

**PENGEMBANGAN SISTEM INFORMASI *REVIEW SMARTPHONE*  
STUDI PADA TNT CELL BOJONEGORO**

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:  
Aula Septi Pertiwi  
NIM: 145150401111081



**PROGRAM STUDI SISTEM INFORMASI  
JURUSAN SISTEM INFORMASI  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2019**

# PENGESAHAN

PENGEMBANGAN SISTEM INFORMASI REVIEW SMARTPHONE  
STUDI PADA TNT CELL BOJONEGORO

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :  
Aulia Septi Pertiwi  
NIM: 145150401111081

Skripsi ini telah diuji dan dinyatakan lulus pada  
04 Januari 2019

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Fajar Pradana, S.ST, M.Eng  
NIP: 19871121 201504 1 004

Dosen Pembimbing II



Dr.Eng. Fitra A. Bachtiar, S.T, M.Eng  
NIK: 201201 840628 1 001

Mengetahui  
Ketua Jurusan Sistem Informasi



  
Dr.Eng. Herman Tolle, S.T, M.T  
NIP. 19740823 200012 1 001



## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 04 Januari 2019



Aulia Septi Pertiwi

NIM: 145150401111081

## PRAKATA

Segala Puji bagi Tuhan Yang Maha Esa, atas rahmat dan kasihnya penulis dapat menyelesaikan skripsi berjudul “Pengembangan Sistem Informasi *Review Smartphone* (Studi Pada : TNT CELL Bojonegoro)”

Penulis tidak akan dapat menyelesaikan penelitian ini tanpa ada dukungan dan bantuan dari orang-orang terdekat, oleh karena itu saya ingin menyampaikan terimakasih kepada :

1. Bapak Fajar Pradana, S.ST., M.Eng., selaku pembimbing akademik dan pembimbing satu yang telah meluangkan waktunya untuk memberikan bimbingan perihal akademik pada penulis selama berjalannya perkuliahan serta memberikan kesempatan kepada penulis untuk menjadi mahasiswa bimbingan skripsi beliau, sehingga penulis mampu menyelesaikan penelitian ini dan dapat menjalani perkuliahan sesuai dengan prosedur yang telah ditetapkan oleh akademik FILKOM dan dapat menyelesaikan penelitian ini.
2. Bapak Dr.Eng.Fitra Abdurrachman Bachtiar, S.T, M.Eng, selaku pembimbing dua yang telah memberikan kesempatan dan waktunya untuk membimbing penulis dalam menyelesaikan penelitian yang dilakukan, serta senantiasa dengan sabar dan teliti dalam memeriksa pengerjaan dokumen skripsi sampai terselesaikannya skripsi ini.
3. Ayah, Ibu serta seluruh keluarga saya yang selalu dengan sabar memberikan dukungan berupa doa, moril dan materil sehingga penulis mampu menyelesaikan penelitian ini.
4. Teman-teman *resident* Burning sun, Mu:in, Celebrate, Face, Octagon, Vera serta Chroma yang senantiasa menyemangati penulis agar penulis termotivasi untuk menyelesaikan penelitian ini.
5. Teman-teman Lembaga Badan Perwakilan Mahasiswa Sistem Informasi dan Eksekutif Mahasiswa Sistem Informasi yang telah menjadi sarana bagi penulis untuk berkembang menjadi pribadi yang solutif, peka terhadap permasalahan di sekitar dan aktif berorganisasi
6. Teman-teman mahasiswa Jurusan Sistem Informasi dan jurusan lainnya dari Fakultas Ilmu Komputer yang tidak dapat penulis sebutkan namanya satu per satu, yang telah menjadi teman yang baik bagi penulis, senantiasa memberikan dukungan terhadap sistem informasi yang dikembangkan serta dukungan moril dan motivasi, sehingga mampu terus menyemangati penulis untuk segera menyelesaikan penelitian yang dikerjakan.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan.

Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 02 Januari 2019

Penulis

[auliaspertiwi@gmail.com](mailto:auliaspertiwi@gmail.com)



## ABSTRAK

**Aulia Septi Pertiwi , Pengembangan Sistem Informasi *Review Smartphone* (Studi Pada TNT CELL Bojonegoro)**

**Pembimbing: Fajar Pradana, S.ST, M.Eng dan Dr.Eng. Fitra A. Bachtiar, S.T, M.Eng**

Teknologi komunikasi pada masa sekarang sudah berkembang semakin pesat, salah satunya adalah *smartphone*. Dari survei yang telah dilakukan pada 44 pengguna *smartphone*, 88% dari responden mengatakan membaca review tentang *Smartphone* yang diinginkan sebelum membeli sebuah *smartphone*. Review dapat tersebar pada berbagai website. TNT Cell merupakan salah satu *counter handphone* yang menjual berbagai merk *smartphone*. Salah satu proses bisnis yang ada pada TNT Cell adalah proses konsultasi antara calon konsumen dengan petugas mengenai sebuah *smartphone*. Pada proses ini petugas akan menjelaskan mengenai deskripsi dari *smartphone* seperti fitur yang terdapat pada *smartphone*. Beberapa konsumen banyak yang tidak dapat memahami dengan baik mengenai kelebihan dan kekurangan dari sebuah *smartphone*, meskipun sudah dijelaskan oleh petugas. Hal ini mempengaruhi proses pengambilan keputusan konsumen dalam hal pembelian sebuah *smartphone*. Berdasarkan permasalahan tersebut dikembangkanlah Sistem Informasi *Review Smartphone*. Sistem ini dapat memberikan informasi mengenai *smartphone* dengan cara menampilkan *review* dari beberapa situs web dan melakukan analisis sentimen pada *review*. Metode *Web Scraping* digunakan pada sistem ini untuk melakukan ekstraksi *review* dari situs Priceprice.com, Pricebook.co.id, dan Iprice.co.id. Dalam proses analisis sentimen *review* digunakan salah satu metode klasifikasi yaitu *Support Vector Machine*. Metode pengembangan perangkat lunak yang digunakan sistem informasi ini adalah metode *Waterfall model*. Sistem ini memiliki dua fitur utama yaitu *scraping* spesifikasi dan *review* serta memproses *review*. Pengujian sistem ini dilakukan dengan menggunakan teknik pengujian validasi, pengujian *compatibility*, dan pengujian *usability*. Hasil dari pengujian validasi adalah keseluruhan kasus pengujian yang diberikan bernilai valid, hal ini menunjukkan bahwa sistem telah dibangun sesuai dengan kebutuhan penggunaannya.

Kata kunci: Sistem Informasi, *Web Scraping*, Analisis Sentimen, *Support Vector Machine*, *Waterfall*



## ABSTRACT

**Aulia Septi Pertiwi, Pengembangan Sistem Informasi *Review Smartphone* (Studi Pada TNT CELL Bojonegoro)**

**Supervisors: Fajar Pradana, S.ST, M.Eng dan Dr.Eng.Fitra A.Bachtiar, S.T, M.Eng**

*Communication technology is growing really fast at this era, one of communication technology is a smartphone. Based on a survey that conducted on 44 smartphone users, 88% of respondents said, they read reviews before buying a smartphone. They can find reviews about smartphone on various websites. TNT Cell is one of the mobile counter that sells various brands of smartphone. One of business process at TNT CELL is the consultation process about smartphone between customer and the officer. The officer will explain the specifications of smartphone that customer wants. Some of the customers do not understand well about the advantages and disadvantages of the smartphone that they want, although it has been explained by the officer. This affects is the customer will be difficult to make a decision of that smartphone. Based on these problems an information system about smartphone review was developed. This system will give the customer information about smartphones by review from several websites and conducting sentiment analysis on reviews. Web scraping methods are used in this system to extract reviews from Priceprice.com, Pricebook.co.id, and Iprice.co.id sites. In the process of sentiment analysis, one of the classification methods that used was Support Vector Machine. The software development method that used in this research is the Waterfall model method. This system has two main features, they are scraping specifications and reviews and processing reviews. Testing phase of this information system was using validation testing, compatibility testing, and usability testing techniques. The result of validation testing is all of the given test case are valid, that mean the system has been built appropriate with the user requirement.*

**Keywords :** *Information System, Web Scraping, Sentiment Analysis, Support Vector Machine, Waterfall*

## DAFTAR ISI

PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
PRAKATA.....	iv
ABSTRAK.....	vi
ABSTRACT .....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR.....	xvi
DAFTAR LAMPIRAN .....	xviii
<b>BAB 1 PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan .....	3
1.4 Manfaat.....	4
1.5 Batasan Masalah.....	4
1.6 Sistematika Pembahasan.....	4
<b>BAB 2 LANDASAN KEPUSTAKAAN .....</b>	<b>7</b>
2.1 Kajian Pustaka .....	7
2.2 Review.....	7
2.3 Sistem Informasi .....	8
2.4 <i>Software Development Life Cycle (SDLC)</i> .....	8
2.4.1 Model <i>Waterfall</i> .....	8
2.5 <i>Website</i> .....	10
2.6 PHP .....	10
2.7 MySQL.....	11
2.8 CodeIgniter .....	11
2.9 Business Process Modelling.....	12
2.9.1 <i>Business Process Modelling Notation (BPMN)</i> .....	12
2.10 <i>Unified Modeling Language (UML)</i> .....	14
2.10.1 <i>Use Case Diagram</i> .....	15





2.10.2	<i>Sequence Diagram</i> .....	16
2.10.3	<i>Class Diagram</i> .....	17
2.11	<i>Web Scraping</i> .....	19
2.12	<i>Text Mining</i> .....	19
2.13	<i>Preprocessing Text</i> .....	19
2.13.1	<i>Tokenisasi</i> .....	20
2.13.2	<i>Data Cleaning</i> .....	20
2.13.3	<i>Case Folding</i> .....	20
2.13.4	<i>Filterisasi</i> .....	21
2.13.5	<i>Stemming</i> .....	21
2.14	<i>Algoritme TF-IDF (Term Frequency-Inverse Document)</i> .....	23
2.15	<i>Sentiment Analysis</i> .....	24
2.16	<i>Support Vector Machine</i> .....	25
2.17	<i>Black Box Testing</i> .....	26
2.17.1	<i>Validation Testing</i> .....	26
2.17.2	<i>Compatibility Testing</i> .....	26
2.18	<i>Usability Testing</i> .....	27
2.18.1	<i>Metrik Effectiveness Usability Testing</i> .....	27
2.18.2	<i>Metrik Efficiency Usability Testing</i> .....	27
2.18.3	<i>Metrik Satisfaction Usability Testing</i> .....	28
2.18.4	<i>Skala Likert</i> .....	29
BAB 3	<b>METODOLOGI PENELITIAN</b> .....	31
3.1	<i>Studi Literatur</i> .....	31
3.2	<i>Analisis Kebutuhan</i> .....	32
3.3	<i>Perancangan Sistem</i> .....	32
3.4	<i>Implementasi</i> .....	32
3.5	<i>Pengujian dan Analisis Hasil</i> .....	33
3.6	<i>Kesimpulan dan Saran</i> .....	33
BAB 4	<b>ANALISIS KEBUTUHAN SISTEM</b> .....	34
4.1	<i>Profil TNT Cell Bojonegoro</i> .....	34
4.2	<i>Analisis Proses Bisnis</i> .....	34
4.2.1	<i>Identifikasi Proses Bisnis Saat Ini</i> .....	34

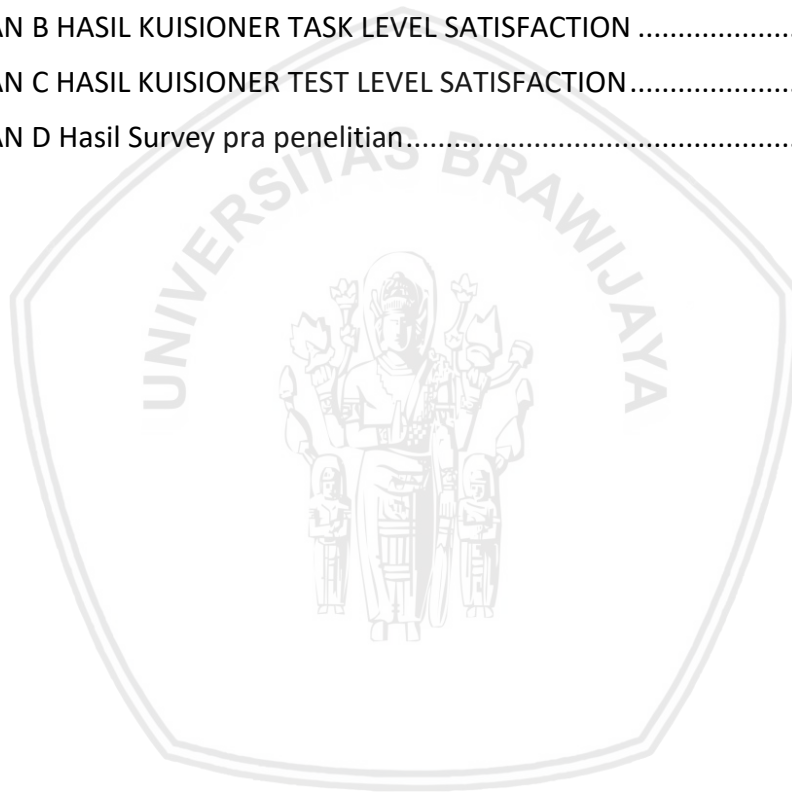
4.2.2 Proses Bisnis Yang Diusulkan .....	35
4.3 Analisis Aktor .....	36
4.3.1 Aturan Penomoran Aktor .....	36
4.3.2 Identifikasi Aktor .....	36
4.4 Analisis Permasalahan .....	37
4.5 Analisis Kebutuhan .....	37
4.5.1 Aturan Penomoran Kebutuhan .....	38
4.5.2 Analisis Kebutuhan User .....	38
4.5.3 Analisis Spesifikasi Kebutuhan Perangkat Lunak .....	40
4.6 Analisis Hasil Rekayasa Kebutuhan .....	42
4.7 Pemodelan Kebutuhan .....	42
4.7.1 Pemodelan <i>Use Case Diagram</i> .....	42
4.7.2 Pemodelan <i>Use Case Scenario</i> .....	43
<b>BAB 5 PERANCANGAN</b> .....	<b>50</b>
5.1 Perancangan Arsitektur Sistem .....	50
5.2 Perancangan <i>Sequence Diagram</i> .....	50
5.2.1 <i>Sequence Diagram Login</i> .....	51
5.2.2 <i>Sequence Diagram Menambah Data Smartphone</i> .....	51
5.2.3 <i>Sequence Diagram Mengubah Data Smartphone</i> .....	52
5.2.4 <i>Sequence Diagram Menghapus Data Smartphone</i> .....	53
5.2.5 <i>Sequence Diagram Memproses Review</i> .....	54
5.2.6 <i>Sequence Diagram Mencari Data Smartphone</i> .....	54
5.2.7 <i>Sequence Diagram Melihat Detail Smartphone</i> .....	55
5.2.8 <i>Sequence Diagram Scraping Spesifikasi dan Review</i> .....	55
5.2.9 <i>Sequence Diagram Melihat Review</i> .....	57
5.2.10 <i>Sequence Diagram Logout</i> .....	57
5.2.11 <i>Sequence Diagram Menambah User</i> .....	58
5.3 Pemodelan <i>Class Diagram</i> .....	58
5.3.1 Diagram Relasi Antar Kelas .....	59
5.3.2 Diagram Relasi Antar Kelas <i>Controller</i> .....	59
5.3.3 Diagram Relasi Antar Kelas Model .....	60
5.4 Perancangan Skema Basis Data .....	61



