi data menggunakan algoritma k-Nearest Neighbor	Sistem dapat melakukan pengklasifikasian terhadap data masukan dengan menggunakan algoritma k-Nearest Neighbor.
Klasifikasi data menggunakan algoritma Modified k-Nearest Neighbor	Sistem dapat melakukan pengklasifikasian terhadap data masukan dengan menggunakan algoritma Modified k-Nearest Neighbor.

Evaluasi hasil klasifikasi	Sistem mengevaluasi hasil klasifikasian dari algoritma menggunakan algoritma algoritma k-Nearest Neighbor ataupun Modified k-Nearest Neighbor.
----------------------------	--

4.2.1.2 Kebutuhan Sistem

Berdasarkan tahapan sebelumnya telah di gambarkan diagram usecase yang menggambarkan secara fungsionalitas dari perangkat lunak yang dibangun. Secara umum perangkat lunak yang dibangun harus memenuhi kebutuhan fungsional dan non fungsional. Kebutuhan fungsional merupakan kebutuhan utama sebuah sistem yang harus dipenuhi sehingga perangkat lunak dapat berfungsi atau dijalankan, untuk kebutuhan fungsional dapat dilihat pada table IV-3. Sedangkan kebutuhan non-fungsional merupakan kebutuhan-kebutuhan yang dapat mendukung kinerja dari sebuah sistem agar dapat berfungsi dengan baik, untuk kebutuhan non-fungsional dapat dilihat pada table IV-4.

Table IV-3. Kebutuhan fungsional.

No.	Fungsional
1	Perangkat lunak dapat mengimplementasikan algoritma <i>k-nearest neighbor</i> untuk melakukan klasifikasi website phishing.
2	Perangkat lunak dapat mengimplementasikan algoritma <i>modified k-nearest neighbor</i> untuk melakukan klasifikasi website phishing.

3.	Perangkat lunak dapat menghitung dan menyajikan hasil perhitungan kinerja algoritma <i>k-nearest neighbor</i> dan <i>modified k-nearest neighbor</i> untuk klasifikasi website phishing
----	---

Table IV-4. Kebutuhan non-fungsional.

No.	Non Fungsional
1	Perangkat lunak memiliki interface yang mudah dipahami dan mudah digunakan oleh pengguna
2	Perangkat lunak dapat memberikan feedback berupa pesan peringatan atau pemberitahuan saat pengguna melakukan kesalahan atau aktivitas diluar kemampuan yang telah ditentukan.

Berdasarkan kebutuhan fungsional dan non fungsional diatas maka perangkat lunak akan mengembangkan Fitur-fitur yang dapat memenuhi kebutuhan tersebut antara lain :

a. Fitur klasifikasi menggunakan algoritma *k-Nearest Neighbor*

Fitur klasifikasi menggunakan algoritma *k-Nearest Neighbor*, bertujuan untuk mengklasifikasikan data yang telah dimasukkan kedalam sistem dan akan diklasifikasikan menggunakan algoritma *k-Nearest Neighbor*. Urutan interaksi antara sistem dan actor terhadap fitur klasifikasi menggunakan algoritma *k-Nearest Neighbor* dapat dilihat pada scenario dibawah dan table IV-5.

No	:	001
Nama use case	:	Klasifikasi dataset <i>website phishing</i> menggunakan algoritma <i>k-Nearest Neighbor</i>
Actor	:	Pengguna
Tujuan	:	Memuat data kedalam sistem dan mengklasifikasikan data masukan menggunakan algoritma yang ditentukan
Deskripsi	:	Aktivitas pengklasifikasian data masukan dengan menggunakan algoritma <i>k-Nearest Neighbor</i> (kNN) dengan <i>k-fold cross validation</i> serta menampilkan hasil evaluasi dari algoritma kNN.
Kondisi awal	:	Pada sistem tidak terdapat data masukan
Kondisi akhir	:	Data masukan dimuat kedalam sistem dan menampilkan hasil dari evaluasi kualitas klasifikasi algoritma <i>k-Nearest Neighbor</i> .

Table IV-5. Skenario Klasifikasi menggunakan algoritma *k- Nearest Neighbor*

Actor	Sistem
1. Mengklik <i>button</i> “Muat”.	
	2. Menampilkan dialog pencarian berkas
3. Memilih data masukan yang sesuai	

	4. Menyampaikan pesan bahwa data berhasil dimuat.
5. Mengklik algoritma k-Nearest Neighbor pada <i>radio button</i>	
6. Menginputkan nilai parameter <i>k</i>	
7. Mengklik <i>button</i> “mulai”	
	8. Dengan teknik <i>k-fold cross validation</i> sistem membagi data menjadi data uji dan data latih.
	9. Sistem melakukan proses klasifikasi pada data masukan
	10. Sistem menampilkan hasil dari evaluasi kinerja algoritma berupa nilai precision, recall, akurasi, waktu komputasi dan memori yang dibutuhkan.
Skenario alternative 1 (jika data masukan dengan format yang tidak sesuai)	
1. Mengklik <i>button</i> “Muat”.	
	2. Menampilkan dialog pencarian berkas.
3. Memelilih data masukan dengan format data yang tidak sesuai.	

	4. Menyampaikan pesan bahwa format data salah.
Skenario alternatif 2 (jika nilai parameter k tidak diinputkan)	
1. Tidak menginputkan nilai parameter k	
	2. Sistem akan menampilkan pesan “masukkan format data yang benar!! Integer untuk parameter k”.
Skenario alternative 3 (jika nilai parameter k diinputkan tipe data yang salah)	
1. Menginputkan nilai parameter k dengan tipe data yang salah yaitu bukan bertipe integer.	
	2. Sistem akan menampilkan pesan “masukkan format data yang benar!! Integer untuk parameter k”.

b. Fitur klasifikasi menggunakan algoritma *Modified k-Nearest Neighbor*

Fitur klasifikasi menggunakan algoritma *Modified k-Nearest Neighbor*, bertujuan untuk mengklasifikasikan data yang telah dimasukkan kedalam sistem dan akan diklasifikasikan menggunakan algoritma *Modified k-Nearest Neighbor*. Urutan interaksi antara sistem dengan actor terhadap fitur klasifikasi menggunakan algoritma *Modified k-Nearest Neighbor* dapat dilihat pada scenario dibawah dan table IV-6.

No	:	002
Nama use case	:	Klasifikasi dataset <i>website phishing</i> menggunakan algoritma <i>Modified k-Nearest Neighbor</i>
Actor	:	Pengguna
Tujuan	:	Mengklasifikasikan data masukan menggunakan algoritma yang ditentukan
Deskripsi	:	Aktivitas pengklasifikasian data masukan menggunakan algoritma <i>Modified k-Nearest Neighbor</i> (MkNN) dengan <i>k-fold cross validation</i> serta menampilkan hasil evaluasi algoritma MkNN.
Kondisi awal	:	Pada sistem tidak terdapat data masukan.
Kondisi akhir	:	Data masukan dimuat kedalam sistem dan menampilkan hasil evaluasi kualitas klasifikasi algoritma

Table IV-6. Skenario klasifikasi menggunakan algoritma *Modified k- Nearest Neighbor*.

