

BAB III

METODOLOGI PENELITIAN

3.1 Pendahuluan

Pada bab sebelumnya telah dijelaskan landasan teori yang mendasari dilakukannya penelitian ini. Selanjutnya akan dibahas unit penelitian yang dilakukan, tahapan penelitian yang akan diimplementasikan, metodologi penelitian yang digunakan serta manajemen proyek pada penelitian. Tahapan penelitian dijadikan sebagai acuan pada setiap fase pengembangan dan memberikan sebuah solusi untuk rumusan masalah dan mencapai tujuan penelitian.

3.2 Unit Penelitian

Unit penelitian yang digunakan pada penelitian ini adalah situs *UCI Machine Learning* yang merupakan situs penyedia *dataset website phishing*.

3.3 Data

3.3.1 Jenis Data dan Sumber Data

Pada penelitian ini menggunakan data sekunder sebagai obyek penelitian yang bersumber dari *UCI Machine Learning* dengan 30 atribut dan 2456 *records* data. Atribut dataset *website phishing* tersebut akan dijelaskan pada table III-1 dibawah ini.

Table III-1. Deskripsi Atribut Dataset Website *Phishing*

Label	Deskripsi
Having IP Address	Menggunakan IP Adress sebagai domain pada URL yang bernilai; {-1 jika tidak menggunakan dan 1 jika menggunakan}.
URL Length	Memiliki panjang URL yang bernilai; {-1 jika panjang URL kurang dari 54 karakter, 0 jika jumlah panjang URL 54 hingga 75 karater, dan 1 jika panjang URL lebih dari 75 karakter}.
Shortening Service	Menggunakan layanan penyingkat URL bernilai ; {-1 jika tidak menggunakan dan 1 jika menggunakan}.
Having “@” Symbol	Menggunakan symbol “@” pada URL yang bernilai; {-1 jika tidak menggunakan dan 1 jika menggunakan}.
Redirecting using “//”	Menggunakan symbol “//” pada URL yang bernilai; {-1 jika tidak menggunakan dan 1 jika menggunakan}.
Adding Prefix or Suffix	Menggunakan symbol “-“ pada domain dalam URL yang bernilai; {-1 jika tidak menggunakan dan 1 jika menggunakan}.
Sub Domain dan Multi Sub Domain	Menggunakan subdomain yang bernilai: {-1 jika tidak menggunakan subdomain, 0 jika menggunakan 1 sub domain dan 1 jika menggunakan lebih dari 1 sub domain}.
SSL Final State	Memiliki sertifikat SSL terpercaya seperti “GeoTrust, GoDaddy, Network Solution, Thawte, Comdo, Doster dan VeriSign” yang bernilai; {-1 jika mempunyai sertifikat terpercaya, 0 jika

	mempunyai sertifikat yang belum dipercaya dan 1 tidak mempunyai sertifikat}.
Domain Register Length	Berlakunya domain berdasarkan periode waktu yang berniali; {-1 jika lebih dari 1 tahun dan 1 jika kurang dari 1 tahun}.
Favicon	Memiliki favicon dari link eksternal yang bernilai; {-1 jika tidak memiliki dan 1 jika memiliki}.
Port	Menggunakan port seperti 21, 22, 23, 80, 445 dan lainnya yang bernilai; {-1 jika tidak menggunakan dan 1 jika menggunakan}
HTTPS Token	Menggunakan HTTPS ke dalam bagian domain pada URL yang bernilai; {-1 jika tidak menggunakan dan 1 jika menggunakan}.
Request URL	Permintaan URL eksternal dari keseluruhan secara persentasi yang bernilai; {-1 jika kurang dari 22%, 0 jika antara 22% hingga 61% dan 1 jika lebih dari 61% }
URL Of Anchor	Penggunaan tag <a> yang mengarah selain ke domain yang sama dari keseluruhan secara persentase yang bernilai: {-1 jika kurang dari 31%, 0 jika antara 31% hingga 67%, dan 1 jika lebih dari 67% }.
Link in <meta>, <script> dan <link> Tags	Menggunakan tag <link>, <meta> dan <script> yang mnegarah selain ke domain yang sama dari keseluruhan secara persentase yang bernialai; {-1 jika kurang dari 17%, 0 jika antar 17% hingga 81% dan 1 jika lebih dari 81% }.
Service From Handler (SFH)	Domanin pemrosesan Server From Handler yang bernilai; {-1 jika domain yang sama, 0 jika domain yang berbeda dan 1 jika kosong}.

Submitting Information to Email	Menggunakan fungsi “mail() atau mail to” dalam PHP untuk mengirim informasi user yang bernilai; {-1 jika tidak menggunakan dan 1 jika menggunakan}.
Abnormal URL	Website memiliki kecocokan dengan catatan yang ditunjukkan pada WHOIS database yang bernilai; {-1 jika cocok dan 1 jika tidak cocok}
Website Forwarding atau Redirect	Pengalihan yang dilakukan website yang bernilai; {-1 jika kurang dari 2 kali, 0 jika antar 2 hingga 4 kali dan 1 jika lebih dari 4 kali}
Status Bar Customization, “onMouseOver” event.	Menggunakan event onMouseOver dan terjadi perubahan pada status bar yang berniali; {-1 jika tidak menggunakan dan 1 jika menggunakan}.
Disabling Right Click	Menggunakan Right Click pada website yang bernilai; {-1 jika diaktifkan dan 1 jika diaktifkan}.
Using Pop-up Window	Menggunakan Pop-up Window dalam menginput data yang bernilai; {-1 jika tidak menggunakan dan 1 jika menggunakan}.
Iframe Redirection	Menggunakan fungsi Iframe yang berniali; {-1 jika tidak menggunakan dan 1 jika menggunakan}.
Age Of Domain	Domain memiliki umur berdasarkan periode waktu yang bernilai; {-1 jika lebih dari atau sama dengan 6 bulan dan 1 kurang dari 6 bulan}.
DNS Record	Memiliki catatan DNS pada domain yang berniali; {-1 jika memiliki dan 1 jika tidak memiliki}.
Website Traffic	Rank dari traffic website dalam basis data alexa yang bernilai;{-1 jika diatas 100.000, 0 jika dibawah 100.000 dan 1 tidak terdaftar }