5.5 Perancangan Antar Muka Pengguna .....	62
5.5.1 Rancangan Antar Muka Halaman <i>Login</i> .....	62
5.5.2 Rancangan Antar Muka Halaman Tambah dan Rubah <i>Data Smartphone</i> .....	63
5.5.3 Rancangan Antar Muka Halaman <i>Review</i> .....	63
5.5.4 Rancangan Antar Muka Halaman Daftar <i>Smartphone</i> .....	64
5.5.5 Rancangan Antar Muka Halaman <i>Home</i> .....	65
5.5.6 Rancangan Antar Muka Halaman <i>Detail Smartphone</i> .....	65
5.5.7 Rancangan Antar Muka Halaman Tambah <i>User</i> .....	66
5.6 Perancangan Algoritme .....	67
5.6.1 Algoritme <i>Scraping</i> Spesifikasi dan <i>Review</i> .....	67
5.6.2 Algoritme Menambah Data <i>Smartphone</i> .....	67
5.6.3 Algoritme Memproses <i>Review</i> .....	68
5.6.4 Algoritme Merubah Data <i>Smartphone</i> .....	68
5.6.5 Algoritme Menghapus Data <i>Smartphone</i> .....	69
5.6.6 Algoritme Menampilkan Detail <i>Smartphone</i> .....	69
BAB 6 IMPLEMENTASI .....	71
6.1 Batasan Implementasi .....	71
6.2 Implementasi Program .....	71
6.2.1 Implementasi <i>Website</i> .....	71
6.2.2 Implementasi Data Definition Language .....	83
6.2.3 Implementasi Antarmuka .....	84
BAB 7 Pengujian dan analisis hasil .....	90
7.1 Pengujian <i>Black Box</i> .....	90
7.1.1 Pengujian Validasi <i>Scraping</i> Spesifikasi dan <i>Review</i> .....	90
7.1.2 Pengujian Validasi Menambah Data <i>Smartphone</i> .....	91
7.1.3 Pengujian Validasi Memproses <i>Review</i> .....	92
7.1.4 Pengujian Validasi Merubah Data <i>Smartphone</i> .....	93
7.1.5 Pengujian Validasi Menghapus Data <i>Smartphone</i> .....	94
7.1.6 Pengujian Validasi Melihat <i>Detail Smartphone</i> .....	95
7.2 Pengujian <i>Compatibility</i> .....	96
7.3 Pengujian <i>Usability</i> .....	96



7.3.1 Metrik <i>Effectiveness</i> .....	97
7.3.2 Metrik <i>Efficiency</i> .....	100
7.3.3 Metrik <i>Satisfaction</i> .....	102
BAB 8 Penutup .....	109
8.1 Kesimpulan.....	109
8.2 Saran .....	110
DAFTAR REFERENSI .....	111
LAMPIRAN A WAWANCARA PEMILIK TNT CELL BOJONEGORO.....	114
LAMPIRAN B HASIL KUISIONER TASK LEVEL SATISFACTION .....	116
LAMPIRAN C HASIL KUISIONER TEST LEVEL SATISFACTION.....	119
LAMPIRAN D Hasil Survey pra penelitian.....	132



## DAFTAR TABEL

Tabel 2.1 Notasi Dasar BPMN .....	13
Tabel 2.2 Simbol-Simbol <i>Use Case Diagram</i> .....	15
Tabel 2.3 Simbol-Simbol Dalam <i>Sequence Diagram</i> .....	16
Tabel 2.4 Simbol-Simbol Dalam <i>Class Diagram</i> .....	18
Tabel 2.5 Ilustrasi Proses Tokenisasi .....	20
Tabel 2.6 Ilustrasi Proses <i>Data Cleaning</i> .....	20
Tabel 2.7 Ilustrasi Proses <i>Case Folding</i> .....	21
Tabel 2.8 Ilustrasi Proses <i>Filterisasi</i> .....	21
Tabel 2.9 Ilustrasi Proses <i>Stemming</i> .....	22
Tabel 2.10 Bobot Nilai Jawaban .....	29
Tabel 2.11 Persentase Nilai Jawaban .....	30
Tabel 4.1 Identifikasi Aktor .....	37
Tabel 4.2 Identifikasi Masalah.....	37
Tabel 4.3 Identifikasi Kebutuhan User .....	38
Tabel 4.4 Identifikasi Kebutuhan Sistem.....	39
Tabel 4.5 Identifikasi Spesifikasi Kebutuhan Fungsional .....	40
Tabel 4.6 Identifikasi Spesifikasi Kebutuhan <i>Non Fungsional</i> .....	42
Tabel 4.7 <i>Use Case Scenario Login</i> .....	43
Tabel 4.8 <i>Use Scenario Menambah Data Smartphone</i> .....	44
Tabel 4.9 <i>Use Case Scenario Merubah Data Smartphone</i> .....	45
Tabel 4.10 <i>Use Case Scenario Menghapus Smartphone</i> .....	45
Tabel 4.11 <i>Use Case Scenario Memproses Review</i> .....	46
Tabel 4.12 <i>Use Case Scenario Mencari Data Smartphone</i> .....	46
Tabel 4.13 <i>Use Case Scenario Melihat Detail Smartphone</i> .....	47
Tabel 4.14 <i>Use Case Scenario Scraping Spesifikasi dan Review</i> .....	47
Tabel 4.15 <i>Use Case Scenario Melihat Review</i> .....	48
Tabel 4.16 <i>Use Case Scenario Logout</i> .....	48
Tabel 4.17 <i>Use Case Scenario Menambah User</i> .....	49
Tabel 5.1 Perancangan Algoritme <i>Scraping Spesifikasi dan Review</i> .....	67
Tabel 5.2 Menambah <i>Data Smartphone</i> .....	67

Tabel 5.3 Perancangan Algoritme Memproses <i>Review</i> .....	68
Tabel 5.4 Perancangan Algoritme Merubah Data <i>Smartphone</i> .....	68
Tabel 5.5 Perancangan Algoritme Menghapus Data <i>Smartphone</i> .....	69
Tabel 5.6 Perancangan Algoritme Menampilkan Detail <i>Smartphone</i> .....	70
Tabel 6.1 Implementasi <i>Scraping</i> Spesifikasi dan <i>Review</i> .....	72
Tabel 6.2 Implementasi <i>Controller DataController</i> Menambah Data <i>Smartphone</i> .....	76
Tabel 6.3 Implementasi Model Data <i>Smartphone</i> Menambah Data <i>Smartphone</i> .....	77
Tabel 6.4 Implementasi <i>Controller DataController</i> Memproses <i>Review</i> .....	78
Tabel 6.5 Implementasi Model <i>SentimentModel</i> Memproses <i>Review</i> .....	80
Tabel 6.6 Implementasi <i>Controller DataController</i> Merubah Data <i>Smartphone</i> .	80
Tabel 6.7 Implementasi Model Data <i>Smartphone</i> Merubah Data <i>Smartphone</i> ....	80
Tabel 6.8 Implementasi <i>Controller DataController</i> Menghapus Data <i>Smartphone</i> .....	81
Tabel 6.9 Implentasi Model Data <i>Smartphone</i> Menghapus Data <i>Smartphone</i> ....	81
Tabel 6.10 Implementasi <i>Controller Home</i> Menampilkan Detail <i>Smartphone</i> ....	81
Tabel 6.11 Implementasi Model Data <i>Smartphone</i> Fungsi <i>GetDetailSmartphone</i>	82
Tabel 6.12 Implementasi Model Data <i>Smartphone</i> Fungsi <i>updateViews</i> .....	83
Tabel 6.13 Implementasi Model <i>SentimentModel</i> .....	83
Tabel 6.14 Implementasi DDL Tabel <i>Smartphone</i> .....	84
Tabel 6.15 Implementasi DDL Tabel <i>Review</i> .....	84
Tabel 6.16 Implementasi DDL Tabel <i>Review</i> .....	84
Tabel 6.17 Implementasi DDL Tabel <i>Situs_Web</i> .....	84
Tabel 7.1 Kasus Uji Pengujian Validasi <i>Scraping</i> Spesifikasi dan <i>Review</i> .....	90
Tabel 7.2 Kasus Uji Pengujian Validasi Menambah Data <i>Smartphone</i> .....	91
Tabel 7.3 Kasus Uji Pengujian Validasi Memproses <i>Review</i> .....	92
Tabel 7.4 Kasus Uji Pengujian Validasi Merubah Data <i>Smartphone</i> .....	93
Tabel 7.5 Kasus Uji Pengujian Validasi Menghapus Data <i>Smartphone</i> .....	94
Tabel 7.6 Kasus Uji Pengujian Validasi Melihat <i>Detail Smartphone</i> .....	95
Tabel 7.7 Tabel Hasil Pengujian Validasi Sistem Informasi <i>Review Smartphone</i> ..	96
Tabel 7.8 Daftar <i>Task Usability Testing</i> Pengguna Admin .....	97
Tabel 7.9 Hasil Keberhasilan <i>Task</i> Pengguna Admin .....	99

Tabel 7.10 Daftar <i>Task Usability Testing</i> Pengguna Konsumen.....	99
Tabel 7.11 Hasil Keberhasilan <i>Task</i> Pengguna Konsumen.....	99
Tabel 7.12 Hasil Pengukuran Waktu <i>Task</i> Pengguna Admin .....	100
Tabel 7.13 Perhitungan Metrik <i>Efficiency</i> Pengguna Konsumen.....	101
Tabel 7.14 Kuisisioner <i>Task Level Satisfaction</i> Pengguna Admin .....	103
Tabel 7.15 Hasil Kuisisioner <i>Test Level Satisfaction</i> Pengguna Konsumen .....	105





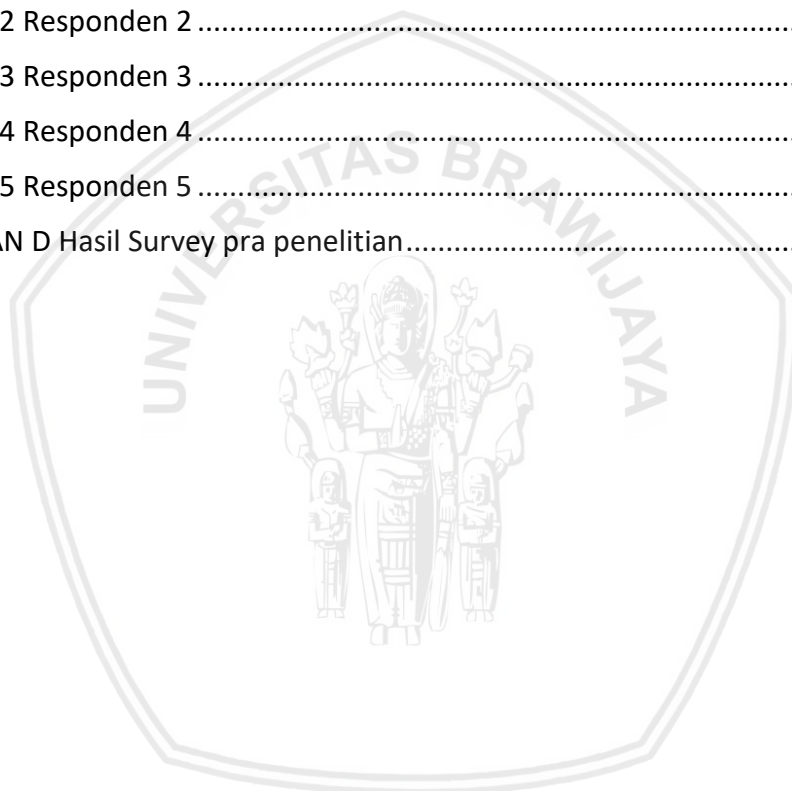
## DAFTAR GAMBAR

Gambar 2.1 Langkah-Langkah Model <i>Waterfall</i> .....	9
Gambar 2.2 Ilustrasi Algoritma <i>Support Vector Machine</i> .....	25
Gambar 2.3 Contoh Skala Likert.....	29
Gambar 3.1 Diagram alur metodologi penelitian .....	31
Gambar 4.1 Proses Bisnis Mencari Dan Membaca Review Smartphone Saat Ini.	34
Gambar 4.2 Proses Bisnis Konsultasi <i>Smartphone</i> Saat Ini.....	35
Gambar 4.3 Proses Bisnis Memasukkan Data <i>Smartphone</i> .....	35
Gambar 4.4 Proses Bisnis Membaca <i>Review</i> Yang Diusulkan.....	36
Gambar 4.5 Aturan Penomoran Kode Aktor.....	36
Gambar 4.6 Aturan Penomoran Kode Kebutuhan.....	38
Gambar 4.7 <i>Use Case Diagram</i> Sistem <i>Review</i> .....	43
Gambar 5.1 Rancangan Arsitektur Diagram Sistem Informasi <i>Review Smartphone</i> .....	50
Gambar 5.2 <i>Sequence Diagram</i> Login.....	51
Gambar 5.3 <i>Sequence Diagram</i> Menambah Data .....	52
Gambar 5.4 <i>Sequence Diagram</i> Mengubah Data <i>Smartphone</i> .....	53
Gambar 5.5 <i>Sequence Diagram</i> Menghapus Data <i>Smartphone</i> .....	53
Gambar 5.6 <i>Sequence Diagram</i> Memproses <i>Review</i> .....	54
Gambar 5.7 <i>Sequence Diagram</i> Mencari Data <i>Smartphone</i> .....	55
Gambar 5.8 <i>Sequence Diagram</i> Melihat Detail <i>Smartphone</i> .....	55
Gambar 5.9 <i>Sequence Diagram Scraping</i> Spesifikasi dan <i>Review</i> .....	56
Gambar 5.10 <i>Sequence Diagram</i> Melihat <i>Review</i> .....	57
Gambar 5.11 <i>Sequence Diagram</i> Logout .....	58
Gambar 5.12 <i>Sequence Diagram</i> Menambah <i>User</i> .....	58
Gambar 5.13 <i>Class Diagram</i> Relasi Antar Kelas .....	59
Gambar 5.14 Diagram Relasi Antar Kelas <i>Controller</i> .....	60
Gambar 5.15 Diagram Relasi Antar Kelas Model.....	61
Gambar 5.16 <i>Physical Data Model</i> Sistem Informasi <i>Review Smartphone</i> .....	62
Gambar 5.17 Rancangan Antar Muka <i>Login</i> .....	63