Actor	Sistem
1. Mengklik <i>button</i> "Muat".	
	2. Menampilkan dialog pencarian berkas

3. Memilih data masukan yang sesuai	
	4. Menyampaikan pesan bahwa data berhasil dimuat.
5. Mengklik algoritma Modified k-Nearest Neighbor pada <i>radio button</i> .	
6. Menginputkan nilai parameter <i>k</i>	
7. Mengklik <i>button</i> “mulai”	
	8. Dengan teknik k fold cross validation sistem membagi data menjadi data latih dan data uji.
	9. Sistem melakukan proses kalsifikasi pada data masukan
	10. Sistem menampilkan hasil dari evaluasi kinerja algoritma berupa nilai <i>precision</i> , <i>recall</i> , akurasi,, waktu komputasi dan memori yang dibutuhkan.
Skenario alternative 1 (jika data masukan dengan format yang tidak sesuai)	
1. Mengklik <i>button</i> “Muat”.	
	2. Menampilkan dialog pencarian berkas.
3. Memelilih data masukan dengan format data yang tidak sesuai.	

	4. Menyampaikan pesan bahwa format data salah.
Skenario alternative 2 (jika nilai parameter k tidak diinputkan)	
1. Tidak menginputkan nilai parameter k	
	2. Sistem akan menampilkan pesan “Masukan format data yang benar!!! Integer untuk parameter k ”.
Skenario alternative 3 (jika nilai parameter k diinputkan tipe data yang salah)	
1. Menginputkan nilai parameter k dengan tipe data yang salah yaitu bukan bertipe integer.	
	2. Sistem akan menampilkan pesan “Masukan format data yang benar!!! Integer untuk parameter k ”.

c. Fitur evaluasi hasil klasifikasi

Fitur evaluasi hasil klasifikasi bertujuan mengevaluasi kinerja dari algoritma k -Nearest Neighbor dan Modified k -Nearest Neighbor pada pengklasifikasian data, pada penelitian ini klasifikasi data menggunakan dataset website phishing, adapun kinerja dari algoritma tersebut berupa nilai dari

akurasi, recall, precision, waktu komputasi dan besarnya memori yang dibutuhkan sistem dalam mengklasifikasikan data masukan. Urutan interaksi antar sistem dengan actor terhadap fitur evaluasi hasil klasifikasi dapat dilihat pada table IV-7.

Table IV-7. Skenario evaluasi hasil klasifikasi.

Actor	Sistem
1. Mengklik <i>button</i> “Mulai”	
	2. Dengan teknik <i>k-fold cross validation</i> sistem membagi data menjadi data latih dan data uji.
	3. Sistem melakukan proses klasifikasi pada data masukan
	4. Sistem menampilkan hasil dari evaluasi kinerja algoritma berupa nilai precision, recall, akurasi, waktu komputasi dan besarnya memori yang dibutuhkan.

4.2.1.3 Analisis dan desain

Tahapan analisis dan desain pada fase insepisi akan menganalisis proses-proses pada perangkat lunak yang akan dibangun yaitu analisis klasifikasi dan analisis hasil klasifikasi. Tujuan dari penelitian ini untuk membandingkan kinerja

kedua algoritma dalam mengklasifikasikan dataset *website phishing* menggunakan algoritma *k-Nearest Neighbor* dan *Modified k-Nearest Neighbor*.

a. Analisis algoritma klasifikasi

Proses klasifikasi pada penelitian ini menggunakan algoritma *k-Nearest Neighbor* (kNN) dan *Modified k-Nearest Neighbor* (MkNN) dengan menggunakan dataset *website phishing*, data tersebut akan di proses menggunakan *metode k-fold cross validation* dimana nilai $k=10$. Dengan 10 *fold cross validation* maka akan menghasilkan 10 subset data dengan ukuran yang sama, dari 10 *fold* tersebut cross validation akan menggunakan 9 *fold* data menjadi data latih dan 1 *fold* data menjadi data uji yang bergantian sebanyak 10 subset data tersebut.

Algoritma *k-Nearest Neighbor* (kNN) dan *Modified k-Nearest Neighbor* (MkNN) penentuan nilai parameter k sangat penting untuk kedua algoritma, karena nilai k adalah jumlah tetangga terdekat yang diinputkan secara manual pada sistem oleh pengguna. Pada penelitian ini untuk mengetahui nilai k terbaik untuk kedua algoritma, sebagai ruang solusi peneliti menggunakan nilai k yang berbeda-beda yaitu $k=3$, $k=5$, $k=7$ dan $k=9$. Pada dasarnya perhitungan algoritma *Modified k-Nearest Neighbor* hampir sama dengan *algoritma k-Nearest Neighbor* dikarenakan, MkNN merupakan hasil dari pengembangan algoritma kNN. Untuk proses penghitungan menggunakan algoritma *k-Nearest Neighbor* dan *Modified k-Nearest Neighbor* dapat dilihat kembali pada bab 2 laporan ini.

b. Analisis kinerja algoritma klasifikasi

Analisis dilakukan untuk mengetahui kinerja dari algoritma klasifikasi *k-Nearest Neighbor* dan *Modified k-Nearest Neighbor* terhadap hasil kinerja algoritma pada proses pengklasifikasian data menggunakan dataset *website phishing*. Kinerja dari algoritma yang akan dianalisis berupa nilai *precision*, *recall* dan *accuracy* yang akan diperoleh dengan melakukan perhitungan *confusion matrix*. Selain itu juga akan diketahui waktu komputasi dari algoritma dan besarnya memori yang digunakan algoritma untuk melakukan pengklasifikasian data menggunakan algoritma *k-Nearest Neighbor* dan *Modified k-Nearest Neighbor*