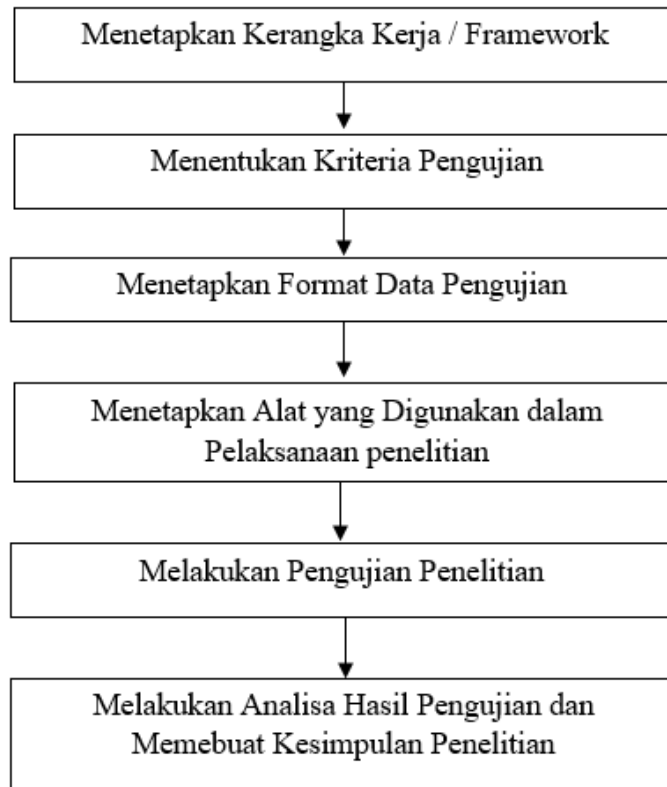
Page Rank	Nilai dari Page Rank website yang bernilai; {-1 lebih dari atau sama dengan 0.2 dan 1 kurang dari 0.2}.
Google Indeks	Website terdapat pada Google Indeks yang bernilai; {-1 jika terdapat dan 1 jika tidak terdapat}
Number fo Link Pointing to Page	Jumlah link eksternal yang menunjuk ke website yang berniali; {-1 jika lebih dari 2 link, 0 jika antara 1 atau 2 link dan 1 jika tidak ada link}.
Statistical Reports	Host berasal dari Top Phishing IPs atau Top Phishing Domains yang dibuat oleh bebrerapa pihak seperti StopBadware dan PhishTanak yang bernilai; {-1 jika tidak dan 1 jika iya}
Result	Hasil identifikasi website yang berniali; {1 jika website tidak berniali phishing dan 1 jika website bernilai phishing}.

3.3.2 Metode Pengumpulan Data

Data diunduh secara manual berekstensi *.arff* pada situs *UCI Machine Learning*, kemudian data tersebut akan dikonversi kedalam file berekstensi *.csv* dan disimpan dalam satu folder.

3.4 Tahapan Penelitian

Untuk membandingkan kinerja dari algoritma *k- Nearest Neighbor* dan *Modified k- Nearest Neighbor*, penelitian akan melakukan tahapan-tahapan yang akan dijelaskan pada sub bab 3.4.1 sampai dengan 3.4.6. Diagram tahapan penelitian dapat dilihat pada Gambar III-1.



Gambar III-1. Diagram Tahapan Penelitian

3.4.1 Menentukan Kerangka Kerja/*Framework*

Penelitian ini memiliki kerangka kerja/ *framework* yang akan dibangun untuk mengetahui kinerja dari algoritma dalam mengklasifikasikan dataset website *phishing* menggunakan algoritma *k- Nearest Neighbor* dan *Modified k- Nearest Neighbor*.. Adapun kerangka kerjanya yaitu :

- 1. Perhitungan nilai k terbaik pada algoritma *k- Nearest Neighbor* dan *Modified k- Nearest Neighbor***

Untuk mengetahui k optimum/terbaik pada algoritma *k-Nearest Neighbor* dan *Modified k-Nearest Neighbor* maka akan dilakukan pengujian menggunakan

terhadap algoritma tersebut dengan menggunakan nilai parameter k yang akan digunakan sebagai ruang solusi yaitu $k=3$, $k=5$, $k=7$ dan $k=9$. Kemudian dihitung nilai jaraknya menggunakan *euclidian distance*. Untuk mendapatkan jarak terdekat maka data yang telah dihitung nilai jaraknya akan diurutkan mulai dari jarak yang terdekat. Kemudian akan diambil setiap tetangga sebanyak nilai k dan dihitung mayoritas kelas yang muncul untuk digunakan sebagai hasil klasifikasi. Setiap nilai k akan dihitung hasil kinerjanya.

2. Perancangan klasifikasi *k-Nearest Neighbor*

Proses klasifikasi dataset website phishing menggunakan algoritma *k-Nearest Neighbor* melalui beberapa tahap yaitu, terlebih dahulu menentukan nilai parameter k , selanjutnya menghitung jarak menggunakan *euclidian distance*, lalu mengelompokkan data berdasarkan hasil perhitungan jarak *euclidian*. dan berdasarkan nilai tetangga terdekat, serta memilih nilai yang sering muncul dari tetangga terdekat sebagai prediksi data selanjutnya.

3. Perancangan klasifikasi *Modified k-Nearest Neighbor*

Klasifikasi website *phishing* menggunakan metode *Modified k-Nearest Neighbor* melalui beberapa tahap yaitu, terlebih dahulu menentukan nilai parameter k , selanjutnya menghitung nilai validitas, setelah itu menghitung *euclidian distance*, mengelompokkan data berdasarkan hasil perhitungan jarak *euclidian* dilanjutkan dengan proses perhitungan *weighted voting* dengan mengalikan nilai validitas dari setiap data latih dengan *wight* berdasarkan pada nilai

jarak *euclidian*. selanjutnya akan dipilih nilai yang sering muncul dari tetangga terdekat sebagai prediksi data selanjutnya.

4. Evaluasi menggunakan *k-fold cross cross validation*

Setelah didapatkan model algoritma *k- Nearest Neighbor* dan algoritma *Modified k-Nearest Neighbor* kemudian akan dievaluasi menggunakan *k-fold cross validation* dimana data akan partisi ke-*k* sebagai data testing dan menggunakan sisa partisi lainnya sebagai data training. Pada penelitian ini akan digunakan *10- fold cross validation* untuk mengevaluasi model pada kedua lagoritma tersebut, sehingga data akan dibagi menjadi 10 fold atau 10 subset data yang memeiliki ukuran yang sama, dimana data tersebut dengan cross validation dari 10 foold tersebut 9 foldnya akan digunakan sebagai data latih dan 1 fold digunakan sebagai data uji dan dilaukan secara bergantian. Sehingga tahap selanjutnya yaitu menghitung rata-rata akurasi dari *k*-kali validasi yang digunakan sebagai validasi final.

5. Perancangan Analisis Kinerja Algoritma

Untuk mengetahui kinerja dari algoritma *k- Nearest Neighbor* dan *Modified k- Nearest Neighbor* untuk klasifikasi website *phishing* perlu dilakukan pengujian dari kinerja kedua algoritma tersebut. Pengujian tersebut bertujuan untuk mengetahui kinerja dari kedua algoritma tersebut. Tahapan pengujian dilakukan setelah kedua algoritma tersebut melalui proses di masing-masing yang akan dievaluasi menggunakan *10- fold cross validation* setelah didapat validasi final

akan di analisis kinerja dari masing-masing algoritma dengan menggunakan *Confusion Matriks* untuk menghitung nilai *precision*, *recall*, dan *accuracy*, serta waktu komputasi, besarnya memori yang dibutuhkan algoritma. Setelah didapatkan hasil evaluasi dari kedua algoritma tersebut akan diketahui kinerja algoritma mana yang lebih baik diantara keduanya.