Gambar 5.18 Rancangan Antar Muka Halaman Menambah dan Merubah <i>Data Smartphone</i> .....	63
Gambar 5.19 Rancangan Antar Muka Halaman Review .....	64
Gambar 5.20 Rancangan Antar Muka Halaman Daftar <i>Smartphone</i> .....	65
Gambar 5.21 Rancangan Antar Muka Halaman <i>Home</i> .....	65
Gambar 5.22 Rancangan Antar Muka Halaman <i>Detail Smartphone</i> .....	66
Gambar 5.23 Rancangan Antar Muka Halaman Tambah <i>User</i> .....	66
Gambar 6.1 Implementasi Antarmuka Halaman Login.....	85
Gambar 6.2 Implementasi Antarmuka Halaman <i>DataSmartphonePage</i> .....	86
Gambar 6.3 Implementasi Antarmuka Halaman <i>ReviewPage</i> .....	86
Gambar 6.4 Implementasi Antarmuka Halaman <i>EditDataPage</i> .....	87
Gambar 6.5 Implementasi Antarmuka Halaman <i>Home</i> .....	88
Gambar 6.6 Implementasi Antarmuka Halaman <i>DetailSmartphone</i> .....	89
Gambar 7.1 Aturan penomoran pengujian validasi.....	90
Gambar 7.2 Kategori Masalah Kompabilitas pada <i>SortSite</i> .....	96
Gambar 7.3 Hasil <i>Compatibility Testing</i> .....	96
Gambar 7.4 Grafik Batang Nilai TBE Pengguna Konsumen.....	102

## DAFTAR LAMPIRAN

LAMPIRAN A WAWANCARA PEMILIK TNT CELL BOJONEGORO.....	114
A.1 Wawancara pemilik TNT CELL Bojonegoro.....	114
LAMPIRAN B HASIL KUISIONER TASK LEVEL SATISFACTION .....	116
B.1.1 Hasil kuisioner task level satisfaction.....	116
LAMPIRAN C HASIL KUISIONER TEST LEVEL SATISFACTION.....	119
C.1 Responden 1 .....	119
C.2 Responden 2 .....	122
C.3 Responden 3 .....	125
C.4 Responden 4 .....	127
C.5 Responden 5 .....	130
LAMPIRAN D Hasil Survey pra penelitian.....	132



## BAB 1 PENDAHULUAN

### 1.1 Latar Belakang

Masyarakat sekarang ini berlomba-lomba untuk memiliki *smartphone* tidak hanya semata-mata sebagai alat komunikasi, melainkan sekaligus sebagai gaya hidup, tren, dan prestise (Kogoya, 2015). Bukan hanya masyarakat, produsen *smartphone* kini juga berlomba-lomba dalam menghadirkan *smartphone* dengan tipe dan harga yang bervariasi. Setiap tipe memiliki kelebihan dan kekurangan masing-masing pada spesifikasi *hardware*, kamera, atau sistem operasi yang digunakan. Dari survey yang telah dilakukan pada 44 pengguna *smartphone*, 88% dari responden mengatakan membaca review tentang *smartphone* yang diinginkan sebelum membeli *smartphone* tersebut.

Review adalah opini pada dan/atau pengalaman konsumen terhadap sebuah produk, layanan atau bisnis (endorsement, 2015). Review dari konsumen biasanya berisi opini mereka mengenai sebuah produk dengan melihat berbagai aspek yang ada pada produk tersebut. Umumnya, aspek sebuah produk merupakan karakteristik dari produk tersebut, seperti contohnya, Samsung Galaxy S6 memiliki aspek seperti baterai, kualitas gambar, harga dan sebagainya (Sangeetha, et al., 2017). Opini yang merupakan isi dari review inilah yang membuat setiap review dapat berbeda dengan review yang lainnya. Selain itu, review produk *smartphone* tidak hanya ditulis pada *website* produsen atau vendor *Smartphone* tersebut. Review dapat tersebar pada berbagai *website*, bahkan review *Smartphone* dapat ditemukan pada berbagai blog pribadi. Belum terintegrasinya review, menyebabkan calon konsumen kesulitan dalam merangkum review karena harus membuka satu persatu *website* untuk membaca review *smartphone* yang diinginkannya.

Selain dengan melihat review di internet terlebih dahulu, beberapa konsumen memilih untuk langsung mendatangi toko penjual *handphone* atau biasa disebut sebagai *counter handphone*. Menurut Heru Sutadi seorang pengamat telekomunikasi Indonesia, “masih banyak masyarakat Indonesia yang lebih memilih untuk membeli *smartphone* secara langsung ketimbang melalui situs belanja *online*” (Wirartri, 2015). Salah satu *counter handphone* di Bojonegoro adalah TNT Cell. TNT Cell merupakan salah satu *counter handphone* yang menjual berbagai merk *smartphone*. Salah satu proses bisnis yang ada pada TNT Cell adalah proses konsultasi antara calon konsumen dengan petugas mengenai sebuah *smartphone*. Pada proses ini petugas akan menjelaskan mengenai deskripsi dari *smartphone* seperti fitur yang terdapat pada *smartphone*. Sering kali untuk menjelaskan satu produk kepada satu konsumen membutuhkan waktu lebih dari 30 Menit, hal ini menyebabkan ketika *counter* ramai beberapa konsumen harus menunggu lama untuk dilayani. Selain itu beberapa konsumen juga banyak yang belum dapat memahami dengan baik mengenai kelebihan dan kekurangan dari sebuah *smartphone*, meskipun sudah dijelaskan oleh petugas. Hal ini mempengaruhi proses pengambilan keputusan konsumen dalam hal pembelian

sebuah *smartphone*. Oleh karena itu, pemilik *counter* TNT menginginkan sebuah sistem yang dapat menampilkan informasi lengkap mengenai sebuah *smartphone* dan ulasan (*review*) dari *smartphone* tersebut. Untuk mempermudah konsumen dan petugas, diharapkan sistem juga dapat memberikan informasi apakah ulasan merupakan ulasan baik atau ulasan buruk.

Solusi yang dapat diterapkan untuk permasalahan diatas adalah dengan mengembangkan sistem informasi yang dapat mengumpulkan *review Smartphone* dari berbagai *website* dan menampilkannya pada satu *website* tersendiri. Salah satu metode untuk mengumpulkan data *review* adalah dengan menggunakan metode *web scraping*. *Web Scraping* adalah proses pengambilan sebuah dokumen semi-terstruktur dari internet, umumnya berupa halaman web dalam bahasa markup seperti HTML atau XHTML, dan menganalisis dokumen tersebut untuk mengekstrak data spesifik yang ada pada halaman untuk digunakan pada kepentingan lain (Turland, 2010). Dengan menggunakan *web scraping* jumlah data yang dapat di akses lebih besar dibandingkan dengan metode pengumpulan data konsumen tersentralisasi yang dilakukan oleh Institut Statistik Nasional Italia (Polidoro, et al., 2015). Oleh karena itu pada penelitian ini *web scraping* digunakan untuk mengumpulkan *review-review Smartphone* yang terdapat pada berbagai situs baik *e-commerce*, *website* vendor, bahkan *review* pada blog pribadi. Lalu *review-review* dari hasil *scraping* akan diintegrasikan ke dalam satu *website* tersendiri, sehingga calon konsumen dapat membaca *review* dari berbagai situs tanpa harus membuka satu persatu situs tersebut.

Namun metode *web scraping* hanya digunakan untuk mengumpulkan *review-review* dari berbagai situs dan untuk ditampilkan ke dalam satu *website* tersendiri, sehingga *review-review* yang ada masih bersifat parsial. Dimana *review-review* hasil dari *scraping* belum dikelompokkan berdasarkan informasi yang ada pada *review* tersebut. Perbedaan informasi atau opini yang terkandung pada setiap *review*, menyulitkan konsumen dalam merangkum informasi dari produk *Smartphone* yang diinginkannya. Untuk itu, perlu adanya rangkuman dari *review-review* yang sudah dikumpulkan sebelumnya. Permasalahan tersebut dapat diselesaikan dengan melakukan *sentiment analysis* terhadap *review-review* yang sudah dikumpulkan.

*Sentiment analysis* atau *opinion mining* adalah studi komputasi yang menganalisis opini, penilaian dan emosi seseorang terhadap suatu entitas seperti produk, layanan, organisasi, ataupun sebuah kejadian beserta atribut yang dimilikinya (Liu, 2010). *Sentiment analysis* merupakan salah satu aktivitas klasifikasi yang mengelompokkan opini atau sentimen sebuah data teks ke dalam kategori yaitu positif atau negatif (Medhat, et al., 2014). *Review* dapat diklasifikasikan ke dalam klasifikasi positif dan negatif (Sangeetha, et al., 2017). Dengan mengklasifikasi konsumen kedalam positif dan negatif, baik petugas maupun konsumen dapat lebih mudah dalam merangkum informasi dari *review* yang ada dan dapat membantu dalam proses pengambilan keputusan calon konsumen.

Dalam membangun perangkat lunak, terdapat beberapa tahapan-tahapan aktivitas yang digunakan sebagai metode pembangunan sebuah perangkat lunak. Hal ini dilakukan agar aktivitas dalam pembangunan sebuah perangkat lunak dapat berjalan dengan sistematis. Tahapan aktivitas atau metode tersebut dikenal sebagai SDLC (*Software Development Life Cycle*). SDLC memiliki berbagai macam metode pengembangan, salah satunya adalah model *Waterfall*. Model *Waterfall* sangat cocok digunakan ketika kualitas dari perangkat lunak merupakan hal terpenting dalam proses pembangunan (Alshamrani & Bahattab, 2015). Model *Waterfall* memiliki beberapa tahapan aktivitas yaitu *Requirement analysis and definition, System and software design, Implementation and unit testing, Integration and system testing, dan Operation and maintenance* (Sommerville, 2011).

Berdasarkan latar belakang yang sudah diuraikan, penulis tertarik untuk melakukan penelitian dengan judul “Pengembangan Sistem Informasi Review *Smartphone* Studi Pada TNT Cell Bojonegoro”. Diharapkan hasil penelitian ini dapat mempermudah petugas TNT Cell dalam menjelaskan produk *Smartphone* kepada calon konsumen sehingga proses jual beli dapat lebih cepat dan memudahkan konsumen dalam memahami produk *smartphone* yang diinginkannya.

## 1.2 Rumusan Masalah

Berdasarkan permasalahan yang telah dipaparkan pada latar belakang, maka rumusan masalah untuk penelitian ini adalah sebagai berikut :

1. Bagaimanakah hasil analisis dan spesifikasi persyaratan sistem informasi review *Smartphone* di TNT Cell yang sesuai dengan kebutuhan organisasi tersebut?
2. Bagaimanakah hasil rancangan sistem perangkat lunak yang sesuai dengan spesifikasi persyaratan sistem tersebut?
3. Bagaimanakah hasil implementasi sistem informasi yang sesuai dengan rancangan sistem yang dibuat?
4. Bagaimana hasil pengujian sistem *website* untuk sistem informasi review *Smartphone* di TNT Cell?

## 1.3 Tujuan

Berdasarkan permasalahan yang diteliti, maka tujuan yang ingin dicapai dari penelitian ini adalah :

1. Menganalisis dan menyusun spesifikasi persyaratan sistem informasi review *smartphone* untuk TNT Cell.
2. Merancang sistem informasi sesuai persyaratan untuk sistem informasi tersebut.
3. Mengimplementasikan rancangan sistem informasi tersebut.
4. Menguji sistem informasi tersebut secara fungsional dan non fungsional.



## 1.4 Manfaat

Manfaat yang diharapkan dari adanya penelitian ini adalah sebagai berikut.

Bagi Mahasiswa :

1. Dapat menerapkan ilmu yang telah didapatkan selama perkuliahan salah satunya dengan menerapkan sistem informasi untuk mendukung proses bisnis pada suatu organisasi.

Bagi Organisasi :

1. Dapat membantu dalam meningkatkan pelayanan terhadap konsumen
2. Dapat menambah nilai keunggulan organisasi

Bagi konsumen :

1. Dapat mempermudah dalam memahami suatu produk, khususnya *Smartphone*
2. Dapat mempermudah dalam proses pengambilan keputusan

## 1.5 Batasan Masalah

Batasan masalah dari penelitian ini adalah :

1. Proses implementasi perangkat lunak pada penelitian ini menggunakan *framework* CodeIgniter
2. Sumber review yang akan diintegrasikan adalah review yang terdapat pada situs priceprice.com, prickebook.com, dan lprice.com
3. Penelitian ini menggunakan *library DOM* untuk mengimplementasikan metode *web scraping* yang sudah tersedia untuk *framework* CodeIgniter
4. Penelitian ini hanya melakukan *sentiment analysis* terhadap review *Smartphone* yang menggunakan Bahasa Indonesia
5. Klasifikasi review menggunakan algoritma *Support Vector Machine*.