4.2.2 Fase Elaboration

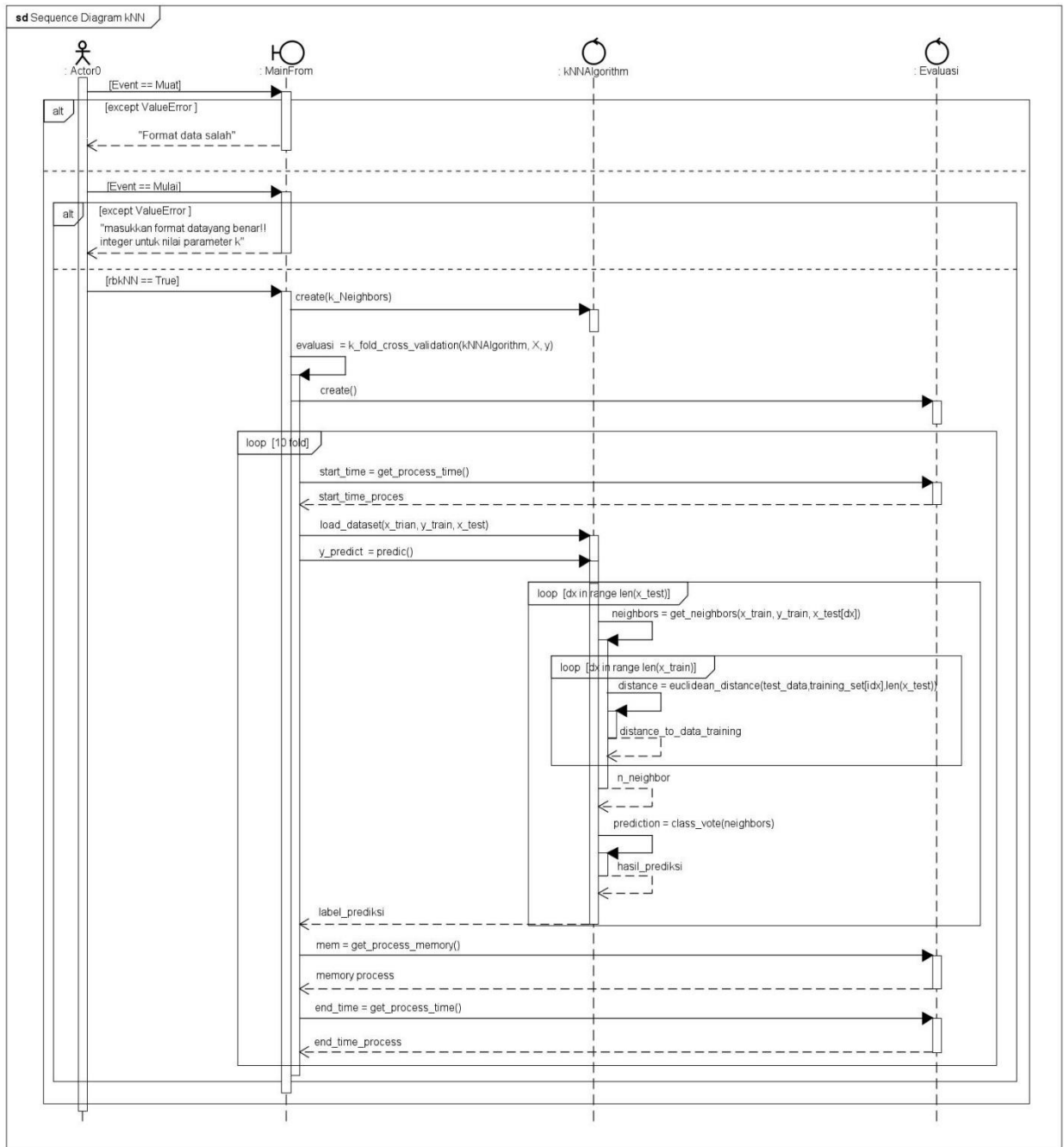
Fase *elaboration* pada *Rational Unified Process* (RUP) merupakan fase kedua dari pengembangan perangkat lunak, pada fase ini akan dilakukan identifikasi terhadap kebutuhan perangkat lunak yang akan dikembangkan. Fase *elaboration* terdiri dari aktivitas membuat antar muka perangkat lunak, pemodelan dalam bentuk diagram sequence dan diagram aktivitas, serta dokumentasi.

4.2.2.1 Pemodelan bisnis

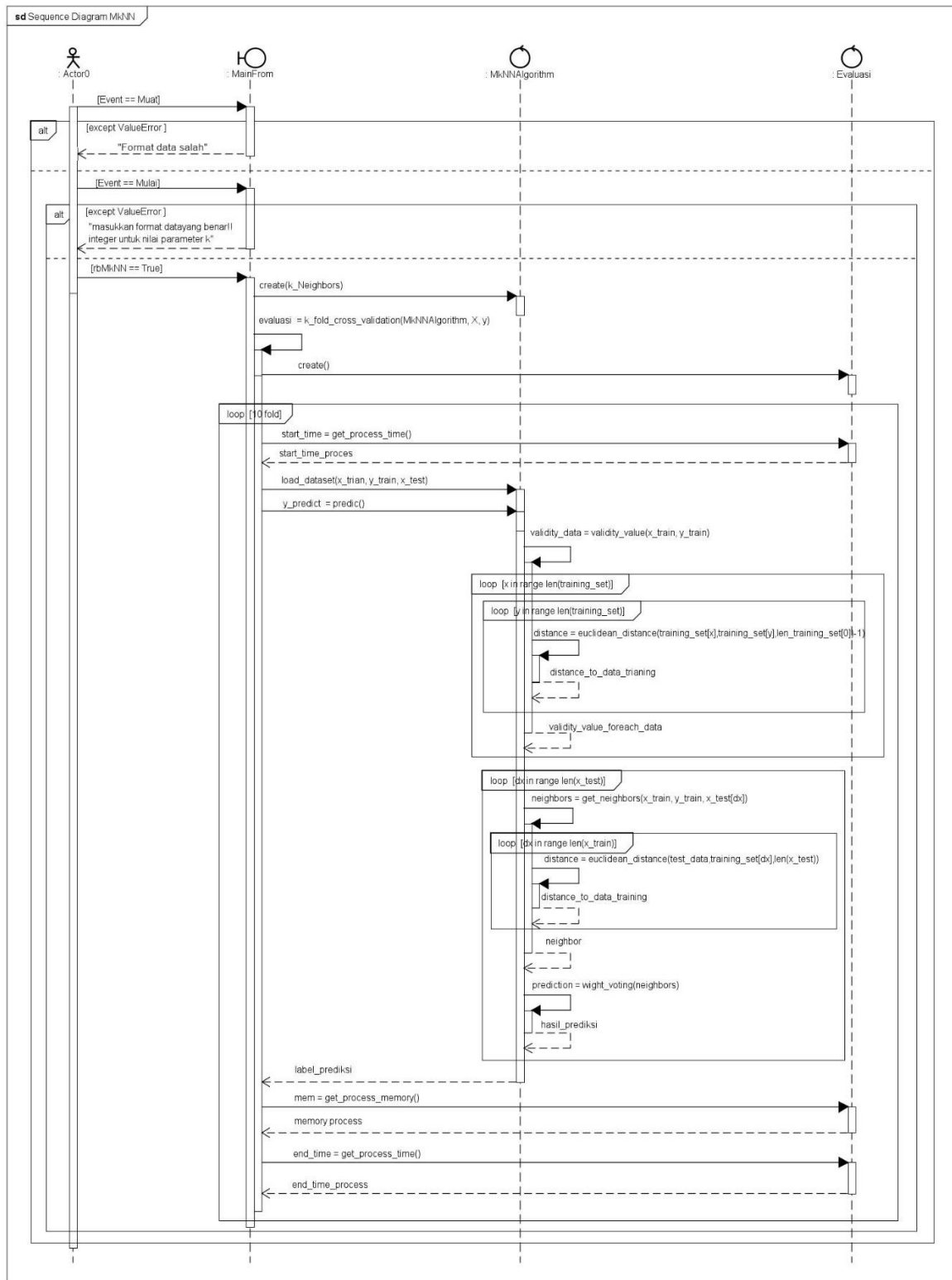
Berdasarkan hasil analisa dari fase *incaption* yang menghasilkan proses bisnis berupa use cas diagram, maka pada pemodelan bisnis untuk fase *elaboration* yang membahas perancangan perangkat lunak akan digambarkan dalam diagram sequence.