3.4.2 Menetapkan Kriteria Pengujian

Secara rinci penjelasan mengenai tahapan ini akan dijelaskan pada bab IV. Pengujian yang dilakukan terhadap kinerja algoritma *k-Nearest Neighbor* dan *Modified k-Nearest Neighbor* dengan nilai *k* yang digunakan sebagai ruang solusi 3, 5, 7 dan 9. Selanjutnya model dari masing-masing algoritma akan dia evaluasi dengan *10- fold cross validation* dan selanjutnya dilakukan pengujian menggunakan *confusion matrix* untuk mengetahui nilai *precision*, *recall*, *accuracy*, serta akan diketahui waktu komputasi dan besarnya memori yang dibutuhkan algoritma *k-Nearest Neighbor* dan *Modified k-Nearest Neighbor* pada klasifikasi data menggunakan dataset *website phishing*. berdasarkan penjelasan sebelumnya pengujian dilakukan dengan bertujuan untuk mengetahui kinerja algoritma mana yang terbaik dari kedua algoritma tersebut.

3.4.3 Menetapkan Format Data Pengujian

Hasil evaluasi kinerja algoritma *k- Nearest Neighbor* (kNN) dan *Modified k-Nearest Neighbor* (MkNN) dengan *Confusion Matriks* diperoleh nilai *precision*, *recall*, dan *accuracy* sehingga dapat diketahui nilai *k* mana yang paling optimal

untuk menghasilkan tingkat akurasi tertinggi, akan digambarkan dalam tabel III-2 untuk *k- Nearest Neighbor* dan dan tabel III-3 untuk *Modified k-Nearest Neighbor*.

Tabel III-2. Rancangan Tabel Untuk Pengujian *Confusion Matriks k-Nearest Neighbor*.

<i>k-Nearest Neighbor</i>				
<i>k</i>	<i>k-fold</i>	<i>Precision</i>	<i>Recall</i>	<i>Accuracy</i>
3	1			
	...			
	10			
5	1			
	...			
	10			
7	1			
	...			
	10			
9	1			
	...			
	10			

Tabel III-3. Rancangan Table Untuk Pengujian *Confusion Matric Modified k-Nearest Neighbor*.

<i>Modified k-Nearest Neighbor</i>				
<i>k</i>	<i>k-fold</i>	<i>Precision</i>	<i>Recall</i>	<i>Accuracy</i>
3	1			
	...			
	10			
5	1			
	...			
	10			
7	1			
	...			
	10			
9	1			
	...			
	10			

Hasil evaluasi dari waktu komputasi dan besarnya memori yang dibutuhkan algoritma, akan digambarkan dalam tabel III-4 untuk algoritma kNN dan tabel III-5 untuk algoritma MkNN.

Tabel III-4. Rancangan Table Untuk Pengujian Waktu Komputasi Dan Memori Algoritma K-Nearest Neighbor

<i>k-Nearest Neighbor</i>			
<i>k</i>	<i>k-fold</i>	Waktu (detik)	Memory (byte)
3	1		
	...		
	10		

5	1		
	...		
	10		
7	1		
	...		
	10		
9	1		
	...		
	10		

Tabel III-5. Rancangan Table Untuk Pengujian Waktu Komputasi Dan Memori

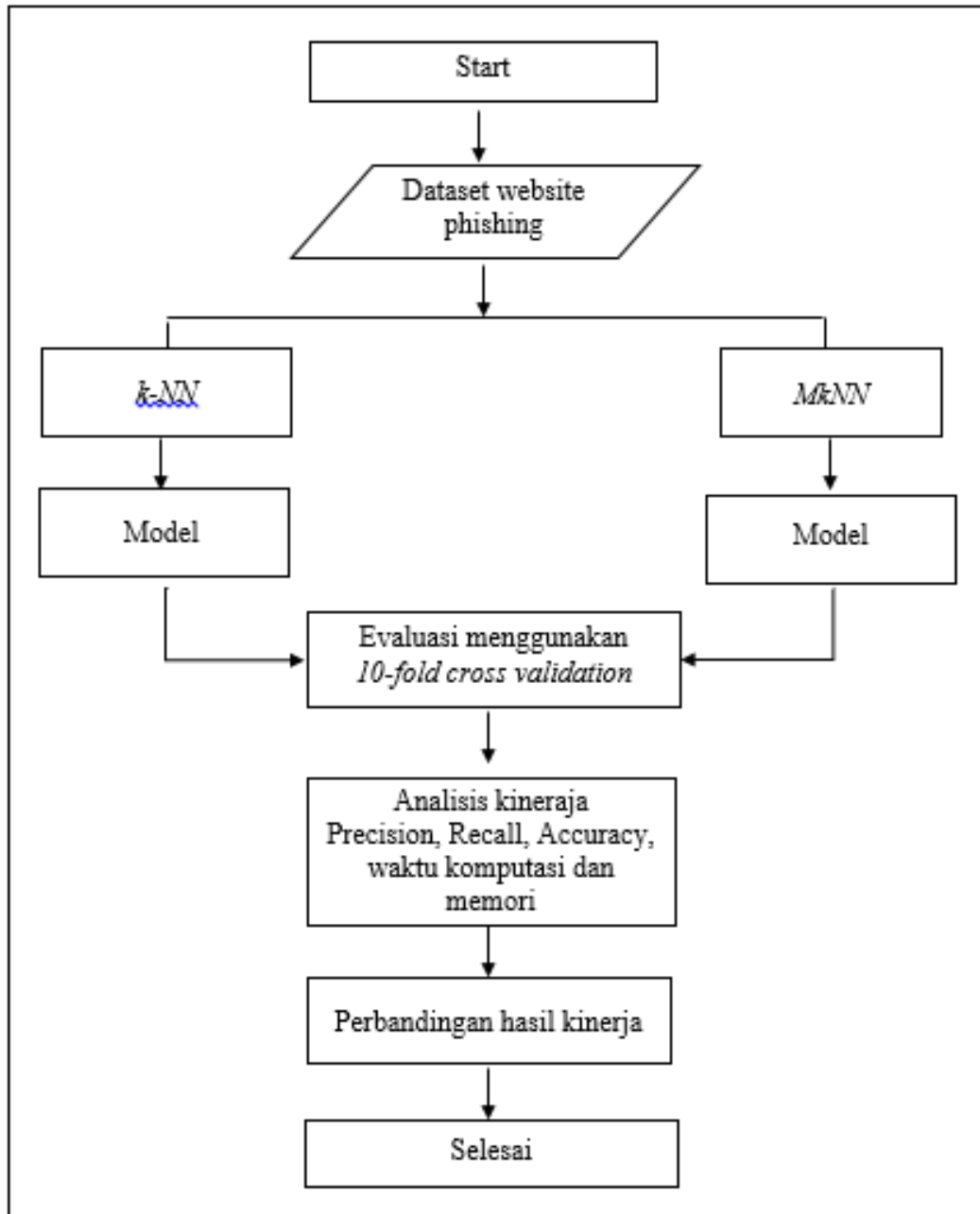
Algoritma Modified K-Nearest Neighbor

<i>Modified k-Nearest Neighbor</i>			
<i>k</i>	<i>k-fold</i>	Waktu (detik)	Memory (byte)
3	1		
	...		
	10		
5	1		
	...		
	10		
7	1		
	...		
	10		
9	1		
	...		
	10		