## 1.6 Sistematika Pembahasan

Dalam sistematika penulisan penelitian ini dibagi menjadi beberapa bab yang saling berkaitan dengan pembagian materi sebagai berikut :

### BAB 1 PENDAHULUAN

Bab ini menjelaskan mengenai penjelasan umum dari penelitian ini, meliputi latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, serta sistematika pembahasan penelitian.

### BAB 2 LANDASAN KEPUSTAKAAN

Bab ini menjelaskan tentang teori-teori dan konsep yang dijadikan sebagai referensi dalam melakukan penelitian. Teori dan konsep tersebut diambil dari berbagai sumber pustaka seperti penelitian sejenis yang pernah dilakukan, buku, jurnal, artikel dan situs web. Teori dan konsep yang diperlukan diantaranya *Software Development Lifecycle (SDLC)*, pengembangan perangkat



lunak dengan metode *waterfall*, pemodelan dan perancangan sistem, pola perancangan MVC, *framework CodeIgniter*, teknik *web scraping*, analisis sentimen dan algoritme yang digunakan serta metode pengujian sistem yang akan digunakan dalam penelitian ini.

### **BAB 3 METODOLOGI PENELITIAN**

Bab ini memaparkan tahapan-tahapan yang dilakukan untuk menyelesaikan topik masalah yang diangkat pada penelitian ini.

### **BAB 4 ANALISIS KEBUTUHAN SISTEM**

Bab ini menjelaskan mengenai tahapan analisis dan indentifikasi kebutuhan dari sistem yang akan dikembangkan.

### **BAB 5 PERANCANGAN SISTEM**

Bab ini membahas tentang perancangan sistem menggunakan diagram UML seperti diagram *Use Case*, *Activity Diagram*, *Sequence Diagram*, dan *Class Diagram*. Pada bagian ini juga dibahas tentang perancangan basis data, dan perancangan *user interface* dari sistem yang akan dibangun.

### **BAB 6 IMPLEMENTASI**

Pada bab ini membahas hasil analisis dan perancangan sistem yang diimplementasikan ke dalam bentuk kode program. Seluruh hasil perancangan akan diimplementasikan termasuk basis data dan antar muka dari sistem.

### **BAB 7 PENGUJIAN**

Bab ini berisi tentang pengujian sistem dengan menggunakan metode pengujian *Black Box* dan *Usability*.

## **BAB 8 PENUTUP**

Pada bab ini terdapat kesimpulan yang didapatkan dari hasil penelitian dan saran untuk penelitian selanjutnya.



## BAB 2 LANDASAN KEPUSTAKAAN

### 2.1 Kajian Pustaka

Penelitian pertama yang dijadikan sebagai kajian pustaka pada penelitian ini adalah mengenai kegunaan teknik *web scraping* dalam survei harga pasar serta tingkat inflasi di Italia dengan Studi Pada harga barang dan tiket pesawat. Dari penelitian tersebut didapatkan beberapa kesimpulan diantaranya yaitu dengan menggunakan teknik *web scraping* untuk survei inflasi, didapatkan hasil dimana jumlah data dapat yang diakses lebih besar dibandingkan dengan metode yang digunakan sebelumnya yaitu metode pengumpulan data konsumen tersentralisasi yang dilakukan oleh Institut Statistik Nasional Italia (Polidoro, et al., 2015).

Jurnal penelitian kedua yang dijadikan sebagai kajian pustaka pada penelitian ini adalah penelitian dilakukan untuk mengetahui tingkat partisipasi memasuki perguruan tinggi di Indonesia. Metode yang digunakan untuk menyelesaikan akar permasalahan yang dihadapi pada penelitian tersebut adalah dengan membangun sebuah situs web yang metode *System Development Lifecycle* (SDLC) nya menggunakan model *Rapid Application Development* (RAD). Hasil dari penelitian tersebut adalah *website* yang dibangun berhasil menunjukkan rasio, dapat mengolah data, dapat menampilkan data tabel pencarian, dan dapat memberikan data dan informasi yang valid dan akurasi yang baik. Perbedaan antara penelitian ini dengan penelitian yang dilakukan oleh Pramana & Mungguna adalah penelitian ini menggunakan *review smartphone* sebagai objek penelitian dan untuk model SDLC yang digunakan pada penelitian ini adalah model *Waterfall*. Persamaan penelitian ini dengan penelitian tersebut adalah mengimplementasikan *web scraping* untuk mengambil data yang diperlukan dari situs yang sudah ditentukan (Pramana & Mungguna, 2015).

Penelitian selanjutnya yang menjadi rujukan pada penelitian ini yaitu penelitian mengenai analisis sentimen pengguna jejaring sosial terhadap suatu topik bahasan yang ada dengan memanfaatkan pesan para pengguna jejaring sosial. Hasil dari analisis dengan menggunakan metode *Support Vector Machine* didapatkan tingkat akurasi sebesar 81% atau sebanyak 243 kicauan yang berkelompok secara tepat dari jumlah keseluruhan kicauan sebanyak 300 kicauan (Fachrurrozi & Yusliani, 2015).

### 2.2 Review

Review adalah opini pada dan/atau pengalaman konsumen terhadap sebuah produk, layanan atau bisnis. Review dapat ditemukan *online* pada situs *review* dan berbagai situs toko *online* (endorsement, 2015). Umumnya, aspek sebuah produk merupakan karakteristik dari produk tersebut, seperti contohnya, Samsung Galaxy S6 memiliki aspek seperti baterai, kualitas gambar, harga dan sebagainya (Sangeetha, et al., 2017). Opini yang merupakan isi dari review inilah yang membuat setiap review dapat berbeda dengan review yang lainnya.

## 2.3 Sistem Informasi

Sistem informasi adalah suatu sistem di dalam organisasi yang digunakan untuk mendukung kebutuhan transaksi, bersifat manajerial, dan mendukung kegiatan strategi dari suatu organisasi dan menyediakan laporan-laporan yang dibutuhkan untuk pihak luar tertentu (Hutahean, 2014).

Menurut Huatahean dalam bukunya yang berjudul “Konsep Sistem Informasi”, sistem informasi terdiri dari komponen-komponen yang disebut dengan blok bangunan yaitu :

1. Blok Masukan (*Input block*), blok ini mewakili data yang masuk kedalam sistem informasi.
2. Blok Model (*Model block*), blok ini terdiri dari kombinasi prosedur, logika dan metode matematik yang digunakan untuk memanipulasi data input dan data yang ada pada basis data sehingga dapat menghasilkan keluaran yang sudah diinginkan.
3. Blok Keluaran (*Output blok*), merupakan blok dimana mewakili produk dari sistem informasi. Berisi informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pengguna sistem.
4. Blok Teknologi (*Technologi block*), dalam sistem informasi terdapat beberapa teknologi yang digunakan yaitu terdiri dari teknisi (*human ware* atau *brain ware*), perangkat keras (*hardware*), dan perangkat lunak (*software*).
5. Blok Basis Data (*Database block*), blok ini merupakan kumpulan dari data yang saling berhubungan antara satu dengan yang lainnya.
6. Blok Kendali (*Control block*), banyak factor yang dapat merusak sistem informasi sehingga diperlukan beberapa pengendalian yang dirancang dan diterapkan untuk mencegah maupun memperbaiki kerusakan sistem informasi.

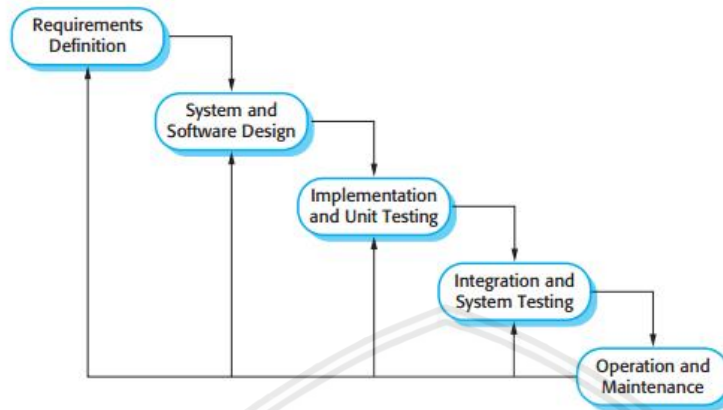
## 2.4 Software Development Life Cycle (SDLC)

*Software Development Life Cycle* (SDLC) adalah sekumpulan tahapan dalam metodologi yang diikuti pada proses mengembangkan dan memperbaiki sebuah sistem informasi (Everett & Jr, 2007). Setiap tahapan merupakan sebuah bagian dari SDLC yang terdiri dari beberapa tipe aktivitas. SDLC memiliki beberapa model pengembangan, diantaranya adalah model *Waterfall* yang akan peneliti pakai untuk pengembangan website pada penelitian ini. Pada sub-bab berikut ini akan dijelaskan mengenai metode *Waterfall* dan bagaimana tahapan-tahapan dalam metode *Waterfall*.

### 2.4.1 Model *Waterfall*

Model ini adalah model proses *software development* yang pertama kali di publikasikan, model ini didapatkan dari proses pengembangan sistem yang umum (Royce, 1970). Dikarenakan dari proses pengerjaan tahap satu ke tahap yang lain mengalir kebawah, model ini dikenal sebagai *Waterfall model*. Model ini

merupakan contoh dari prinsip proses yang bersifat *plan-driven*, dimana keseluruhan aktivitas dari setiap tahapan harus direncanakan terlebih dahulu sebelum di kerjakan (Sommerville, 2011). Tahapan-tahapan atau langkah-langkah dalam model *Waterfall* ditunjukkan pada Gambar 2.1.



**Gambar 2.1 Langkah-Langkah Model *Waterfall***

Sumber : Sommerville (2011)

Pada Gambar 2.1 menjelaskan langkah-langkah atau tahapan-tahapan pada model *waterfall*, dimana pada setiap tahapan harus dilakukan secara berurutan kebawah. Dalam menerapkan model *waterfall*, penelitian ini tidak sampai pada tahap *operation and maintainance*.

Tahap *Requirement analysis and definition* adalah tahap menganalisis dan mengidentifikasi kebutuhan dari perangkat lunak yang akan dibangun. Pada tahap ini seluruh kebutuhan harus diidentifikasi dengan baik seperti kebutuhan fungsional dan non fungsional sistem. Tahap selanjutnya adalah *System and software design*. Tahap ini merupakan tahap pembuatan desain sistem, dan kebutuhan yang sudah diidentifikasi pada tahap sebelumnya dirubah ke dalam sebuah struktur pemodelan. Pada penelitian ini, pemodelan dilakukan dengan pendekatan *Object-Oriented Analysis and Design (OOAD)* yang memungkinkan penggambaran sistem dilakukan melalui sudut pandang obyek yang terlibat kedalam sistem. Karena pendekatan yang digunakan dalam pemodelan sistem menggunakan pendekatan OOAD, maka pemodelan sistem dibuat dengan mengikuti standarisasi UML (*Unified Modeling Language*).

Setelah tahap *System and software design*, tahap selanjutnya adalah *Implementation and unit testing*. Tahap ini merupakan tahap menerjemahkan model kedalam barisan kode program. Pada penelitian ini, implementasi akan menggunakan *framework CodeIgniter*. *CodeIgniter* dipilih karena memiliki pola perancangan MVC, dimana MVC merupakan konsep dengan pendekatan *Object-Oriented Analysis and Design (OOAD)*. Pada tahap ini juga dilakukan testing terhadap unit atau bagian sistem yang sudah selesai dibangun.

Tahap terakhir dari model *Waterfall* yang dilakukan pada penelitian ini adalah tahap *Integration and system testing*. Unit atau bagian sistem yang sudah selesai

di bangun dan sudah melalui testing unit, pada tahap ini digabungkan menjadi satu kesatuan sistem yang utuh. Setelah itu akan dilakukan testing untuk keseluruhan sistem. Testing dapat dilakukan secara *Black Box* maupun *White Box*.

## 2.5 Website

*Website* adalah beberapa halaman web dimana informasi dalam bentuk teks, gambar diam, suara dan lain-lain dipresentasikan dalam bentuk *hypertext* dan dapat diakses oleh perangkat lunak yang disebut dengan browser. Informasi pada sebuah *website* pada umumnya ditulis dalam format HTML. *Website* merupakan fasilitas internet yang menghubungkan dokumen dalam lingkup lokal maupun jarak jauh. Dokumen pada *website* disebut dengan *web page* dan *link* dalam *website* memungkinkan pengguna bisa berpindah dari satu page ke page lain (*hypertext*), baik diantara *page* yang disimpan dalam *server* yang sama maupun *server* diseluruh dunia. *Pages* diakses dan dibaca melalui *browser* seperti *Netscape Navigator*, *Internet Explorer*, *Mozilla Firefox* dan aplikasi *browser* lainnya (Lukmanul & Musalini, 2004).

Berdasarkan sifatnya *website* dibagi menjadi dua yaitu statis dan dinamis. Bersifat statis apabila isi informasi *website* tetap, jarang berubah, dan isi informasinya searah hanya dari pemilik *website*. Bersifat *dinamis* apabila isi informasi *website* selalu berubah-ubah, dan isi informasinya interaktif dua arah berasal dari pemilik serta pengguna *website*. Contoh *website* statis adalah berisi profil perusahaan, sedangkan *website* dinamis adalah seperti *Friendster*, *Multiply*, dll. Dalam sisi pengembangannya, *website* statis hanya bisa diupdate oleh pemiliknya saja, sedangkan *website* dinamis bisa diupdate oleh pengguna maupun pemilik.

## 2.6 PHP

*Hyper Preprocessor* (PHP) yaitu bahasa pemrograman *web server-side* yang bersifat *open source*. PHP merupakan *script* yang terintegrasi dengan HTML dan berada pada *server* (*server side HTML embedded scripting*) (Anhar, 2010). Dapat didefinisikan PHP adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*.

Ketika menggunakan PHP sebagai *server-side embedded script language* maka server akan melakukan hal-hal sebagai berikut :

1. Membaca permintaan dari client/browser
2. Mencari halaman/page di server
3. Melakukan instruksi yang diberikan oleh PHP untuk melakukan modifikasi pada halaman/page.
4. Mengirim kembali halaman tersebut kepada client melalui internet atau intranet.