a. Diagram *sequence*

Diagram *sequence* adalah diagram yang menggambarkan rangkaian interaksi antara objek atau kelas dalam sebuah sistem. Pada penelitian ini terdapat 2 buah *sequence diagram* yaitu, diagram *sequence* pada use case klasifikasi data menggunakan algoritma *k-Nearest Neighbor* dapat dilihat pada gambar IV-2. dan diagram *sequence* pada use case klasifikasi data menggunakan algoritma *Modified k-Nearest Neighbor* dapat dilihat pada gambar IV-3.



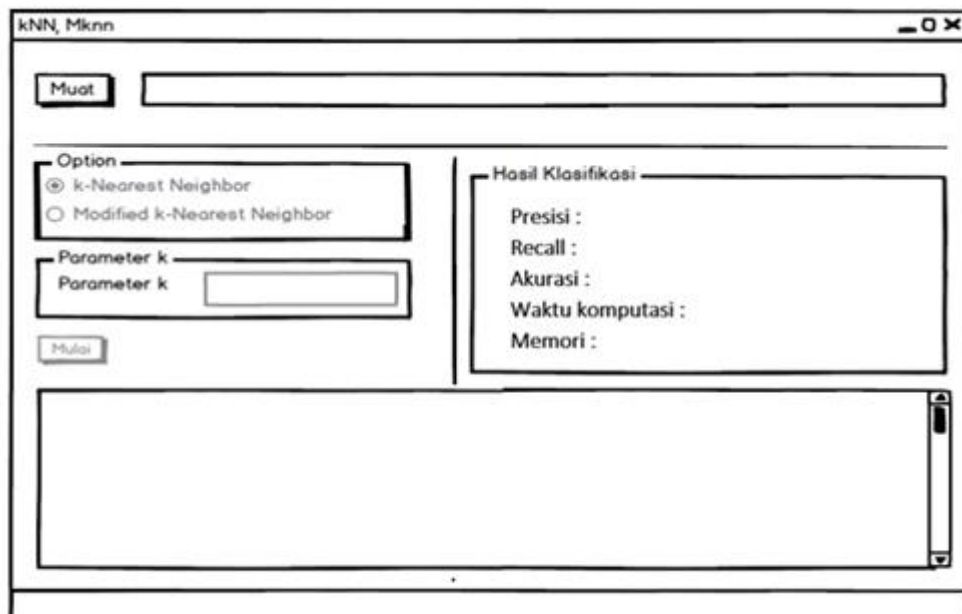
Gambar IV-2. Diagram *Sequence* Klasifikasi data menggunakan algoritma *k-Nearest Neighbor* (kNN)



Gambar IV-3. Diagram *Sequence* Klasifikasi data menggunakan algoritma *Modified k-Nearest Neighbor (MkNN)*.

4.2.2.2 Kebutuhan sistem

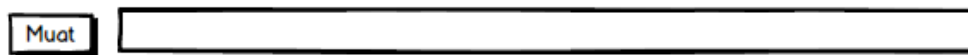
Berdasarkan pemodelan bisnis sebelumnya telah digambarkan menggunakan diagram sequence rangkaian interaksi antara objek atau kelas pada sebuah sistem atau perangkat lunak. Sehingga pada subbab ini membahas tahapan selanjutnya yaitu perancangan antar muka dari perangkat lunak yang akan dikembangkan. Untuk rancangan antar muka secara keseluruhan dapat dilihat pada gambar IV-4. Untuk memahami masing-masing fitur pada antar muka dapat dilihat pada gambar IV-5 sampai dengan gambar IV-11. Pada gambar IV-5 terdapat fitur muat data, gambar IV-6 terdapat fitur pilih algoritma, gambar IV-7 terdapat fitur input parameter k , gambar IV-8 terdapat fitur *button* mulai, gambar IV-9 fitur kinerja algoritma dan gambar IV-10 terdapat fitur menampilkan hasil pada setiap k -fold data.



Gambar IV-4. Rancangan antar muka perangkat lunak

a. Fitur Muat data

Proses perangkat lunak akan dimulai dengan menginputkan data terlebih dahulu, untuk menginputkan data ke dalam sistem pengguna harus menginputkan data pada fitur muat data, ketikkan *button* muat di klik oleh pengguna maka sistem akan menampilkan dialog pencarian berkas untuk memilih berkas yang akan diproses ke dalam sistem. Rancangan muat data dapat dilihat pada gambar IV-5.



Gambar IV-5 Rancangan muat data

b. Fitur pilih algoritma

Setelah pengguna memuat data maka selanjutnya sistem akan memproses dan pengguna dapat memilih algoritma yang akan digunakan untuk dilakukan proses klasifikasi, terdapat 2 pilihan pada sistem dengan menggunakan *radio button* yaitu algoritma *k- Nearest Neighbor* dan *Modified k- Nearest Neighbor*, dengan rules jika pengguna memilih algoritma *k-Nearest Neighbor Neighbor* maka algoritma *k- Nearest Neighbor* saja yang akan dilakukan perhitungan, berlaku sebaliknya. Rancangan pilih algoritma dapat dilihat pada gambar IV-6.



Gambar IV-6. Rancangan pilih algoritma

c. Fitur Input parameter k

Setelah pengguna memilih algoritma yang akan digunakan untuk mengklasifikasikan data masukan, maka selanjutnya pengguna harus melakukan penginputan terhadap nilai parameter k, penginputan terhadap parameter k sangat penting untuk mendukung jalannya sistem dari algoritma klasifikasi yang di pilih pengguna. Rancangan input nilai parameter k dapat dilihat pada table IV-7.

The image shows a user interface element for entering a parameter value. It consists of a rectangular box with a double border. On the left side of the box, the text "Parameter k" is written twice, one above the other. On the right side of the box, there is a smaller, empty rectangular input field.