3.4.4 Menentukan Alat yang Digunakan dalam Pelaksanaan Penelitian

Untuk melaksanakan penelitian ini yang bertujuan perbandingan kinerja dari algoritma *k-Nearest Neighbor* dan *Modified k-Nearest Neighbor* penulis akan membutuhkan alat penelitian berupa perangkat lunak yang dapat digunakan untuk mengevaluasi kinerja algoritma *k-Nearest Neighbor* dan *Modified k-Nearest Neighbor*. Oleh karena itu, penulis akan mengembangkan perangkat lunak yang dapat digunakan untuk melakukan pengklasifikasian data menggunakan dataset *website phishing* yang akan diklasifikasikan menggunakan algoritma *k-Nearest Neighbor* dan *Modified k-Nearest Neighbor*, yang bertujuan untuk mengetahui kinerja dari algoritma berupa nilai *precision*, *recall*, *accuracy*, waktu komputasi dan besarnya memori yang dibutuhkan, serta untuk membuktikan pernyataan yang mengatakan bahwa algoritma *Modified k-Nearest Neighbor* lebih baik dari pada algoritma *k-Nearest Neighbor*.

3.4.5 Melakukan Pengujian Penelitian



Gambar III-2. Tahapn Pengujian Penelitian

3.4.6 Melakukan Analisa Hasil Pengujian dan Membuat Kesimpulan

Analisa dilakuka untuk mengetahui kinerja dari algoritma *k- Nearest Neighbor* dan *Modified k- Nearest Neighbor* dengan menggunakan *confusion matrix* yang diperoleh dari algoritma k- Nearest Neighor dan Modified k- Nearest Neighbor, hasil pengujian algoritma berupa nilai *precision*, *recall*, *accuracy*, waktu komputasi dan besarnya memori yang digunakan setiap algoritma. Setelah hasil tersebut diperoleh selanjutnya data akan ditampilkan dan dilakukan perbandingan antara masing-masing algoritma dengan nilai *k* yang ada, hasil analisis pengujian algoritma akan digambarkan dalam table III-6. Setelah proses pengujian dilakukan maka tahapan sealjutnya penulis akan membuat kesimpulan dari hasil pengujian yang dilakukan dan akan dijelaskan pada bab V.

Tabel III-6. Tabel Rancangan Hasil Pengujian Kinerja Algoritma *k-Nearest Neighbor* dan *Modified k-Nearest Neighbor*

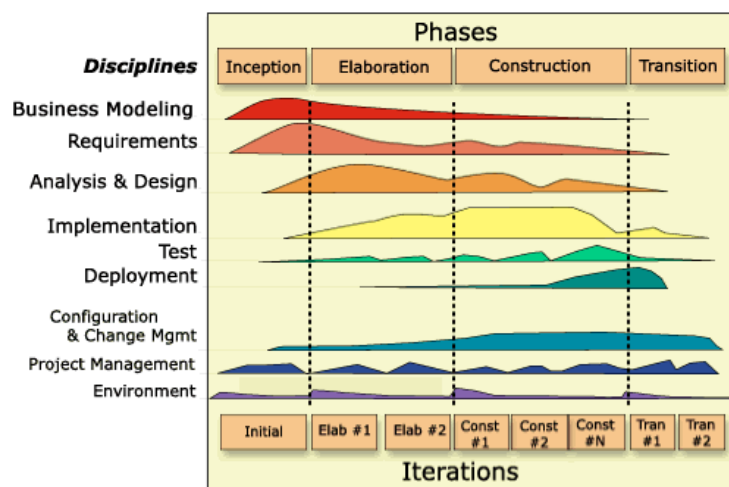
<i>k</i>	<i>k-Nearest Neighbor</i>					<i>Modified k k-Nearest Neighbor</i>				
	<i>Precision</i>	<i>Recall</i>	<i>Accuracy</i>	<i>Waktu</i>	<i>Memori</i>	<i>Precision</i>	<i>Recall</i>	<i>Accuracy</i>	<i>Waktu</i>	<i>Memori</i>
3										
5										
7										
9										

3.5 Metodologi Pengembangan Perangkat Lunak

Metode pengembangan perangkat lunak yang akan digunakan pada penelitian ini yaitu pengembangan perangkat lunak yang berorientasi objek menggunakan metode *Rational Unified Process* (RUP).

3.5.1 Konsep Rational Unified Process (RUP)

Rational Unified Process (RUP) adalah kerangka kerja proses rekayasa perangkat lunak. Secara umum, langkah-langkah yang akan dilakukan pada pengembangan perangkat lunak RUP terdapat 4 fase yaitu fase *inception*, *elaboration*, *construction*, dan *transition*. Tujuannya adalah untuk memastikan produksi perangkat lunak berkualitas tinggi yang memenuhi kebutuhan end-users, dalam jadwal dan anggaran yang dapat diprediksi. RUP adalah salah satu metode pengembangan perangkat lunak yang menggunakan konsep berorientasi objek, dengan aktifitas yang berfokus pada pengembangan model dengan menggunakan Unified Model Language (UML).



Gambar III-3. Model Rational Unified Process

Dari gambar diatas dapat dilihat bahwa Model RUP memiliki dua dimensi tahapan yaitu :

- Dimensi *horizontal*, dimana dimensi tersebut mewakili aspek dinamis dari pengembangan perangkat lunak. Aspek dinamis terdapat tahapan-tahapan pengembangan perangkat lunak yang terdiri dari beberapa fase. Pada dimensi ini terdiri atas beberapa tahapan/fase yaitu *inception*, *elaboration*, *construction*, dan *transition*.
- Dimensi *vertical*, dimana dimensi tersebut mewakili aspek statis dari pengembangan perangkat lunak, aspek tersebut berupa komponen proses, kegiatan alur kerja, artefak dan pekerjaan dalam proses pengembangan sebuah perangkat lunak. Pada dimensi ini terdiri dari business modeling, requirement, analysis and design, implementation, test, deployment, configuration and change manegement, project management dan enviroentment.