Kelebihan PHP :



1. Bahasa pemrograman php sudah menjadi populer dan diminati oleh banyak pengguna dikarenakan kesederhanaan dan kemampuannya dalam menghasilkan berbagai aplikasi berbasis web.
2. PHP juga merupakan salah satu bahasa *server side* yang memang di desain khusus untuk aplikasi berbasis web.
3. PHP merupakan *open source product* dan kini telah mencapai versi ke tujuh.

## 2.7 MySQL

MySQL adalah salah satu Database Management System (DBMS) seperti Oracle, Ms SQL, PostgreSQL, dan lainnya. MySQL berfungsi untuk mengolah database menggunakan bahasa SQL. MySQL bersifat open source sehingga bisa digunakan secara gratis (Anhar, 2010).

Bahasa SQL (*Structured Query Language*) sendiri adalah sebuah konsep pengoperasian basis data yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. SQL digunakan dalam PHP untuk berkomunikasi dengan basis data dan melakukan pengolahan data. Terdapat tiga perintah dasar yang harus diketahui dalam menggunakan SQL yaitu *CREATE*, *INSERT*, dan *DELETE*.

## 2.8 CodeIgniter

*CodeIgniter* adalah *framework* pengembangan aplikasi (*Application Development Framework*) dengan menggunakan PHP, suatu kerangka untuk bekerja atau membuat program dengan menggunakan PHP yang lebih sistematis. Pemrograman tidak perlu membuat program dari awal (*from scratch*), karena CI menyediakan sekumpulan *library* yang banyak yang diperlukan untuk menyelesaikan pekerjaan yang umum, dengan menggunakan antar muka dan struktur logika yang sederhana untuk mengakses librarinya. Pemrograman dapat memfokuskan diri pada kode yang harus dibuat untuk menyelesaikan suatu pekerjaan (Sidik, 2012).

Konsep alur kerja yang diterapkan pada CI adalah konsep Model, View, Controller (MVC). Dengan konsep MVC segala macam logika dan layout telah dipisahkan, sehingga si programmer dan designer dapat mengerjakan masing – masing tugasnya secara fokus. Konsep model MVC juga dapat menuntun para programmer untuk membangun web dengan cara terstruktur.

1. Model, digunakan sebagai presentasi database. Dalam CodeIgniter, segala macam perintah query SQL diletakkan dalam file model, seperti insert, delete dan select.
2. View, suatu halaman khusus yang digunakan untuk menampilkan informasi pada *client*. Seluruh perintah yang dikelola oleh *controller* dan *model* dikirim kepada *view* sesuai dengan perintah yang *direquest*.
3. Controller, digunakan sebagai jembatan antara *view* dan model.



## 2.9 Business Process Modelling

*Business Process Modelling* (BPM) atau pemodelan proses bisnis merupakan rancangan dan eksekusi suatu proses bisnis (Havey, 2005). Proses bisnis sendiri adalah kumpulan aktivitas yang mendukung proses-proses dalam perusahaan ataupun organisasi. Tujuan dari BPM adalah untuk memberikan gambaran mengenai tahapan-tahapan yang perlu dilakukan oleh sebuah organisasi untuk mencapai suatu tujuan. Adanya BPM bermanfaat dalam memudahkan pemahaman alur proses bisnis secara terintegrasi (Pendidikan, 2015).

### 2.9.1 Business Process Modelling Notation (BPMN)

*Business Process Modelling Notation* (BPMN) merupakan suatu standar yang dikeluarkan oleh sebuah organisasi *non-profit* yang disebut *Open Management Group* (OMG). BPMN digunakan sebagai *tools* untuk menggambarkan suatu proses bisnis dan mendeskripsikannya secara teknis bagaimana proses bisnis tersebut dieksekusi untuk keperluan otomatisasi (Programmer, 2013).






Tujuan dari BPMN adalah menyediakan bahasa umum yang dapat dipahami oleh semua pihak yang terlibat dalam proses bisnis. Menurut *Object Management Group* (2011), BPMN memiliki 5 kategori elemen dasar yaitu:

1. *Flow Object*, merupakan elemen grafis untuk mendefinisikan tingkah laku (*behavior*) sebuah proses bisnis. Terdapat 3 macam *flow object*, antara lain *events*, *activities*, *gateways* (Object Management Group, 2011).
2. *Data*, data direpresentasikan dengan 4 elemen antara lain data *objects*, data *inputs*, data *outputs*, dan data *stores* (Object Management Group, 2011).
3. *Connecting Object*, digunakan untuk menghubungkan *flow object* dengan *flow object* lainnya atau informasi lainnya. Terdapat 4 jenis *connector* antara lain *Sequence flows*, *Message flows*, *Associations*, dan *Data Associations* (Object Management Group, 2011).
4. *Swimlanes*, merupakan tempat untuk mengelompokkan elemen permodelan utama. Terdapat dua cara pengelompokkan melalui *swimlanes* yaitu *pool* dan *lanes* (Object Management Group, 2011).
5. *Artifact*, merupakan elemen yang digunakan untuk memberikan informasi tambahan mengenai proses. Terdapat dua *artifacts* yang ter-standarisasi yaitu *group* dan *text annotation* (Object Management Group, 2011).

Berdasarkan Object Management Group (2011), setiap elemen BPMN memiliki notasi yang berbeda-beda. Penjelasan mengenai elemen, notasi, dan deskripsi fungsi yang sering digunakan dalam BPMN ditunjukkan pada Tabel 2.1.

Tabel 2.1 Notasi Dasar BPMN

No.	Notasi	Elemen	Deskripsi
1.		<i>Event</i>	Sebuah event merepresentasikan segala sesuatu yang terjadi pada proses bisnis yang sedang berjalan. Terdapat tiga jenis event berdasarkan pengaruh terhadap alur yaitu <i>start</i> , <i>intermediate</i> , dan <i>end</i> .
2.		<i>Activity</i>	Sebuah <i>activity</i> merepresentasikan tugas atau <i>task</i> yang dikerjakan di dalam suatu proses bisnis.
3.		<i>Gateway</i>	<i>Gateway</i> digunakan untuk memperlihatkan pilihan yang berbeda. <i>Gateway</i> juga menjelaskan mengenai percabangan dan penggabungan <i>sequence flows</i> dalam sebuah proses.
4.		<i>Sequence Flow</i>	<i>Sequence flow</i> digunakan untuk menunjukkan alur dari aktifitas dalam sebuah proses bisnis.
5.		<i>Message Flow</i>	<i>Message flow</i> merepresentasikan alur pesan antar dua partisipan.
6.		<i>Association</i>	<i>Association</i> digunakan merepresentasikan asosiasi berbagai <i>artifacts</i> , <i>data</i> , maupun <i>flow object</i> . Kepala panah pada <i>association</i> berabarti arah dari alur (contoh : <i>data</i> ) bila diperlukan.
7.		<i>Pool</i>	Sebuah <i>pool</i> merepresentasikan sebuah entitas di dalam proses bisnis. <i>Pool</i> biasanya mewakili sebuah organisasi atau perusahaan.

8.		<i>Lane</i>	<i>Lane</i> merepresentasikan divisi-divisi yang ada didalam sebuah organisasi atau perusahaan. Terdapat dua jenis <i>lane</i> yaitu <i>lane horizontal</i> dan <i>lane vertical</i> .
9.		<i>Data Object</i>	<i>Data object</i> merepresentasikan data yang dibutuhkan atau informasi yang dihasilkan dari suatu <i>activity</i> pada proses bisnis.
10.		<i>Message</i>	<i>Message</i> merpresentasikan pesan isi dari komunikasi antara dua partisipan di dalam proses bisnis.
11.		<i>Group</i>	<i>Group</i> digunakan untuk mengelompokkan <i>activities</i> yang memiliki kategori sama/
12.		<i>Text Annotation</i>	<i>Text annotation</i> digunakan untuk menyajikan informasi teks tambahan mengenai diagram atau model BPMN.

## 2.10 Unified Modeling Language (UML)

Menurut Object Management Group (OMG), Unified Modelling Language (UML) adalah bahasa grafis yang digunakan untuk memvisualisasikan dan membuat artefak dari sistem perangkat lunak pada saat proses pengembangan. UML menawarkan standar dalam membuat cetak biru sistem, termasuk hal – hal konseptual seperti proses bisnis dan fungsional sistem serta hal-hal lain seperti bahasa pemrograman, skema basis data dan komponen dari perangkat lunak yang dapat digunakan kembali.

UML adalah sekumpulan tool yang disetujui oleh Object Management Group (OMG) sebagai standar untuk memodelkan sistem dalam object oriented software development (OOSD) (Mishra, 2011). Diagram UML dapat menggambarkan lima sudut pandang dari sistem.




1. Sudut pandang user, sudut pandang ini mendefinisikan fungsionalitas yang tersedia dari sistem untuk user. Sudut pandang user merupakan sudut pandang black-box sistem dimana struktur dalam dan implementasi sistem tidak terlihat. UML diagram untuk sudut pandang ini adalah diagram use-case (Mishra, 2011).

2. Sudut pandang struktural, mendefinisikan hubungan antar objek yang digunakan untuk memahami cara kerja dari sistem dan implementasinya. Diagram yang digunakan adalah class diagram dan *object diagram* (Mishra, 2011).
3. Sudut pandang behavioral, sudut pandang ini menggambarkan bagaimana objek saling berinteraksi dengan objek lainnya. Diagram yang digunakan adalah *State Chart Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Collaboration Diagram* (Mishra, 2011).
4. Sudut pandang implementasi, menggambarkan komponen penting dari sistem dan dependensinya. Diagram yang digunakan adalah component diagram (Mishra, 2011).
5. Sudut pandang lingkungan, sudut pandang ini memodelkan bagaimana komponen yang berbeda diimplementasikan pada perangkat keras yang berbeda. Diagram yang digunakan adalah *Deployment Diagram* (Mishra, 2011).


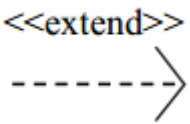
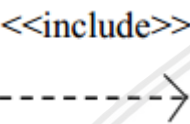
### 2.10.1 Use Case Diagram

*Use Case diagram* adalah sebuah diagram yang digunakan untuk menggambarkan fungsionalitas beserta tingkah laku dari sebuah sistem. *Use Case* juga menggambarkan interaksi antara pengguna sistem dengan sistem (Rumbaugh, et al., 2005). *Use Case* dapat digunakan untuk mengetahui fungsi yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan pada *Use Case diagram* ditunjukkan pada Tabel 2.2.

**Tabel 2.2 Simbol-Simbol Use Case Diagram**

NO.	Simbol	Nama	Deskripsi
1		Aktor	Merupakan representasi dari entitas yang berinteraksi dengan sistem. Aktor dapat merupakan orang, perangkat keras atau yang lainnya.
2		<i>Use case</i>	Merupakan fungsionalitas dari sistem sebagai unit yang saling bertukar pesan antar unit atau aktor
		<i>Association</i>	Menggambarkan komunikasi antara aktor dengan <i>use case</i> . <i>association</i> juga berarti aktor terlibat dalam sebuah <i>use case</i> .




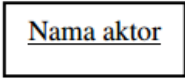
NO.	Simbol	Nama	Deskripsi
		<i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
		<i>Extend</i>	Relasi antara <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan. Dimana <i>use case</i> tersebut dapat berdiri sendiri.
		<i>Include</i>	Relasi antara <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjelaskan fungsinya atau sebagai syarat dijalankannya <i>use case</i> tersebut.



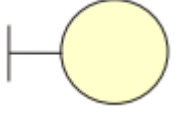





Sumber : (Sukamto & Shalahudin, 2013)

### 2.10.2 Sequence Diagram

Sebuah *Sequence diagram* menggambarkan interaksi antar objek dengan objek lainnya yang terdapat pada suatu sistem serta pesan yang dilewatkan diantar objek. Diagram ini menampilkan interaksi antar objek dalam bentuk grafik dua dimensi yaitu dimensi vertikal dan horizontal. Dimensi vertical merupakan dimensi yang menunjukkan urutan interaksi dalam suatu waktu, sedangkan dimensi horizontal menunjukkan peran yang merepresentasikan objek individual dalam kolaborasi (Rumbaugh, et al., 2005). *Sequence diagram* memiliki banyak notasi atau symbol. Simbol yang digunakan dalam *Sequence diagram* ditunjukkan pada Tabel 2.3.

**Tabel 2.3 Simbol-Simbol Dalam Sequence Diagram**

No.	Simbol	Nama	Deskripsi
1.	 atau 	<i>Actor</i>	Merepresentasikan entitas, dapat berupa orang, proses atau sistem lain yang berinteraksi dengan sistem yang akan dibangun.

No.	Simbol	Nama	Deskripsi
2.		<i>Lifeline</i>	Merpresentasikan kehidupan dari suatu objek.
3.		<i>Entity class</i>	Kelas Entitas bagian dari sistem yang menjadi landasan dalam membuat basis data.
4.		<i>Boundary class</i>	Kelas yang menjadi <i>interface</i> atau antar muka untuk interaksi antara sistem dengan satu atau lebih aktor.
5.		<i>Control class</i>	Objek kelas yang berisi logika sistem.
6.		<i>Message</i>	Menjelaskan mengenai pesan yang dikirim antar kelas.
7.		<i>Recursive</i>	Merepresentasikan pengiriman pesan yang dikirim pada dirinya sendiri.
8.		<i>Activatiom</i>	Mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
9.		<i>Return</i>	Menyatakan suatu kembalian dari operasi suatu objek, arah panah mengarah kepada objek yang menerima kembalian.

Sumber : (Sukamto & Shalahudin, 2013)

### 2.10.3 Class Diagram

“*Class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem”. *Class diagram* juga memperlihatkan peran dari entitas dalam menentukan perilaku sistem. Suatu kelas memiliki atribut dan *method*. Atribut merupakan variabel-variabel yang dimiliki oleh kelas, sedangkan *method* adalah fungsi-fungsi atau operasi yang dimiliki oleh kelas (Sukamto & Shalahudin, 2013).







*Class diagram* meliputi *Class*, *Relasi*, *Association*, *Generalization* dan *Agregation*, *Attribut*, *Operation*, *Visibility*, tingkat akses objek eksternal kepada





suatu operasi atau atribut. Simbol-simbol yang digunakan dalam *Class diagram* ditunjukkan pada Tabel 2.4.