Gambar IV-7. Rancangan Input nilai parameter k

d. Fitur *button* mulai

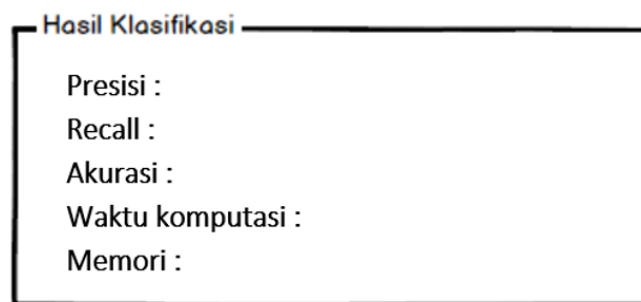
Setelah semua proses atau tahapan pada sistem di jalankan oleh pengguna, selanjutnya pengguna akan mengklik *button* mulai, dengan mengklik *button* mulai maka sistem akan menjalankan proses pengklasifikasian terhadap data masukan dengan menggunakan algoritma yang telah dipilih pada radio *button*. Rancangan *button* mulai dapat dilihat pada gambar IV-8.



Gambar IV-8. Rancangan *button* 'mulai'

e. Fitur menampilkan kinerja algoritma

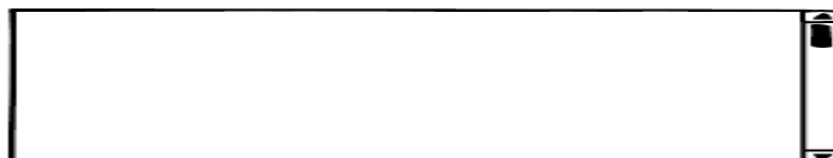
Selanjutnya setelah pengguna mengklik tombol mulai sistem akan melakukan proses pengklasifikasian dan menghasilkan output berupa nilai *precision*, *recall*, *accuracy*, waktu komputasi dan memori hasil dari 10 *fold cross validation*. Rancangan fitur menampilkan kinerja algoritma dapat dilihat pada table IV-9.



Gambar IV-9. Rancangan Menampilkan kinerja algoritma

f. Fitur menampilkan hasil setiap k-fold data

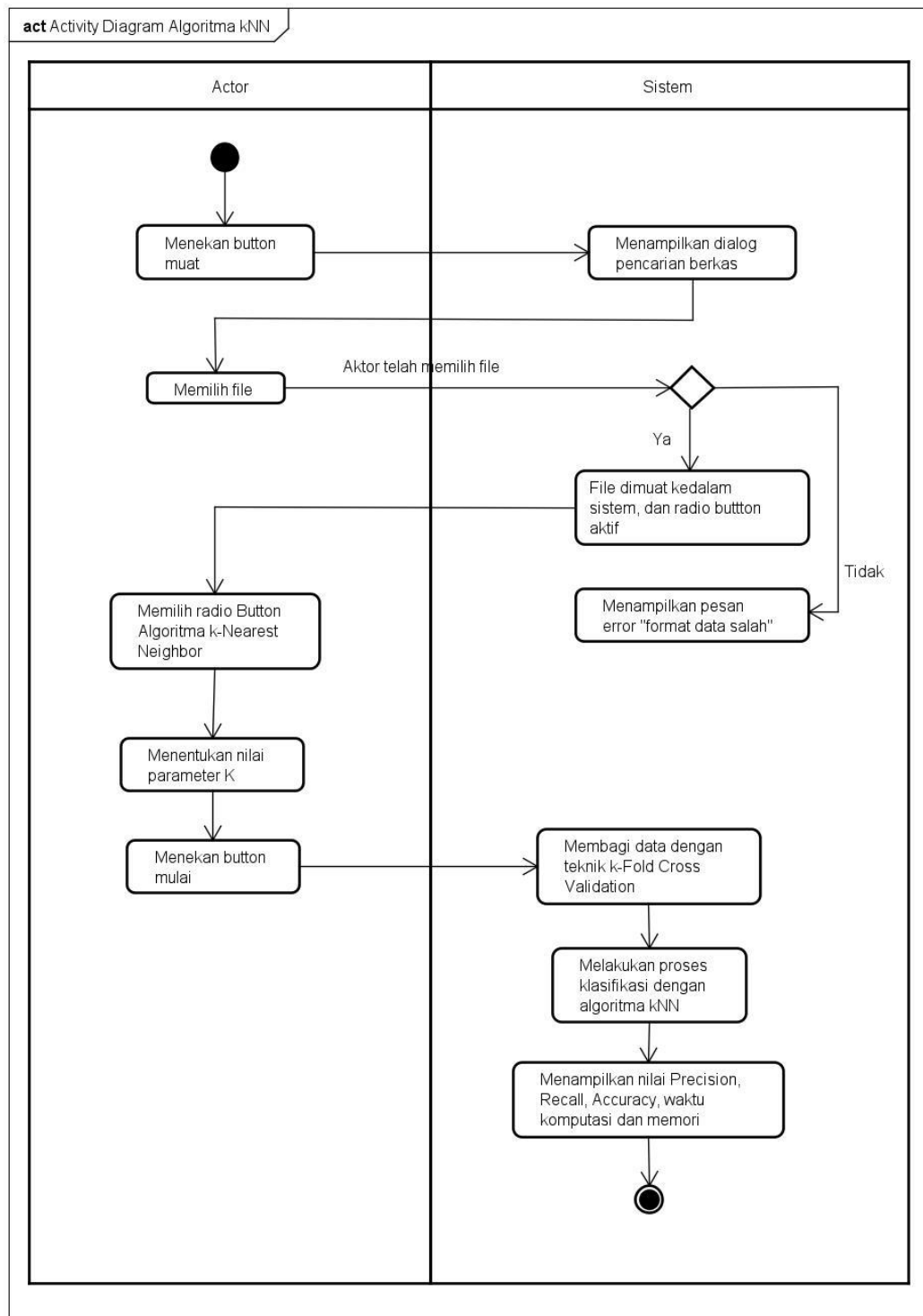
Setelah sistem melakukan klasifikasi terhadap data masukan, selain menampilkan hasil kinerja terbaik dari 10-*fold cross validation*. Sistem juga menampilkan hasil kinerja algoritma berupa nilai *precision*, *recall* dan *accuracy* waktu dan memori dari setiap pembagian data dalam *k-fold cross validation*. Rancangan menampilkan hasil setiap *k-fold* data dapat dilihat pada gambar IV-10.