3.5.2 Fase –fase RUP

1. Inception

Inception merupakan fase pertama dari model RUP, pada fase ini akan memahami semua tentang ruang lingkup, tujuan proyek dan mendapatkan cukup informasi untuk mengkonfirmasi bahwa proyek harus dilanjutkan atau memungkinkan tidak perlu dilanjutkan. Fase *inception* meliputi; tahap *business modeling* dimana penulis akan menentukan kebutuhan pengguna dan fitur dari perangkat lunak, tahap *requirements* penulis akan

mengumpulkan data dengan mengunduh dataset *website phishing*, tahap *analysis* dan *design* penulis akan membuat *use case* diagram, pada tahap *implementation* penulis mendokumentasikan kebutuhan pengguna, dan pada tahap test penulis akan memastikan apakah kebutuhan pengguna dan fitur pada perangkat lunak sudah valid.

2. *Elaboration*

Elaboration merupakan fase kedua dari model RUP yang bertujuan untuk menentukan arsitektur dasar dari sebuah perangkat lunak. sehingga pada fase ini akan menganalisa masalah utama yang terjadi, menyusun pondasi arsitektur, dan membangun rencana proyek. Fase *elaboration* meliputi; tahap *business modeling* dimana penulis akan menentukan arsitektur perangkat lunak, desain basis data dan desain interface sesuai kebutuhan pengguna yang telah ditentukan sebelumnya, tahap *requirements* penulis akan mengkaji ulang kebutuhan pengguna dan melengkapi kebutuhan jika perlu ditambahkan, tahap *analysis* dan *design* penulis akan membuat diagram aktivitas dan diagram *sequence*, tahap *implementation* penulis mendokumentasikan arsitektur perangkat lunak, desain basis data, desain interface, diagram aktivitas dan diagram *use case* serta penulis memastikan semuanya sudah valid pada tahap test.

3. *Construction*

Construction merupakan fase ketiga dari model RUP yang bertujuan untuk mengimplementasikan hasil desain ke dalam bahasa pemrograman dan memastikan implementasi sesuai rancangan. Fase *construction* meliputi; tahap *business modeling* dimana penulis menentukan kelas-kelas yang dibutuhkan perangkat lunak, tahap *requirements* penulis akan menentukan Bahasa pemrograman yang akan digunakan serta mengidentifikasi kebutuhan lainnya yang dibutuhkan dalam proses pengembangan perangkat lunak seperti perangkat keras dan perangkat lunak/software yang akan penulis gunakan untuk mengembangkan perangkat lunak, tahap *analysis* dan *design* penulis akan membuat class diagram, tahap *implementation* penulis akan mengimplementasikan kelas-kelas yang telah ditentukan kedalam bahasa pemrograman yang penulis gunakan, dan pada tahap *test* penulis akan melakukan pengujian unit testing untuk memastikan kelas-kelas yang telah ditentukan tersebut telah terimplementasikan.

4. *Transition*

Transition merupakan fase keempat dan terakhir pada model RUP, fase ini akan memastikan pemodelkan perangkat lunak sesuai dengan rancangan sebelumnya. Fase *transition* meliputi; tahap *business modeling* dimana penulis akan membuat scenario pengujian dari perangkat lunak yang dikembangkan, tahap *requirements* penulis akan menentukan tools

pengujian yang diperlukan dimana tools tersebut merupakan perangkat keras yang sama pada fase *construction*, tahap *analisis* dan *desain* penulis akan membuat table scenario pengujian, tahap *implementation* penulis akan melakukan pengujian perangkat lunak berdasarkan scenario yang telah dibuat, dan pada tahap *test* penulis meninjau ulang tahapan penelitian.

3.6 Manajemen Proyek Penelitian

Manajemen proyek penelitian adalah perencanaan aktivitas-aktivitas yang dilakukan peneliti yang dimulai dari tahapan awal yaitu inialisasi masalah sampai dengan tahapan akhir yaitu membuat kesimpulan dari penelitian. Kegiatan yang akan berlangsung selama proses penelitian dilakukan, akan di gambarkan dalam sebuah *Work Breakdown Structure* (WBS) yang dapat dilihat pada table III-7.

Tabel III-7 Penjadwalan Penelitian dalam *Work Breakdown Structure* (WBS)

ID	Task Name	Duration	Start	Finish	Predcessors
	Perbandingan kinerja algoritma k-nearest neighbor dan modified k-nearest neighbor pada klasifikasi <i>website phishing</i>.	221 days	Mon 04/02/19	Fri 06/12/19	
	Menentukan ruang lingkup dan unit pada penelitian.	35 days	Mon 04/02/19	Mon 25/03/19	
T1	Menentukan masalah penelitian	10 days	Mon 04/02/19	Fri 15/02/19	
T2	Membuat latar belakang dan rumusan masalah	8 days	Mon 18/02/19	Wed 27/02/19	T1
T3	Menentukan tujuan dan manfaat penelitian	5 days	Thu 28/02/19	Wed 06/03/19	T2
T4	Menentukan batasan masalah	5 days	Thu 07/03/19	Wed 13/03/19	T3
T5	Menentukan unit penelitian	6 days	Thu 14/03/19	Thu 21/03/19	T4
M1	Tersedia dokumen hasil tahapan penelitian	1 day	Mon 25/03/19	Mon 25/03/19	T5
	Menentukan dasar teori yang berkaitan dengan penelitian	27 days	Tue 26/03/19	Wed 01/05/19	
T6	Mengumpulkan jurnal, pape, dan literature ilmiah yang berkaitan dengan penelitian	12 days	Tue 26/03/19	Wed 10/04/19	T1
T7	Mempelajari kerja algoritma k-nearest neighbor dan modified k-nearest neighbor	14 days	Thu 11/04/19	Tue 30/04/19	T6
M2	Tersedia dokumen hasil tahapan penelitian	1 day	Wed 01/05/19	Wed 01/05/19	T7
	Menentukan kriteria pengujian	21 days	Thu 02/05/19	Thu 30/05/19	
T8	Menentukan metode perhitungan jarak	13 days	Thu 02/05/19	Mon 20/05/19	T6