**Tabel 2.4 Simbol-Simbol Dalam *Class Diagram***

No	Simbol	Nama	Deskripsi
1.		<i>Class</i>	Kelas pada struktur sistem
		<i>Interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
		<i>Association</i>	Relasi yang menunjukkan hubungan antar kelas.
		<i>Generalization</i>	Relasi yang merepresentasikan pewarisan dari kelas utama ( <i>parent class</i> ) ke kelas anak ( <i>child class</i> )
		<i>Dependency</i>	Relasi yang menunjukan kebergantungan antar <i>class</i>
		<i>Aggregation</i>	Relasi yang merepresentasikan agregrasi yaitu sebuah kelas terdiri dari beberapa komponen kecil.

Sumber : Sukamto & Shalahudin (2013)

Pemodelan *class diagram* dibagi menjadi tiga bentuk berbeda berdasarkan tahapan pengembangan perangkat lunak. Tiga tahapan tersebut yaitu definisi kebutuhan, analisis, dan perancangan (Scott, 2004). Tiga bentuk *Class diagram* tersebut antara lain :

1. *Domain-Level Class Diagram*

Pada *domain-level class diagram*, *class diagram* hanya menampilkan nama *class* saja.

2. *Analysis-Level Class Diagram*

Pada *analysis-level Class diagram*, *class diagram* menampilkan nama *class* dan atribut dari setiap kelas. *Class diagram* pada level ini tidak menunjukkan operasi pada *class* karena hal tersebut masuk *fase* perancangan.

3. *Design-Level Class Diagram*

*Class diagram* pada *design-level* digambarkan secara lengkap yaitu terdiri dari nama *class*, atribut, dan operasi sesuai dengan standar *Class diagram* pada notasi *Unified Modeling Language (UML)*.



## 2.11 Web Scraping

*Web scraping* adalah proses pengambilan sebuah dokumen semi-terstruktur dari internet, umumnya berupa halaman web dalam bahasa markup seperti HTML atau XHTML, dan menganalisis dokumen tersebut untuk mengekstrak data spesifik yang ada pada halaman untuk digunakan pada kepentingan lain (Turland, 2010). *Web scraping* dikenal juga sebagai *screen scraping*. Secara teknikal, *web scraping* tidak termasuk di dalam *data mining* karena pada penerapannya untuk melihat pola semantik atau tren dalam sekumpulan data yang telah didapatkan. Aplikasi *web scraping* hanya berfokus pada mendapatkan data melalui pengambilan dan ekstraksi dan dapat melibatkan data dengan variasi ukuran yang signifikan.

Terdapat beberapa library atau ekstensi yang dapat digunakan untuk melakukan scraping dengan menggunakan bahasa PHP, diantaranya adalah sebagai berikut:

1. cURL

cURL merupakan salah satu ekstensi yang tersedia pada PHP. Tersedia sejak PHP 4.0.2, membungkus sebuah library yang disebut libcurl, yang mengimplementasikan logika client untuk berbagai protokol internet termasuk HTTP dan HTTPS.

2. Document Object Model (DOM)

Document Object Model (DOM) mendapatkan namanya dari API bahasa standar *independent* untuk menavigasi dan memanipulasi dokumen XML dan HTML yang terbentuk dengan baik. Standar ini di kelola dan direkomendasikan oleh *World Wide Web Consortium (W3C)*, sebuah organisasi yang ditujukan untuk standar yang terkait dengan internet. Ekstensi DOM hanya tersedia pada PHP 5.

## 2.12 Text Mining

*Text mining* adalah proses ekstraksi informasi yang bermanfaat dari sumber data melalui identifikasi dan eksplorasi pola menarik. *Text mining* dan *data mining* memiliki proses yang sama yaitu sistem yang bergantung praproses rutin, algoritme penemuan pola, dan *interface*. Berbeda dengan data mining, sumber data dari text mining adalah sekumpulan dokumen dimana teks dan polanya tidak terstruktur, sehingga praprosesnya berfokus pada identifikasi dan ekstraksi fitur pada dokumen tersebut, tidak seperti pada *data mining* yang sumber datanya berupa data terstruktur sehingga praprosesnya berfokus pada normalisasi data pada *database* (Feldman & Sanger, 2007).

Tahapan proses pokok dalam *text mining* yaitu pemrosesan awal teks (*text preprocessing*), transformasi teks (*text transformation*), pemilihan fitur (*feature selection*), dan penemuan pola (*pattern discovery*) (Prasetyo, 2012).

## 2.13 Preprocessing Text

*Preprocessing text* merupakan awal dari *text mining*. Tahap ini mencakup semua proses untuk mempersiapkan data yang akan digunakan pada sistem *text*

*mining* dalam operasi *knowledge discovery* (Feldman & Sanger, 2007). Tahap *preprocessing text* melakukan *analisis semantic* (kebenaran arti) dan sintatis (kebenaran susunan) terhadap teks. Tujuan dari pemrosesan awal adalah untuk mempersiapkan teks menjadi data yang akan diolah lebih lanjut. Secara umum tahap-tahap *preprocessing text* dalam *text mining* pada dokumen adalah Tokenisasi, *Data cleaning*, *Case folding*, Filterisasi, Dan *Stemming*.

### 2.13.1 Tokenisasi

Tokenisasi adalah prosedur memisahkan teks menjadi kata-kata, frasa atau bagian berarti lain yang disebut token. Dengan kata lain, tokenisasi adalah bentuk segmentasi teks. Biasanya, segmentasi dilakukan dengan hanya mempertimbangkan karakter alfabet atau alfanumerik yang dibatasi oleh karakter non-alfanumerik seperti spasi (Aysal & Gunal, 2014). Contoh tahap proses tokenisasi dapat ditunjukkan pada Tabel 2.5.

**Tabel 2.5 Ilustrasi Proses Tokenisasi**

Data Input	Hasil Proses Tokenisasi
Samsung Note 8 keren, enak banget buat dipakai kerja. Tugas kantor jadi bisa dikerjakan dimana saja dan mudah #SAMSUNG	['Samsung', 'Note', '8', 'keren,', 'enak', 'banget', 'buat', 'dipakai', 'kerja.', 'Tugas', 'kantor', 'jadi', 'bisa', 'dikerjakan', 'dimana', 'saja', 'dan', 'mudah', '#SAMSUNG']

### 2.13.2 Data Cleaning

*Data cleaning* merupakan tahap dimana unsur-unsur yang tidak memiliki pengaruh pada *processing text* seperti *link url* (http) dan *hashtag* (#) di hapus dari teks atau dokumen. Ilustrasi dari proses *Data cleaning* ditunjukkan pada Tabel 2.6.

**Tabel 2.6 Ilustrasi Proses Data Cleaning**

Data Input	Hasil Proses Data Cleaning
['Samsung', 'Note', '8', 'keren,', 'enak', 'banget', 'buat', 'dipakai', 'kerja.', 'Tugas', 'kantor', 'jadi', 'bisa', 'dikerjakan', 'dimana', 'saja', 'dan', 'mudah', '#SAMSUNG']	['Samsung', 'Note', '8', 'keren,', 'enak', 'banget', 'buat', 'dipakai', 'kerja.', 'Tugas', 'kantor', 'jadi', 'bisa', 'dikerjakan', 'dimana', 'saja', 'dan', 'mudah']

### 2.13.3 Case Folding

*Case folding* merupakan tahap dimana keseluruhan huruf pada teks atau dokumen di ubah ke dalam huruf kecil (*lowercase*) dan karakter selain huruf dihilangkan dan dianggap sebagai *delimiter*. Ilustrasi dari tahap *Case folding* ditunjukkan pada Tabel 2.7.



**Tabel 2.7 Ilustrasi Proses Case Folding**

Data Input	Hasil Proses Case Folding
['Samsung', 'Note', '8', 'keren,', 'enak', 'banget', 'buat', 'dipakai', 'kerja.', 'Tugas', 'kantor', 'jadi', 'bisa', 'dikerjakan', 'dimana', 'saja', 'dan', 'mudah']	['samsung', 'note', 'keren,', 'enak', 'banget', 'buat', 'dipakai', 'kerja', 'tugas', 'kantor', 'jadi', 'bisa', 'dikerjakan', 'dimana', 'saja', 'dan', 'mudah']

### 2.13.4 Filterisasi

Filterisasi atau biasa juga disebut dengan *stopword removal* adalah tahap menghilangkan kata yang tak memiliki makna pada teks atau dokumen sehingga yang tersisa pada teks atau dokumen adalah kata-kata penting. Proses pengambilan kata-kata penting ini dilakukan dengan cara mencocokkan kata dari teks atau dokumen ke kata-kata *stopword* yang ada pada basis data atau *stoplist*. Pada penelitian ini proses filterisasi dibantu dengan daftar *stopword* dari Rachel Tatmant yaitu *Sentiment Lexicon for 81 Languages* yang dapat didapatkan pada situs web kagler.com serta *stoplist* dari Masdevid yaitu *ID-OpinionWords* yang dapat di dapatkan pada github. Ilustrasi dari proses *filterisasi* ditunjukkan pada Tabel 2.8.

**Tabel 2.8 Ilustrasi Proses Filterisasi**

Data Input	Hasil Proses Filterisasi
['samsung', 'note', 'keren,', 'enak', 'banget', 'buat', 'dipakai', 'kerja', 'tugas', 'kantor', 'jadi', 'bisa', 'dikerjakan', 'dimana', 'saja', 'dan', 'mudah']	['samsung', 'note', 'keren,', 'enak', 'banget', 'buat', 'dipakai', 'kerja', 'tugas', 'kantor', 'bisa', 'dikerjakan', 'dimana', 'mudah']

### 2.13.5 Stemming

Tahap *stemming* adalah tahap mengubah setiap kata pada teks atau dokumen hasil filterisasi menjadi kata dasar dengan cara menghilangkan setiap kata imbuhan. Tahap ini banyak digunakan untuk teks berbahasa Inggris dan lebih sulit diterapkan pada teks berbahasa Indonesia. Hal ini dikarenakan Bahasa Indonesia tidak memiliki rumus bentuk baku yang permanen (Prasetyo, 2012).

Proses *stemming* pada teks berbahasa Indonesia berbeda dengan *stemming* pada tekse berbahasa Inggris. Pada teks berbahasa Inggris, proses yang diperlukan hanya proses menghilangkan *sufiks*. Sedangkan pada teks berbahasa Indonesia, diperlukan proses untuk menghilangkan *sufiks*, *prefiks*, dan *konfiks* (Agusta, 2009). Ilustrasi dari proses *stemming* ditunjukkan pada Tabel 2.9.



Tabel 2.9 Ilustrasi Proses *Stemming*

Data Input	Hasil Proses <i>Stemming</i>
['samsung', 'note', 'keren,', 'enak', 'banget', 'buat', 'dipakai', 'kerja', 'tugas', 'kantor', 'jadi', 'bisa', 'dikerjakan', 'dimana', 'saja', 'dan', 'mudah']	['samsung', 'note', 'keren,', 'enak', 'banget', 'buat', 'pakai', 'kerja', 'tugas', 'kantor', 'bisa', 'kerja', 'dimana', 'mudah']

Pada proses *stemming* Bahasa Indonesia terdapat banyak permasalahan yang harus dihadapi (Asian, 2007), diantaranya yaitu :

1. Pada Bahasa Indonesia imbuhan cukup kompleks, diantaranya :
  - Prefiks, imbuhan yang diberikan pada depan kata. Contohnya : ber-lima.
  - Suffiks, imbuhan yang diberikan pada kalimat kata. Contohnya : minuma-an.
  - Konfiks, imbuhan yang berada pada awal dan akhir kata. Contohnya : per-tumbuh-an.
  - Infiks, imbuhan yang berada di tengah kata. Contohnya : kemilau
  - Imbuhan dari bahasa asing. Contohnya : sosial-osaso
  - Adanya aturan perubahan prefiks, seperti (me-) menjadi (meng-, mem-, men-, meny-)
2. Terdapat satu kata yang berasal dari kata dasar berbeda dan dapat memiliki dua atau lebih makna atau biasa disebut sebagai ambiguitas rasa kata (*word sense ambiguity*). Contohnya:
  - Berikan -> Ber-ikan
  - Berikan -> Beri-kan
3. *Stemming* yang berlebih (*Overstemming*)

Untuk mencegah *stemming* yang berlebih (*overstemming*), algoritma dari proses *stemming* membutuhkan daftar kata dasar. Contoh dari *library* yang menyediakan daftar kata dasar untuk Bahasa Indonesia adalah *library* Sastrawi. Contoh dari *stemming* yang berlebih adalah pada kata 'berikan', jika berdasarkan aturan pemenggalan kata tersebut dapat menjadi kata dasar 'i' karena aturan pemenggalan 'ber-i-kan'.

#### 4. *Understemming*

*Understemming* biasanya terjadi karena terdapat kata yang jika dipenggal dapat menjadi beberapa kata dasar dikarenakan kata dasar tersebut terdapat pada daftar kata dasar. Contoh dari *understemming* adalah kata 'mengecek' yang seharusnya menjadi 'menge-cek' dengan kata dasar 'cek' menjadi

- 'meng-ecek'. Hal ini terjadi dikarenakan ada kata 'ecek' pada daftar kata dasar yang digunakan pada proses *stemming*.
5. Proses *stemming* bergantung pada kamus/daftar kata  
Kamus/daftar kata dasar sangat mempengaruhi hasil dari proses *stemming*. Adanya kelebihan atau kekurangan pada daftar kata dasar dapat menyebabkan terjadinya *overstemming* atau *understemming*.
  6. Jika dilakukan *stemming* secara manual, sering kali terjadi penggunaan Bahasa Indonesia yang tidak konsisten.  
Proses *stemming* yang dilakukan secara manual sering kali memiliki perbedaan dalam menentukan *stem* sebuah kata. Contohnya adalah, apakah kata 'bagi' merupakan kata dasar dari 'bagian'?
  7. Terdapat bentuk kata jamak pada Bahasa Indonesia. Contohnya adalah 'buku-buku', yang mana kata dasarnya adalah 'buku'.
  8. Pada Bahasa Indonesia terdapat banyak kata serapan dari bahasa asing.
  9. Kesalahan pada penulisan. Hal ini menyebabkan proses *stemming* tidak dapat melakukan *stem* pada sebuah kata. Contohnya adalah 'pengurangan', harusnya adalah 'pengurangan'.
  10. Adanya kata akronim, seperti contohnya 'pemilu', ketika masuk proses *stem* menjadi 'pe-milu'.
  11. Nama benda (*proper noun*) seperti nama kota atau nama orang seharusnya dalam proses *stem* tidak dilakukan *stem*, namun sering kali nama benda pada proses *stem* Bahasa Indonesia masih dilakukan *stem* pada kata. Contohnya adalah nama orang 'Abdullah' dilakukan *stem* menjadi 'Abdul'.