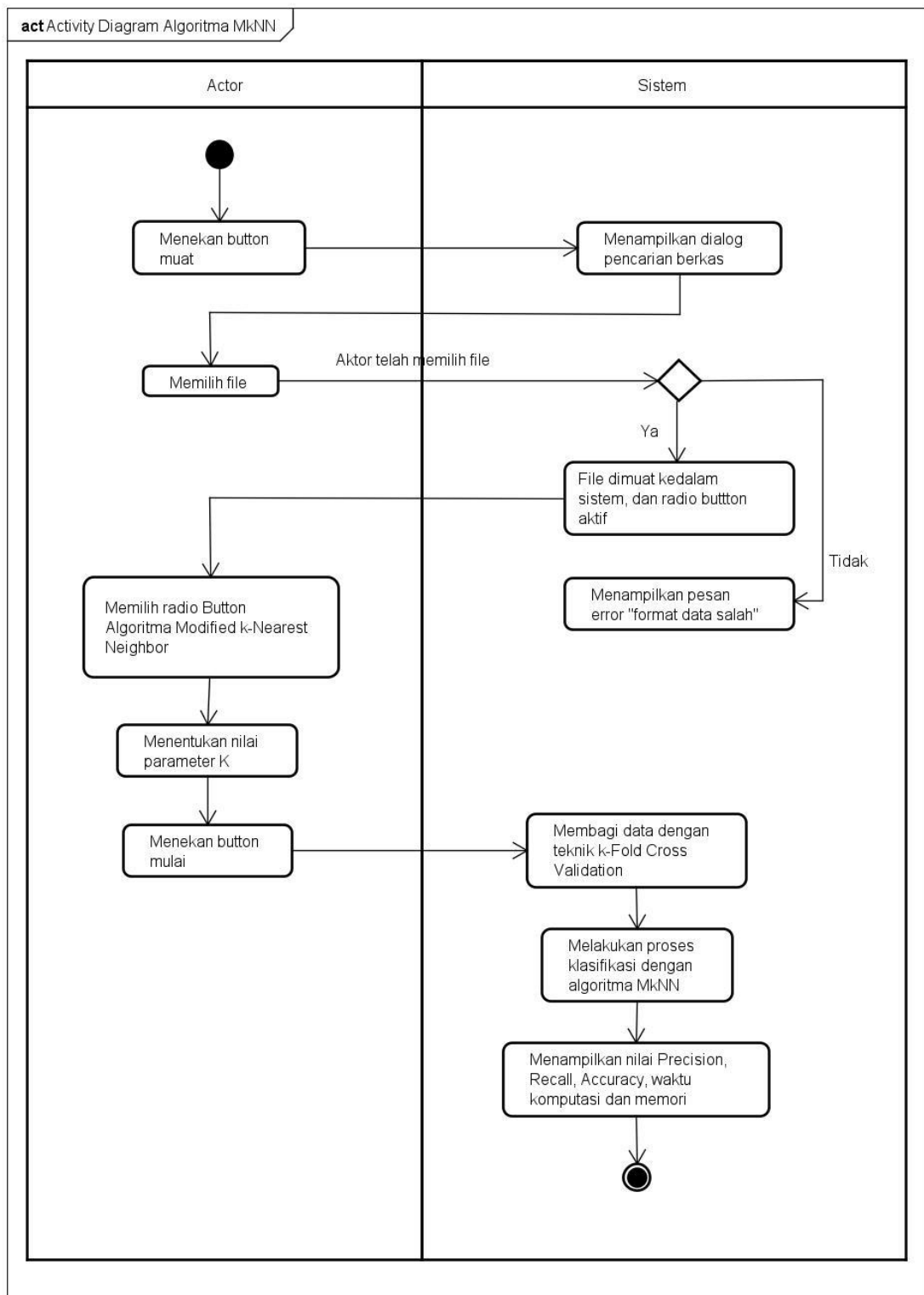
Gambar IV-10. Rancangan menampilkan hasil setiap k-fold data.

4.2.2.3 Analisis dan desain

Tahapan analisis dan desain pada fase elaboration akan menjelaskan lebih rinci alur proses dalam sistem yang di bangun menggunakan diagram aktivitas. Diagram aktivitas adalah diagram yang menggambarkan urutan aktivitas pada setiap use case rancangan sebuah perangkat lunak. Pada perancangan perangkat lunak terdapat 3 buah diagram aktivitas yaitu; Diagram aktivitas pada *use case* klasifikasi data menggunakan algoritma *k-Nearest Neighbor* dapat dilihat pada gambar IV-11, dan diagram aktivitas pada *use case* klasifikasi data menggunakan algoritma *Modifeid k- Nearest Neighbor* dapat dilihat pada gambar IV-12.



Gambar IV-11. Diagram aktivitas klasifikasi data menggunakan algoritma *k-Nearest Neighbor*



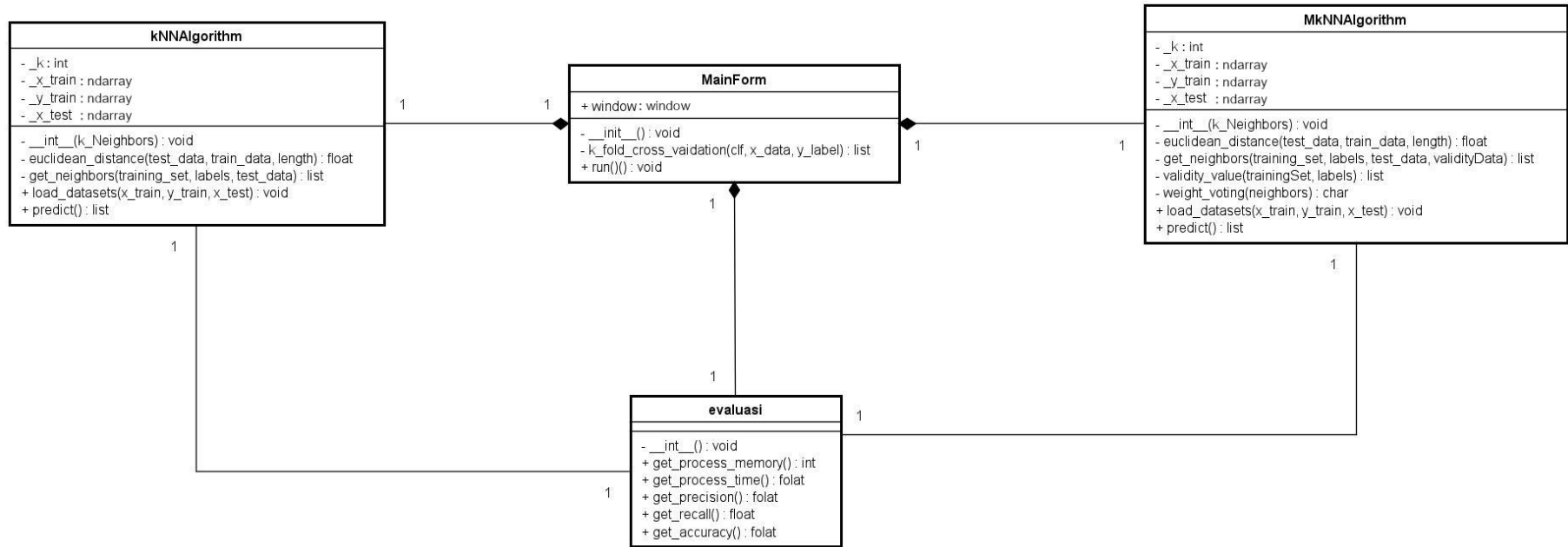
Gambar IV-12. Diagram aktivitas klasifikasi data menggunakan algoritma *Modified k-Nearest Neighbor*.