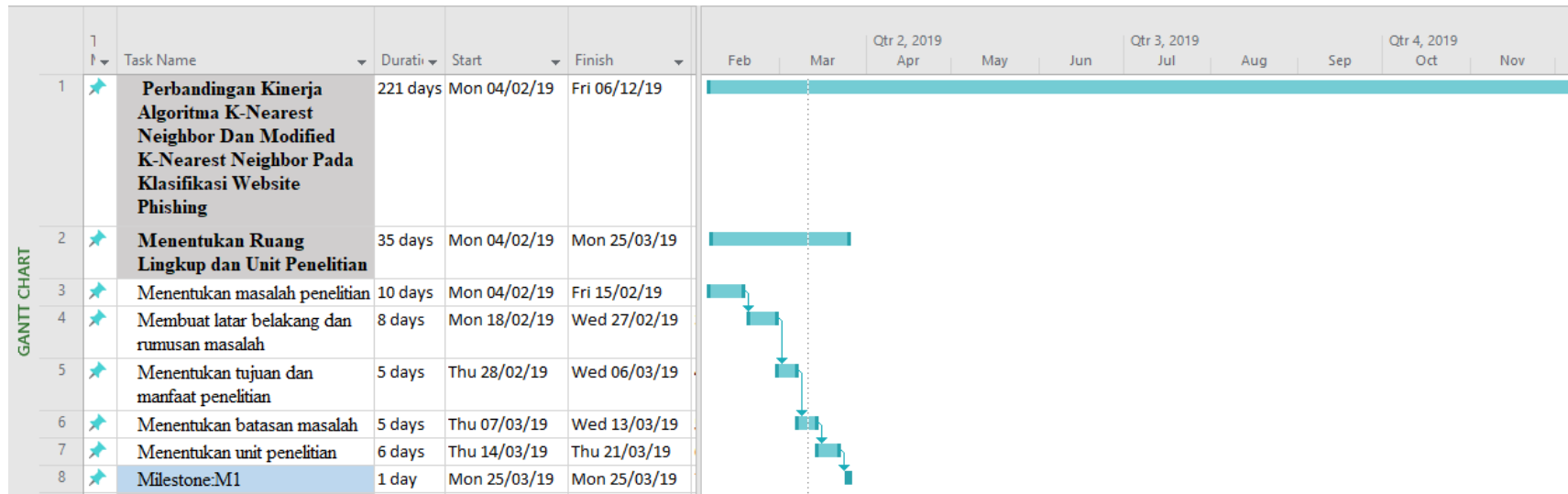
T9	Menentukan metode untuk mengevaluasi kinerja algoritma	7 days	Tue 21/05/19	Wed 29/05/19	T6
M3	Tersedia dokumen hasil tahapan penelitian	1 day	Thu 30/05/19	Thu 30/05/19	T7, T9
	Menentukan alat yang digunakan untuk pelaksanaan penelitian	119 days	Fri 31/05/19	Tue 12/11/19	
	<i>Inception</i>	13 days	Fri 31/05/19	Mon 17/06/19	
	<i>Busines modeling</i>	3 days	Fri 31/05/19	Mon 03/06/19	
T10	Menentukan user requirements dan fungsionalitas perangkat lunak	3 days	Fri 31/05/19	Mon 03/06/19	T1, T4, T5
	<i>Requirement</i>	4 days	Tue 04/06/19	Fri 07/06/19	
T11	Mengumpulkan dataset penelitian	4 days	Tue 04/06/19	Fri 07/06/19	T4,T5
	<i>Construction</i>	2 days	Mon 10/06/19	Tue 11/06/19	
T12	Membuat diagram use case	2 days	Mon 10/06/19	Tue 11/06/19	T10
	<i>Implementation</i>	2 days	Wed 12/06/19	Thu 13/06/19	
T13	Membuat dokumentasi	2 days	Wed 12/06/19	Thu 13/06/19	T10
	<i>Testing</i>	2 days	Fri 14/06/19	Mon 17/06/19	
T14	Memastikan user requirements dan fungsionalitas sudah valid	2 days	Fri 14/06/19	Mon 17/06/19	T13
	<i>Elaboration</i>	13 days	Tue 18/06/19	Thu 04/07/19	
	<i>Business Modelling</i>	3 days	Tue 18/06/19	Thu 20/06/19	

T15	Menentukan arsitektur perangkat lunak, desain basis data dan desain antarmuka	3 days	Tue 18/06/19	Thu 20/06/19	T14
	Requirement	3 days	Fri 21/06/19	Tue 25/06/19	
T16	Melengkapi user requirements yang telah didefinisikan pada fase inception	3 days	Fri 21/06/19	Tue 25/06/19	T10
	Analysis and design	3 days	Wed 26/06/19	Fri 28/06/19	
T17	Membuat diagram aktivitas dan diagram sequence	3 days	Wed 26/06/19	Fri 28/06/19	T12, T15
	Implementation	3 days	Mon 01/07/19	Wed 03/07/19	
T18	Membuat dokumentasi	3 days	Mon 01/07/19	Wed 03/07/19	T17
	Testing	1 day	Thu 04/07/19	Thu 04/07/19	
T19	Memastikan arsitektur perangkat lunak, desain basis data dan desain antarmuka sudah valid.	1 day	Thu 04/07/19	Thu 04/07/19	T18
	Construction	78 days	Fri 05/07/19	Tue 22/10/19	
	Business modelling	5 days	Fri 05/07/19	Thu 11/07/19	
T20	Menentukan kelas-kelas pada perangkat lunak	5 days	Fri 05/07/19	Thu 11/07/19	T19
	Requirements	5 days	Fri 12/07/19	Thu 18/07/19	
T21	Menentukan Bahasa pemrograman yang digunakan untuk mengembangkan perangkat lunak	3 days	Fri 12/07/19	Tue 16/07/19	T7, T8, T9
T22	Menentukan kebutuhan perangkat keras yang digunakan	2 days	Wed 17/07/19	Thu 18/07/19	T21
	Analisis and design	6 days	Fri 19/07/19	Fri 26/07/19	

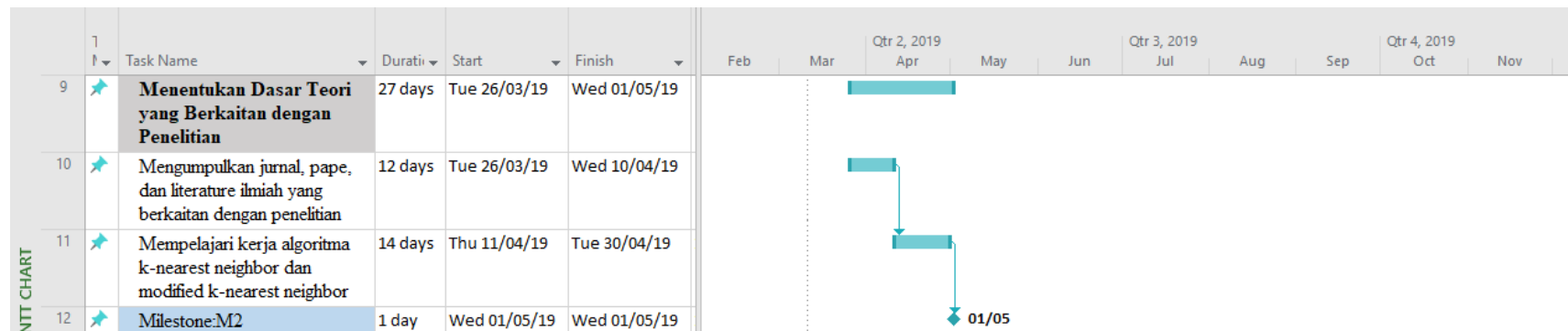
T23	Mambuat diagram kelas	6 days	Fri 19/07/19	Fri 26/07/19	T21, T22
	Implementation	32 days	Fri 26/07/19	Mon 09/09/19	
T24	Mengimplemantasikan kelas-kelas ke dalam kode perogram	32 days	Mon 29/07/19	Tue 10/09/19	T23
	Testing	30 days	Wed 11/09/19	Tue 22/10/19	
725	Melakukan unit testing	30 days	Wed 11/09/19	Tue 22/10/19	T23
	Transition	15 days	Wed 23/10/19	Tue 12/11/19	
	Business Modelling	2 days	Wed 23/10/19	Thu 24/10/19	
T26	Membuat rencana atau skenario pengujian	2 days	Wed 23/10/19	Thu 24/10/19	T25
	Requirements	2 days	Fri 25/10/19	Mon 28/10/19	
T27	Menentukan tools pengujian yang diperlukan dan sampel pertanyaan	2 days	Fri 25/10/19	Mon 28/10/19	T25
	Analysis and design	2 days	Tue 29/10/19	Wed 30/10/19	
T28	Membuat table skenario pengujian	2 days	Tue 29/10/19	Wed 30/10/19	T26, T27
	Implementation	4 days	Thu 31/10/19	Tue 05/11/19	
T29	Melakukan pengujian terhadap perangkat lunak berdasarkan skenario atau rencana pengujian	4 days	Thu 31/10/19	Tue 05/11/19	T26, T27
	Testing	4 days	Wed 06/11/19	Mon 11/11/19	
T30	Menguji skenario pengujian	4 days	Wed 06/11/19	Mon 11/11/19	T26, T27, T28
M4	Tersedia dokumen hasil tahapan penelitian	1 day	Tue 12/11/19	Tue 12/11/19	T30