Untuk menyelesaikan masalah pada proses *stemming* Bahasa Indonesia, maka diperlukan algoritma yang tepat. Salah satu contoh algoritme *stemmer* adalah algoritma nazief & adriani, yang ditemukan oleh Bobby Nazief dan Mirna Adriani pada tahun 1996.

Saat ini juga sudah tersedia *library stemmer* dengan algoritma yang diterapkan lebih dari satu algoritma *stemming*. Salah satunya adalah sastrawi stemmer, *library* ini merupakan sebuah *library stemmer* sederhana yang didesain untuk dapat digunakan secara mudah. Algoritma *stemming* Bahasa Indonesia yang diterapkan pada *library* ini adalah Algoritme nazief & adriani, kemudian ditingkatkan oleh Algoritme CS (*Confix Stripping*), lalu ditingkatkan lagi oleh *Modified ECS*. Dengan menggunakan algoritme-algoritme tersebut, *library* ini dapat mengatasi beberapa persoalan *stemming* yang ada diatas.

## 2.14 Algoritme TF-IDF (*Term Frequency-Inverse Document*)

Setelah tahapam *preprocessing text*, kumpulan dari hubungan suatu kata (*term*) hasil dari *Preprocessing text* di ubah menjadi *indeks* yaitu dengan memberikan nilai atau bobot pada setiap kata. Indeks digunakan sebagai wakil



dokumen untuk mempermudah proses *text mining* selanjutnya. Terdapat beberapa metode untuk memberikan bobot pada kata, salah satunya adalah dengan menggunakan Algoritme TF-IDF.

*Term Frequency-Invers Document* (TF-IDF) adalah cara pemberian bobot hubungan suatu kata (*term*) terhadap dokumen (Mulyana, et al., 2012). Untuk dokumen tunggal tiap kalimat dianggap sebagai dokumen. Metode ini menggabungkan dua konsep untuk perhitungan bobot, yaitu *Term Frequency* (TF) merupakan frekuensi kemunculan kata (t) pada dokumen (d). *Document frequency* (DF) adalah banyaknya dokumen yang mengandung kata (t). Untuk rumus perhitungan TF ditunjukkan pada Persamaan 2-1.

$$TF = \begin{cases} 1 + \log_{10}(f_{t,d}), & f_{t,d} > 0 \\ 0, & f_{t,d} = 0 \end{cases} \quad (2-1)$$

Selanjutnya untuk menghitung TF-IDF, kita harus menghitung IDF (*Inverse Document Frequency*). IDF merupakan kebalikan dari DF, pada IDF *term* yang kemunculannya paling banyak pada dokumen merupakan *term* dengan bobot paling tinggi. Rumus perhitungan IDF ditunjukkan pada Persamaan 2-2.

$$IDF_j = \log(D/df_j) \quad (2-2)$$

Dimana (D) adalah jumlah semua dokumen koleksi sedangkan (dfj) adalah jumlah dokumen yang mengandung term (tj).

Untuk perhitungan TF-IDF, TF yang biasa digunakan untuk perhitungan adalah perhitungan TF murni, dimana nilai TF diberikan berdasarkan jumlah kemunculan suatu term pada dokumen. Dengan demikian rumus umum untuk perhitungan bobot term TF-IDF adalah mengalikan nilai TF murni dengan nilai IDF. Rumus perhitungan TF-IDF ditunjukkan pada Persamaan 2-3 dan 2-4.

$$W_{ij} = tf_{ij} \times idf_j \quad (2-3)$$

$$W_{ij} = tf_{ij} \times \log(D/df_j) \quad (2-4)$$

Dimana (Wij) adalah bobot term (tj) terhadap dokumen (di). Sedangkan (tfij) adalah jumlah kemunculan term (tj) dalam dokumen (di). (D) adalah jumlah semua dokumen yang ada dalam database dan (dfj) adalah jumlah dokumen yang mengandung term (tj).

### 2.15 Sentiment Analysis

Sentiment analysis atau opinion mining adalah proses memahami, mengekstrak dan mengelola data tekstual secara otomatis untuk mendapatkan informasi sentimen yang terkandung dalam suatu kalimat opini. *Sentiment analysis* dilakukan untuk melihat pendapat atau kecenderungan opini seseorang terhadap sebuah masalah atau objek. Dalam *sentiment analysis* sebuah pendapat

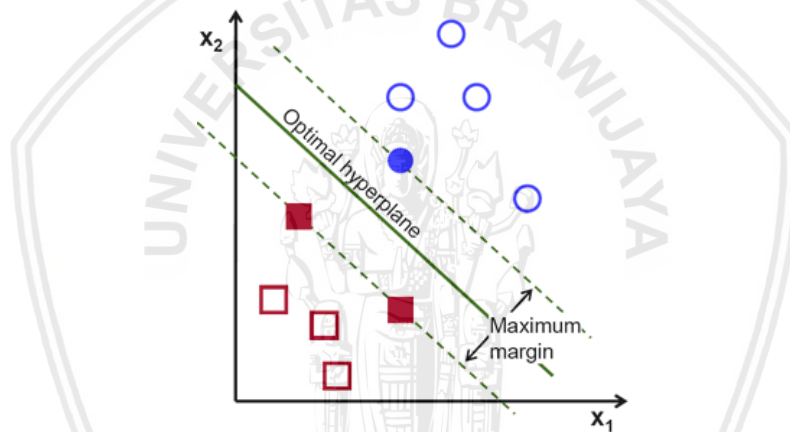


atau opini seseorang dapat diklasifikasikan menjadi opini negatif atau positif (Bo Pang, 2008).

## 2.16 Support Vector Machine

*Support Vector Machine* (SVM) adalah suatu teknik untuk melakukan prediksi, baik dalam kasus klasifikasi maupun regresi. SVM berada dalam satu kelas dengan *Artificial Neural Network* (ANN) dalam hal fungsi dan kondisi permasalahan yang bisa diselesaikan. Keduanya masuk dalam kelas *Supervised Learning* (Santoso, 2007).

SVM adalah sebuah algoritma yang diusulkan oleh Vapnik pada tahun 1995. SVM tergolong metode klasifikasi baru dan telah banyak dijadikan metode dalam sejumlah penelitian, seperti *pattern*, *recognition*, *regresi*, dan estimasi. SVM menggunakan masukan dari data *training* untuk menemukan *hyperlane* yang dapat mengklasifikasikan dua atau lebih tipe data untuk kemudian memproses atribut-atribut dari masalah klasifikasi (Nisa, 2013).



Gambar 2.2 Ilustrasi Algoritma *Support Vector Machine*

Pemodelan dari *Support Vector Machine* adalah sebagai berikut :

1. Titik data :  $x_i = \{x_1, x_2, \dots, x_n\} \in R^n$
2. Kelas data :  $y_i \in \{-1, +1\}$
3. Pasangan data dan kelas :  $\{(x_i, y_i)\}_{i=1}^N$
4. Fungsi keputusan klasifikasi  $sign(h(x))$  :

$$f(x) = w \cdot x + b \text{ atau } f(x) = \sum_{i=1}^m a_i y_i k(x, x_i) + b \quad (2-5)$$

Keterangan :

- N : jumlah data
- n : jumlah fitur atau dimensi dari dat
- w : parameter *hyperline* yang dicari (garis tegak lurus antara garis *hyperline* dan titik *Support vector*)
- x : titik data input *Support Vector Machine*
- $a_i$  : nilai bobot setiap titik data
- $K(x, x_i)$  : fungsi kernel

- b : parameter *hyperline* yang dicari (nilai bias)

Saat pertama kali diperkenalkan oleh Vapnik, SVM hanya dapat mengklasifikasikan data ke dalam dua kelas (klasifikasi biner). Namun, penelitian lebih lanjut mengembangkan SVM sehingga dapat mengklasifikasikan data yang memiliki lebih dari dua kelas.

## 2.17 Black Box Testing

Black Box testing atau biasa disebut *behavioral* testing adalah pengujian sistem yang berfokus pada pengujian kebutuhan fungsional sistem. *Black Box testing* memungkinkan software engineer untuk membuat set kondisi input yang dapat dikerjakan oleh semua fungsi yang ada pada sistem. Tujuan dari *Black Box testing* adalah untuk menemukan kesalahan seperti 1) fungsi yang salah atau hilang, 2) *interface errors*, atau 3) kesalahan performa (Pressman, 2010).

Salah satu metode black box testing adalah fungsional *testing*. *Functional testing* adalah teknik pengujian dimana hanya melakukan pengamatan nilai keluaran untuk nilai masukan tertentu, tanpa ada upaya untuk menganalisis kode program. Fungsional *testing* dilakukan berdasarkan kebutuhan fungsional sistem yang sudah didefinisikan (Agarwal, et al., 2010). Pada penelitian ini teknik pengujian *Black Box* yang digunakan adalah teknik *validation testing* dan *compatibility testing*.

### 2.17.1 Validation Testing

Pengujian validasi atau *validation testing* adalah pengujian yang bertujuan untuk melakukan validasi terhadap aksi-aksi yang dilakukan oleh pengguna ke sistem dan keluaran yang dilakukan oleh sistem akibat dari aksi pengguna (Pressman, 2010). Validasi disini memiliki arti membandingkan apakah keluaran dari sistem akibat dari aksi yang dilakukan sesuai dengan harapan awal pengguna. Pada pengujian ini terdapat dua hasil kemungkinan yang muncul yaitu pertama fungsi yang di uji sesuai dengan kebutuhan dan sesuai dengan harapan awal pengguna, hasil yang kedua adalah fungsi yang diujikan tidak sesuai dengan kebutuhan dan harapan awal pengguna.

### 2.17.2 Compatibility Testing

Pengujian *compatibility* atau *compatibility testing* adalah pengujian yang dilakukan untuk mengetahui apakah sistem yang dibangun dapat berjalan dengan baik pada lingkungan yang berbeda, seperti komputer yang berbeda, *browser*, sistem informasi dan koneksi jaringan (Pressman, 2010).

Terdapat berbagai tipe dalam pengujian *compatibility* salah satu diantaranya adalah pengujian *combility browser*. Pengujian *compatibility browser* bertujuan untuk mengetahui *browser* apa saja yang dapat menjalankan sistem dengan baik. Pada penelitian ini akan dilakukan pengujian *compatibility browser* dengan menggunakan bantuan *tools* SortSite.

## 2.18 Usability Testing

*Usability testing* atau pengujian *usability* adalah proses pengujian yang melibatkan perwakilan dari pengguna perangkat lunak yang akan diuji untuk mengevaluasi sejauh mana perangkat lunak memenuhi kegunaan tertentu (Rubin & Chisnell, 2008). Pengujian *usability* dilakukan dengan cara memberikan *task-task* tertentu kepada perwakilan pengguna perangkat lunak. Terdapat beberapa metrik yang dapat digunakan untuk membantu dalam mengukur hasil dari pengujian *usability*. Berdasarkan ISO/IEC 9216-4, metrik *usability* harus mencakup tiga hal yaitu *effectiveness*, *Efficiency*, dan *satisfaction* (Mifsud, 2015).

### 2.18.1 Metrik Effectiveness Usability Testing

*Effectiveness* dalam *usability testing* mencakup tingkat akurasi dan kelengkapan dimana pengguna mencapai tujuan tertentu (Mifsud, 2015). Metrik *effectiveness* dapat dihitung dengan menghitung tingkat pengguna menyelesaikan *task-task* tertentu. Dalam metrik *effectiveness* tingkat kelengkapan pengguna dalam menyelesaikan suatu *task* dihitung dalam angka biner, dengan angka '1' jika pengguna berhasil menyelesaikan sebuah *task* dan angka '0' jika pengguna gagal dalam menyelesaikan sebuah *task*.

*Effectiveness* dapat direpresentasikan ke dalam persen dengan menggunakan rumus 2-6 dibawah ini.

$$\text{Effectiveness} = \frac{\text{Jumlah task yang berhasil diselesaikan}}{\text{Total jumlah task yang dilakukan}} \times 100\% \quad (2-6)$$

### 2.18.2 Metrik Efficiency Usability Testing

Metrik *efficiency* mencakup sumber daya waktu yang dibutuhkan oleh pengguna dalam menyelesaikan suatu *task* (Mifsud, 2015). Waktu tersebut dapat dalam satuan detik maupun menit. *Efficiency* dapat dihitung dengan menggunakan *Time-Based Efficiency* (TBE). *Time-Based Efficiency* dapat dihitung dengan menggunakan rumus 2-7 berikut ini.

$$TBE = \frac{\sum_{j=1}^R \sum_{i=1}^N \frac{n_{ij}}{t_{ij}}}{N \times R} \quad (2-7)$$

Keterangan:

N = Jumlah total dari *task* yang diberikan.

R = Jumlah pengguna yang melakukan pengujian.

$n_{ij}$  = Hasil dari *task* ke-i yang dilakukan oleh pengguna j; jika *task* berhasil diselesaikan maka nilainya adalah 1, sedangkan jika *task* tidak berhasil diselesaikan maka nilainya adalah 0.

$t_{ij}$  = jumlah waktu yang dibutuhkan oleh pengguna j dalam menyelesaikan *task* ke-i. Jika *task* tidak berhasil diselesaikan maka

nilai dari  $t_{ij}$  adalah waktu yang diperlukan oleh pengguna sampai pengguna keluar dari *task* tersebut.