4.2.3 Fase *Contruction*

Fase *contruction* pada *Rational Unified Process* (RUP) merupakan fase yang berfokus pada pengembangan perangkat lunak berdasarkan rancangan yang telah ditentukan pada fase-fase sebelumnya, hasil dari fase *contruction* yaitu menghasilkan sebuah perangkat lunak yang telah dikembangkan, sehingga perangkat lunak tersebut akan digunakan sebagai alat penelitian yang akan digunakan oleh peneliti.

4.2.3.1 Pemodelan bisnis

Pada tahapan pemodelan bisnis untuk fase *contruction* akan menjelaskan tentang kelas-kelas yang terdapat pada sistem dan hubungan antar kelas yang akan digambarkan dalam diagram kelas dimodelkan dengan UML. Diagram kelas adalah diagram yang menggambarkan hubungan antara kelas pada suatu sistem. Diagram kelas pada perangkat lunak yang dibangun untuk pengklasifikasian data menggunakan algoritma *k-Nearest Nighbor* dan *Modified k-Nearest Nighbor* terdapat 4 buah kelas yang terdiri dari 1 kelas *boundary*, dan 3 kelas *countrroller* yang secara rinci dapat dilihat pada gambar IV-13.



Gambar IV-13 Diagram kelas

4.2.3.2 Kebutuhan sistem

Kebutuhan sistem akan mejekaskan kelas-kelas yang telah dirancang pada diagram kelas yang dapat dilihat pada gambar IV-13. Kemudian kelas-kelas tersebut akan diimplementasikan dalam Bahasa pemrograman yang digunakan yaitu bahas pemrograman python. Kelas-kelas yang akan dimplementasikan tersebut dapat dilihat pada table IV-8.

Table IV-8. Implementasi kelas-kelas.

No	Kelas	Nama File	Keterangan
1	MainForm	main.py	Kelas boundary yang membuat akses pengguna dengan sistem
2	Evaluasi	evaluasi.py	Kelas controller yang digunakan untuk mengatasi proses hasil klasifikasi algoritma.
3	kNNAlgorithm	classification.py	Kelas controller yang mengatasi proses klasifikasi data masukan menggunakan algoritma k-Nearest Neighbor
4	MkNNAlgorithm	Classification.py	Kelas countroller yang menangani proses klasifikasi data masukan menggunakan algoritma Modified k-Nearest Neighbor

4.2.4 Fase Transition

Fase transition pada Rational Unified Process (RUP) merupakan tahapan atau fase akhir dari pengembangan sebuah perangkat lunak, pada fase ini akan dilakukan pengujian terhadap perangkat lunak yang telah dikembangkan pada fase sebelumnya.

4.2.4.1 Pemodelan bisnis

Perangkat lunak yang telah dikembangkan akan dilakukan pengujian pada fase transition, sebelum dilakukan pengujian akan dibuat rencana pengujian terhadap perangkat lunak menggunakan pengujian black box dan white box berdasarkan use-case yang telah dibuat pada fase sebelumnya, yang berguna untuk memastikan setiap use-case dapat terpenuhi pada perancangan perangkat lunak.

4.2.4.2 Kebutuhan sistem

Pada subbab ini akan dilakukan pengujian terhadap perangkat lunak, oleh karena itu peneliti menggunakan perangkat keras dan perangkat lunak yang sama dengan fase construction, perangkat keras tersebut berupa :

1. Laptop merek Lenovo
2. Processor Intel® Core i5
3. RAM 4GB
4. Hardisk

Sedangkan untuk perangkat lunak yang digunakan pada pengujian berupa :

1. Sistem operasi windows 10
2. JetBrains Pycharm Python IDE.

4.2.4.3 Pengujian

Pada subbab ini akan dilakukan pengujian terhadap perangkat lunak, sebelum melakukan pengujian akan dibuat rancangan pengujian yang selanjutnya digunakan untuk diimplementasikan pada pengujian perangkat lunak.

1. Rencana pengujian

Rencana pengujian perangkat lunak akan dijelaskan dalam bentuk table-table dibawah ini, dimana pengujian dilakukan berdasarkan use-case yang telah dirancang sebelumnya.

a. Rencana pengujian use case klasifikasi dataset *website phishing* menggunakan algoritma *k-Nearest Neighbor* (kNN)

Rencana pengujian dari use-case klasifikasi dataset *website phishing* menggunakan algoritma *k- Nearest neighbor* dapat dilihat pada table IV-9

Table IV-9. Rencana pengujian use case klasifikasi dataset *website phishing* menggunakan *algoritma k- Nearest Neighbor*

No	ID	Pengujian	Metode pengujian	Tingkat pengujian
1	US-01-001	Mengklik <i>button</i> muat dan memilih data masukan dengan format data yang sesuai	<i>Black Box</i>	Pengujian Unit
2	US-01-002	Mengklik <i>button</i> muat dan memilih data masukan dengan format data yang tidak sesuai	<i>Black Box</i>	Pengujian Unit
3	US-01-003	Menginput nilai parameter <i>k</i> pada kolom parameter <i>k</i>	<i>Black Box</i>	Pengujian Unit

4	US-01-004	Mengklik <i>button</i> “Mulai”	<i>Black Box</i>	Pengujian Unit
5	US-01-005	Melakukan proses pengklasifikasian menggunakan algoritma <i>k-Nearest Neighbor</i> .	<i>White Box</i>	Pengujian Unit

b. Rencana pengujian use case klasifikasi dataset website phishing menggunakan algoritma Modified k-Nearest Neighbor

Rencana pengujian use case klasifikasi dataset *website phishing* menggunakan algoritma MkNN dapat dilihat pada table IV-10.

Table IV-10. Rencana pengujian use case klasifikasi dataset *website phishing* menggunakan algoritma *Modified k-Nearest Neighbor*

No	ID	Pengujian	Metode pengujian	Tingkat Pengujian
1	US-02-001	Mengklik <i>button</i> muat dan memilih data masukan dengan format data yang sesuai	<i>Black Box</i>	Pengujian Unit
2	US-02-002	Mengklik <i>button</i> muat dan memilih data masukan dengan format data yang tidak sesuai	<i>Black Box</i>	Pengujian Unit
3	US-02-003	Memilih <i>radio button</i> “ <i>Modified k-Nearest Neighbor</i> ”	Black Box	Pengujian Unit
4	US-02-004	Memasukkan nilai parameter <i>k</i> pada kolom parameter <i>k</i>	Black Box	Pengujian Unit
5	US-02-005	Mengklik <i>button</i> “mulai”	Black Box	Pengujian Unit
6	US-02-006	Melakukan proses pengklasifikasian menggunakan algoritma <i>Modified k-Nearest Neighbor</i> .	White Box	Pengujian Unit

2. Implementasi Pengujian

Pada implementasi akan dilakukan pengujian perangkat lunak berdasarkan rancangan pengujian yang telah di buat sebelumnya.

a. Pengujian use case klasifikasi *website phishing* menggunakan algoritma *k-Nearest Neighbor*

Pengujian use case klasifikasi *website phishing* menggunakan algoritma *k-Nearest Neighbor* dapat dilihat pada table IV-11.