	Melakukan pengujian penelitian	10 days	Wed 13/11/19	Tue 26/11/19	
T31	Menentukan rencana hasil penelitian	4 days	Wed 13/11/19	Mon 18/11/19	T3, T5
T32	Melakukan pengujian penelitian berdasarkan hasil pengujian perangkat lunak.	5 days	Tue 19/11/19	Mon 25/11/19	T29
M5	Tersedia dokumen hasil tahapan penelitian	1 day	Tue 26/11/19	Tue 26/11/19	T31, T32
	Melakukan analisa hasil pengujian dan membuat kesimpulan	9 days	Wed 27/11/19	Mon 09/12/19	
T33	Melakukan analisa terhadap hasil pengujian penelitian dengan membandingkan hasil kinerja algoritma	6 days	Wed 27/11/19	Wed 04/12/19	T32
T34	Membuat kesimpulan dan saran berdasarkan analisa terhadap hasil pengujian.	2 days	Thu 05/12/19	Fri 06/12/19	T33
M6	Tersedia dokumen hasil tahapan penelitian	1 day	Mon 09/12/19	Mon 09/12/19	T34

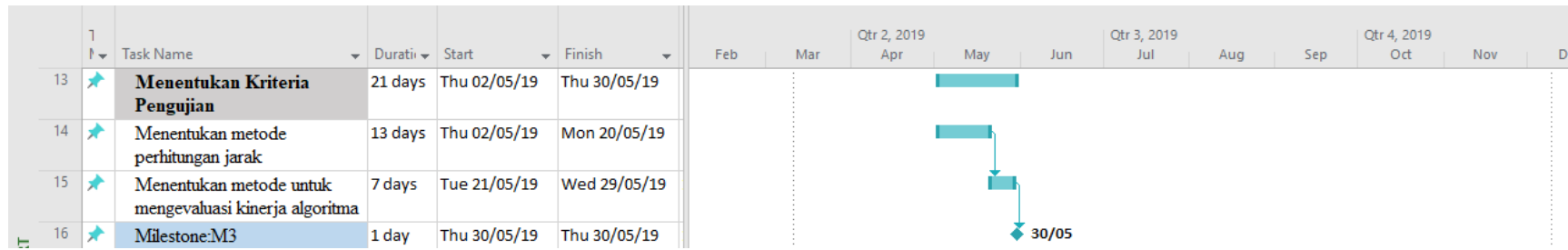
Selanjutnya penjadwalan penelitian yang dilakukan akan digambarkan dalam *gantt chart* menggunakan tool Microsoft project 2016, dimana *gantt chart* tersebut dapat dilihat pada Gambar III-4. *Gantt chart* pada tahap menentukan ruang lingkup penelitian, Gambar III-5. *Gantt chart* pada tahap menentukan dasar teori yang berkaitan dengan penelitian, Gambar III-6. *Gantt chart* pada tahap menentukan kriteria pengujian, Gambar III-7 sampai dengan Gambar III.10. *Gantt chart* pada tahap menentukan alat yang digunakan untuk pelaksanaan penelitian pada masing –masing fase *inception, elaboration, construction dan transition.*, Gambar III-11 *Gantt chart* pada tahap melakukan pengujian penelitian dan Gambar III-12 *Gantt chart* pada tahap analisis hasil pengujian dan membuat kesimpulan.



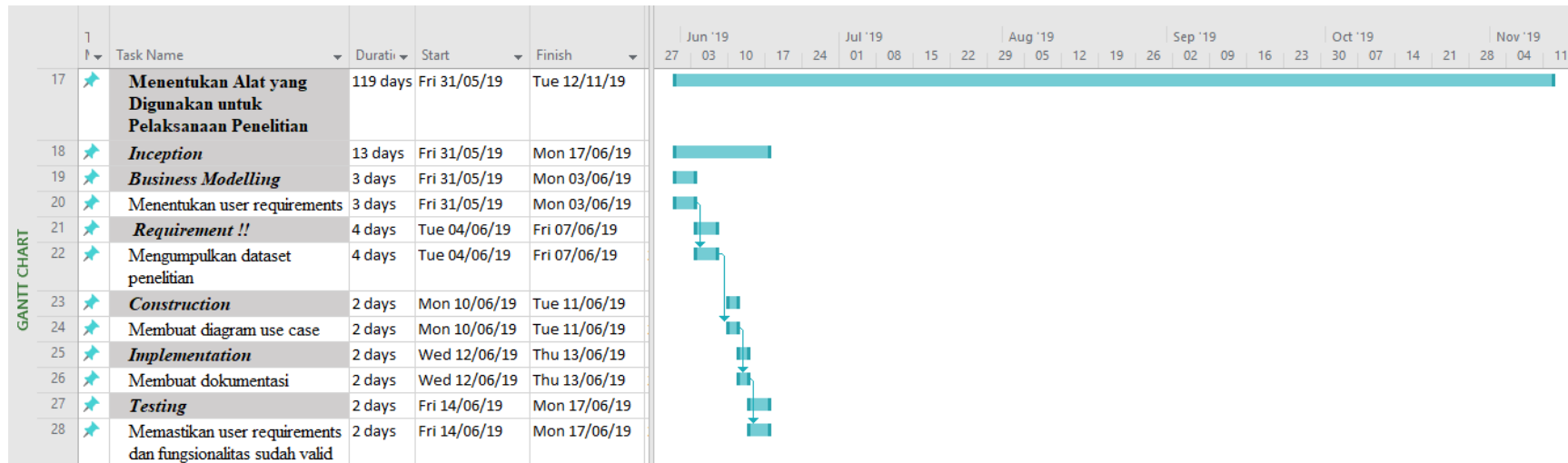
Gambar III-4. Gantt chart pada tahap menetapkan ruang lingkup serta unit penelitian



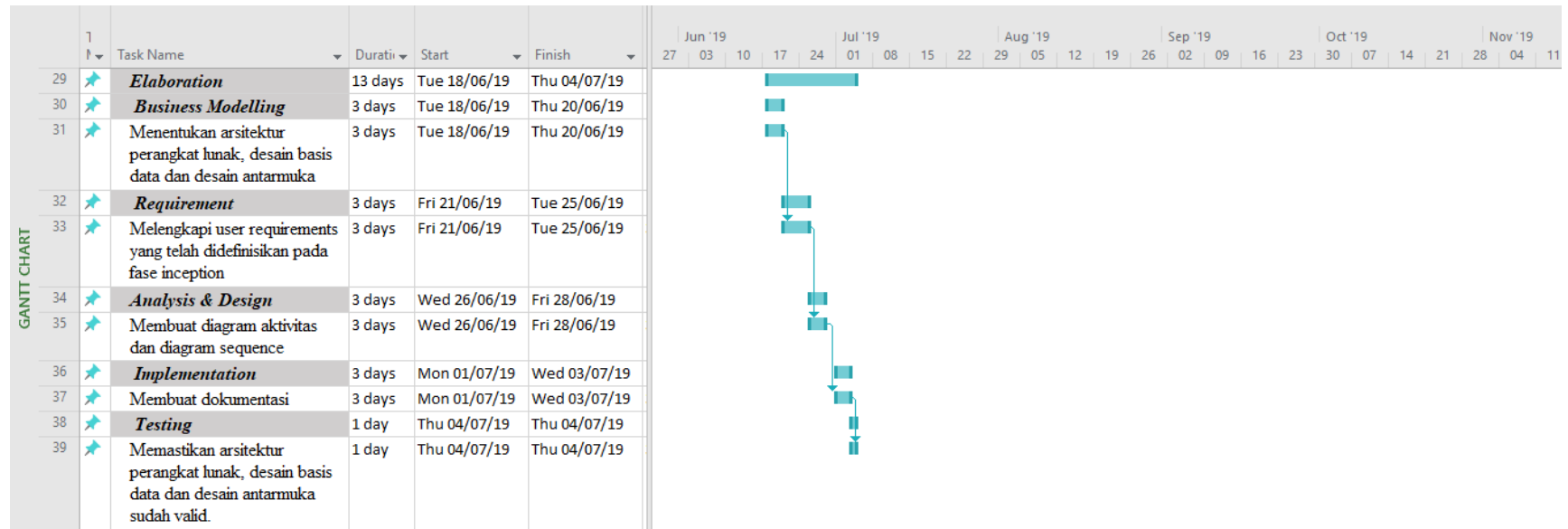
Gambar III-5. Gantt chart menentukan dasar teori yang berkaitan dengan penelitian



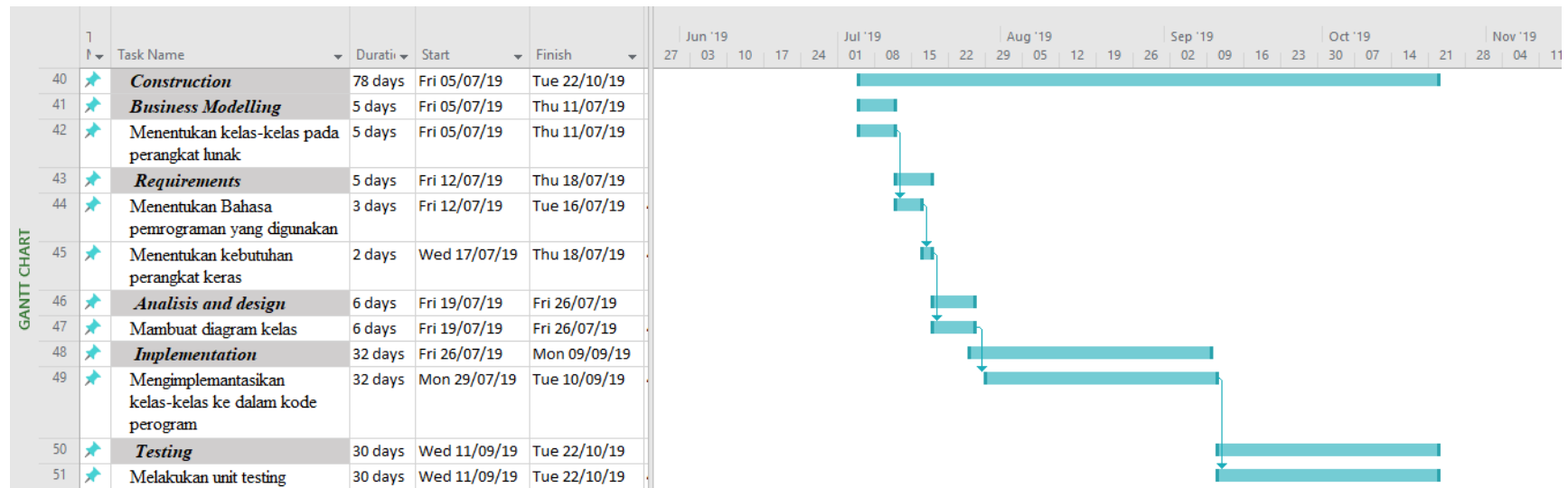
Gambar III-6. Gantt chart menentukan kriteria pengujian



Gambar III-7. Gantt chart menentukan alat yang digunakan pada penelitian untuk fase inception.



Gambar III-8. *Gantt chart* menentukan alat yang digunakan pada penelitian untuk fase elaboration



Gambar III-9. Gantt chart menentukan alat yang digunakan pada penelitian untuk fase construction

1.6 Kesimpulan

Berdasarkan uraian diatas telah dijelaskan tahapan dan langkah-langkah yang akan dilakukan pada penelitian ini dalam pengembangan perangkat lunak untuk melakukan klasifikasi data menggunakan algoritma k-Nearest Neighbor dan Modified k-Nearest Neighbor, serta telah digambarkan penjadwalan penelitian yang bertujuan agar sistem yang dirancang dapat diselesaikan berdasarkan waktu yang telah dijadwalkan. Selanjutnya aktivitas yang akan dilakukan yaitu mengembangkan perangkat lunak yang akan digunakan sebagai alat penelitian pada penelitian ini.