Table IV-11. Pengujian use case klasifikasi *website phishing* menggunakan algoritma *k-Nearest Neighbor*.

ID	Pengujian	Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Hasil yang didapatkan	Kesimpulan pengujian
US-01-001	Mengklik button muat dan memilih data masukan dengan menggunakan format data yang sesuai	Mengklik button “mulai”	Peneliti menginputkan dataset yang formatnya sesuai	Dataset yang siap diproses untuk tahap selanjutnya	Sistem melakukan proses algoritma	Terpenuhi
US-01-002	Mengklik button muat dan memilih data masukan dengan format data yang tidak sesuai	Mengklik button “mulai”	Peneliti menginputkan dataset yang formatnya tidak sesuai.	Sistem menampilkan pesan “format data salah!”	Sistem menampilkan pesan “format data salah”	Terpenuhi

US-01-003	Memilih <i>radio button</i> “ <i>k-Nearest Neighbor</i> ”	Memilih <i>radio button</i> “ <i>k-Nearest Neighbor</i> ”	Peneliti mengklik <i>radio button</i> untuk mengaktifkannya	<i>Button</i> dapat di klik dan kolom input parameter <i>k</i> dapat digunakan.	Tombol dapat di klik dan kolom input parameter <i>k</i> dapat digunakan	Terpenuhi
US-01-004	Memasukkan nilai parameter <i>k</i> pada kolom parameter <i>k</i>	Menginputkan nilai pada kolom parameter <i>k</i>	Peneliti menginputkan nilai parameter <i>k</i> bertipe integer	Nilai dapat diambil dan ditampilkan pada interface	Nilai dapat diambil dan ditampilkn pada interface	Terprnuhi
US-01-005	Mengklik <i>button</i> “mulai”	Mengklik <i>button</i> “mulai”	Peneliti mengklik <i>butoon</i> untuk mengaktifkannya	<i>Button</i> dapat di klik dan mengaktifkan perhitungan algoritma yang dipilih.	<i>Button</i> dapat di klik dan mengaktifkan perhitungan algoritma yang dipilih.	Terpenuhi
US-01-006	Melakukan proses pengklasifikasian menggunakan algoritma “ <i>k-Nearest Neighbor</i> ”.	Mengklik <i>button</i> “mulai”	Peneliti menginputkan data masukan berupa dataset yang berekstensi .csv	Hasil proses klasifikasi berupa nilai <i>precision</i> , <i>recall</i> , <i>accuracy</i> , waktu komputasi dan memori.	Hasil pengklasifikasian.	Terpenuhi

b. Pengujian use case klasifikasi website phishing dengan menggunakan algoritma Modified k-Nearest Neighbor

Pengujian use case klasifikasi website phishing menggunakan algoritma Modified k-Nearest Neighbor dapat dilihat pada table IV-12.

Table IV-12. Pengujian use case klasifikasi website phishing menggunakan algoritma Modified k-Nearest Neighbor.

ID	Pengujian	Prosedur pengujian	Masukkan	Keluaran yang diharapkan	Hasil yang didapatkan	Kesimpulan pengujian
US-02-001	Mengklik button muat dan memilih data masukan dengan menggunakan format data yang sesuai	Mengklik button “mulai”	Peneliti menginputkan dataset yang formatnya sesuai	Dataset yang siap diproses untuk tahap selanjutnya	Sistem melakukan proses algoritma	Terpenuhi
US-02-002	Mengklik button muat dan memilih data masukan dengan format data yang tidak sesuai	Mengklik button “mulai”	Peneliti menginputkan dataset yang formatnya tidak sesuai.	Sistem menampilkan pesan “format data salah!”	Sistem menampilkan pesan “format data salah”	Terpenuhi
US-02-003	Memilih <i>radio button</i> “ <i>Modified k-Nearest Neighbor</i> ”	Memilih <i>radio button</i> “ <i>Modified</i> ”	Peneliti mengklik <i>radio button</i>	<i>Button</i> dapat di klik dan kolom input parameter k dapat digunakan.	Tombol dapat di klik dan kolom input	Terpenuhi

		<i>k-Nearest Neighbor</i> ”	untuk mengaktifkannya		parameter k dapat digunakan	
US-02-004	Memasukkan nilai parameter <i>k</i> pada kolom parameter <i>k</i>	Menginputkan nilai pada kolom parameter <i>k</i>	Peneliti meninputkan nilai parameter <i>k</i> bertipe integer	Nilai dapat diambil dan ditampilkan pada interface	Nilai dapat diambil dan ditampilkan pada interface	Terpenuhi
US-02-005	Mengklik <i>button</i> “mulai”	Mengklik <i>button</i> “mulai”	Peneliti mengklik <i>button</i> untuk mengaktifkannya.	Button dapat di klik dan mengaktifkan perhitungan algoritma yang dipilih.	Button dapat di klik dan mengaktifkan perhitungan algoritma yang dipilih.	Terpenuhi
US-02-006	Melakukan proses pengklasifikasian menggunakan algoritma “ <i>Modified k-Nearest Neighbor</i> ”.	Mengklik <i>button</i> “mulai”	Peneliti menginputkan data masukan berupa dataset yang berekstensi .csv	Hasil proses klasifikasi berupa nilai <i>precision</i> , <i>recall</i> , <i>accuracy</i> , waktu komputasi dan memori.	Hasil pengklasifikasian.	Terpenuhi

4.3 Kesimpulan

Pengembangan perangkat lunak pada penelitian ini menggunakan metode RUP untuk mengembangkan perangkat lunak perbandingan kinerja algoritma *k-Nearest Neighbor* dan Modified *k-Nearest Neighbor* pada klasifikasi *website phishing*. Metode RUP memiliki 4 fase dalam pengembangan perangkat lunak yaitu fase *inception*, *elaboration*, *construction*, dan *transition*. Pada fase *inception* dilakukan identifikasi kebutuhan perangkat lunak yang digambarkan dalam use case diagram, pada fase *elaboration* merancang interface dan menggambarkan alur proses sistem menggunakan diagram aktivitas dan diagram *sequence*, pada fase *construction* mengimplementasikan rancangan dari perangkat lunak kedalam bahasa pemrograman berdasarkan diagram class yang telah dibuat, dan fase terakhir atau fase *transition* melakukan pengujian terhadap perangkat lunak yang telah dikembangkan untuk memastikan perangkat lunak telah memenuhi kebutuhan yang sebelumnya telah ditentukan, agar perangkat lunak yang dirancang dapat digunakan sebagai alat penelitian pada penelitian yang akan dilakukan