### 2.18.3 Metrik *Satisfaction Usability Testing*

Metrik *satisfaction* mencakup pada tingkat kepuasan dan penerimaan pengguna terhadap suatu perangkat atau sistem. *Satisfaction* pengguna dapat diukur dengan menggunakan *standardized satisfaction questionnaires* atau dapat disebut sebagai kuisisioner kepuasan yang terstandarisasi. Kuisisioner ini dapat diberikan ketika pengguna telah menyelesaikan suatu *task* atau ketika sesi pengujian *usability* telah selesai (Mifsud, 2015).

Untuk menghitung *satisfaction* pada penelitian ini menggunakan kuisisioner *task level satisfaction* yang diberikan setelah setiap *task* selesai dilakukan. Terdapat beberapa bentuk pertanyaan yang dapat digunakan dalam kuisisioner *task level satisfaction*, salah satunya adalah *Single Ease Question (SEQ)*. SEQ merupakan salah satu bentuk pertanyaan dalam kuisisioner yang berupa pertanyaan mengenai seberapa mudah atau sulit pengguna dalam menyelesaikan *task* yang telah diberikan, jawaban dari pertanyaan diberikan dalam bentuk poin-poin skala *likert*. *Single Ease Question (SEQ)* digunakan karena pertanyaan yang diberikan pendek dan mudah untuk ditanggapi serta mudah dalam menghitung nilai hasilnya (Sauro, 2012).

Selain dengan menggunakan kuisisioner *task level satisfaction*, penelitian ini juga menggunakan kuisisioner *test level satisfaction*. Metode pertanyaan yang digunakan pada kuisisioner *test level satisfaction* adalah metode pertanyaan *System Usability Scale (SUS)*. Berdasarkan Nathan Thomas (2015) pada SUS ini sudah terdapat 10 pertanyaan template yang setiap jawaban dari pertanyaan memiliki skor 1 sampai 5. Dimana 1 berarti tidak sangat setuju dan 5 berarti sangat tidak setuju. Sepuluh pertanyaan template tersebut adalah sebagai berikut

1. Saya pikir saya akan sering untuk menggunakan sistem ini
2. Saya menemukan sistem yang tidak rumit untuk digunakan
3. Saya pikir sistemnya mudah untuk digunakan
4. Saya pikir saya akan membutuhkan dukungan dari orang teknisi untuk dapat menggunakan sistem ini
5. Saya menemukan berbagai fungsi dalam sistem ini terintegrasi dengan baik
6. Saya pikir ada terlalu banyak inkonsistensi dalam sistem ini
7. Saya akan membayangkan bahwa kebanyakan orang akan belajar menggunakan sistem ini dengan sangat cepat.
8. Saya menemukan sistem yang sangat rumit untuk digunakan
9. Saya merasa sangat yakin menggunakan sistem ini.
10. Saya perlu belajar banyak hal sebelum saya dapat menggunakan sistem ini.

Perhitungan bobot untuk setiap pertanyaan pada *system usability scale* (SUS) memiliki peraturan sebagai berikut

1. Untuk setiap pertanyaan pada nomor ganjil, nilai dari jawaban dikurangi satu.
2. Untuk setiap pertanyaan pada nomor genap, nilainya adalah 5 dikurangi nilai dari jawaban.
3. Kemudian jumlahkan keseluruhan nilai baru yang di dapatkan dan hasilnya dikalikan 2.5

Perangkat lunak yang baik adalah perangkat lunak yang memiliki skor SUS lebih dari sama dengan 70 (Sauro, 2012). Untuk mengetahui nilai skor SUS keseluruhan sistem, maka digunakan rumus

$$\text{Nilai rata - rata} = \sum_{i=1}^n \frac{x_i}{N} \quad (2-8)$$

#### 2.18.4 Skala Likert

Skala *Likert* adalah skala tanggapan psikometri yang digunakan dalam kuisioner untuk mendapatkan tingkat persetujuan dari responden melalui sebuah pernyataan atau dalam serangkaian pernyataan (Bertram, 2016). Skala *likert* pada umumnya menggunakan skala poin yang terdiri dari 5 poin yang berurutan, dengan pada salah satu ujung sisi poin berarti “sangat tidak setuju” sedangkan sisi ujung poin yang lainnya berarti “sangat setuju”. Poin diantara kedua sisi ujung tersebut berarti “antara setuju maupun tidak setuju”. Setiap level diberikan angka numerik dimulai dari angka satu kemudian bertambah satu pada setiap level selanjutnya. Berikut ini pada gambar 2.3 adalah contoh dari skala *likert*.



**Gambar 2.3 Contoh Skala Likert**

Sumber : Bertram (2016)

Proses perhitungan skala *likert* dimulai dengan memberikan bobot nilai pada setiap poin jawaban. Berikut ini pada tabel 2.10 merupakan bobot dari setiap pilihan jawaban yang akan digunakan untuk proses perhitungan skala *likert* pada penelitian ini. Sedangkan pada tabel 2.11 merupakan prosentase bobot dari setiap pilihan jawaban.

**Tabel 2.10 Bobot Nilai Jawaban**

Pilihan Jawaban	Bobot Nilai
A = Sangat Setuju	5
B = Setuju	4



C = Netral	3
D = Kurang	2
E = Sangat Kurang	1

Sumber: (Choizes, 2017)

**Tabel 2.11 Persentase Nilai Jawaban**

Interval Persentase	Interpretasi Skor
80 hingga 100%	Sangat Setuju
60 hingga 79,99%	Setuju
40 hingga 59,99%	Netral
20 hingga 39,99%	Tidak Setuju
0 hingga 19,99%	Sangat Kurang Setuju

Sumber: (Choizes, 2017)

Selanjutnya setelah setiap pilihan jawaban telah ditentukan bobot nilai dan prosentasenya, kemudian dilakukan perkalian pada setiap jawaban dengan mengkalikan jumlah jawaban dengan bobot nilainya. Perhitungan ini dijelaskan pada rumus 2-8.

1. Jawaban Sangat Setuju (A) =  $n \times 5$
2. Jawaban Setuju (B) =  $n \times 4$
3. Jawaban Netral (C) =  $n \times 3$
4. Jawaban Tidak Setuju (D) =  $n \times 2$
5. Jawaban Sangat Tidak Setuju (E) =  $n \times 1$

$$\text{Total Nilai} = (n \times 5) + (n \times 4) + (n \times 3) + (n \times 2) + (n \times 1) \quad (2-9)$$

Dimana n merupakan jumlah dari responden yang menjawab. Selanjutnya dilakukan perhitungan kembali untuk mencari hasil interpretasi yaitu mencari skor tertinggi (X) dan skor terendah (Y) dari skala *likert*. Rumus yang digunakan untuk mencari Y dan X dijelaskan pada rumus 2-9 dan 2-10 dengan N1 merupakan nilai tertinggi dari skala *likert*, N2 merupakan nilai terendah dari skala *likert*, n merupakan jumlah responden dan U adalah jumlah dari kasus yang diujikan.

$$Y = N1 \times n \times U \quad (2-10)$$

$$X = N2 \times n \times U \quad (2-11)$$

Untuk mengetahui nilai akhir dari skala *likert* maka digunakan rumus seperti pada rumus 2-12 dibawah ini.

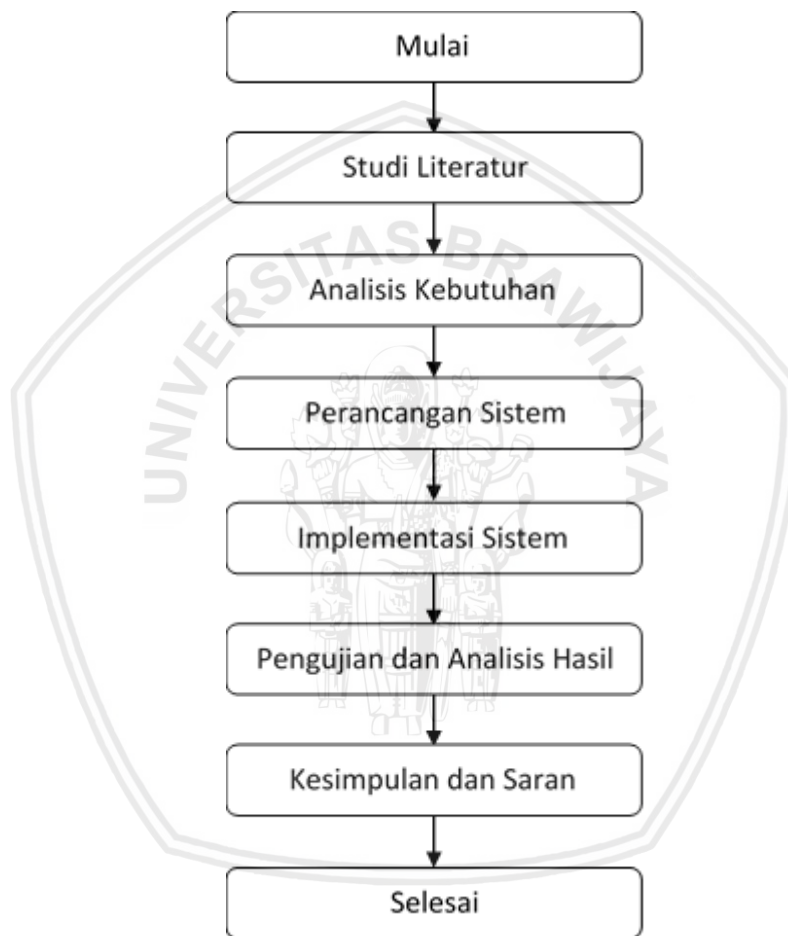
$$\text{Rumus Index \%} = (\text{Total Nilai} \div Y) \times 100 \quad (2-12)$$





## BAB 3 METODOLOGI PENELITIAN

Bab ini menjelaskan metodologi atau tahapan-tahapan yang dilakukan dalam penelitian pengembangan Sistem Informasi *Review Smartphone*. Tahapan-tahapan yang dilakukan pada penelitian ini adalah dimulai dengan melakukan studi literatur, kemudian dilanjutkan dengan tahapan analisis kebutuhan, perancangan sistem, implementasi, pengujian sistem, kemudian diakhiri dengan kesimpulan dari penelitian dan saran untuk penelitian selanjutnya. Gambar 3.1 dibawah ini menunjukkan tahapan-tahapan yang akan dilakukan pada penelitian ini.



Gambar 3.1 Diagram alur metodologi penelitian

### 3.1 Studi Literatur

Studi literatur merupakan proses pengumpulan data dan informasi yang berkaitan dengan dasar-dasar ilmu atau materi yang dibutuhkan pada penelitian ini. Studi literatur pada penelitian ini dilakukan pada buku-buku referensi, penelitian terdahulu, serta sumber-sumber dari *internet* yang berhubungan dengan topik pengembangan sistem informasi dan *text mining*.

### 3.2 Analisis Kebutuhan

Analisis kebutuhan merupakan tahapan dimana dilakukan proses pengumpulan data dan informasi untuk mendapatkan kebutuhan-kebutuhan yang diperlukan guna menyelesaikan masalah yang ada. Pada tahapan analisis kebutuhan dilakukan *observasi* atau pengamatan langsung proses bisnis konsultasi *smartphone* pada TNT Cell guna menemukan masalah yang terdapat pada proses jual beli. Selain itu dilakukan juga wawancara yang Wawancara melibatkan pemilik dari TNT Cell Bojonegoro sebagai narasumber. Dari hasil wawancara akan dibuat daftar kebutuhan dari sistem yang akan dibuat. Daftar kebutuhan dari sistem akan divalidasi terlebih dahulu oleh pemilik TNT Cell sebelum masuk kedalam tahap perancangan sistem.

Pada tahapan ini ditentukan pula situs web yang akan di *scraping* datanya. Salah satu hal yang perlu dipertimbangkan ketika menentukan situs web yang akan dipilih untuk *scraping* datanya diperlukan aksesibilitas yang tidak terbatas dari data yang akan di ambil (Cao, et al., 2011). Untuk melakukan *scraping* review, perlu diperhatikan juga bahwa situs web yang data reviewnya akan diambil, data reviewnya dapat dibaca oleh semua orang tanpa harus mendaftar atau menjadi member dari situs web tersebut (Hughes & Cohen, 2011). Oleh karena itu pada penelitian ini dipilih situs web Priceprice.com, Pricebook.co.id, serta lprice.co.id sebagai situs yang akan diambil data reviewnya. Ketiga situs tersebut dipilih dikarenakan tidak terbatasnya aksesibilitas pada keseluruhan review yang ada di situs web tersebut dan untuk membaca review pada situs tersebut, tidak diharuskan untuk mendaftar atau menjadi member dari situs web tersebut.

### 3.3 Perancangan Sistem

Pada tahap perancangan sistem akan dilakukan perancangan desain sistem berdasarkan analisis kebutuhan yang sudah divalidasi oleh pemilik TNT Cell Bojonegoro. Pada penelitian ini proses bisnis pada TNT Cell akan digambarkan ke dalam BPMN lalu perancangan sistem dilakukan dengan menggunakan *Unified Modeling Language* (UML). Hasil dari perancangan sistem adalah *Sequence Diagram*, *Class Diagram*, *Physical Data Model*, dan perancangan antar muka yang dibuat sesuai dengan kebutuhan.

### 3.4 Implementasi

Pada tahap ini hasil dari perancangan sistem di implementasikan menjadi kode program. Dalam tahap implementasi Sistem Informasi *Review Smartphone*, implementasi dilakukan dengan menggunakan bahasa pemrograman PHP, *database* MySQL, *framework* CodeIgniter, *library* DOM, serta *tools* dan *library* lain yang mendukung proses implementasi lainnya.

### 3.5 Pengujian dan Analisis Hasil

Pengujian pada sistem dilakukan guna memastikan apakah sistem yang dibangun dapat berjalan dengan baik dan telah sesuai dengan kebutuhan yang telah disepakati antara pengembang dan pemilik TNT Cell.

### 3.6 Kesimpulan dan Saran

Setelah sistem selesai dibuat dan diuji, maka selanjutnya dibuatkan kesimpulan dari penelitian ini. Kesimpulan berisi hasil dari setiap tahapan yang dilakukan pada penelitian ini. Pada penelitian ini juga terdapat saran yang diberikan untuk penelitian selanjutnya. Saran merupakan masukan dari peneliti untuk penelitian selanjutnya yang memiliki topik sama dengan penelitian ini. Pembuatan saran didasari dari hasil penelitian yang telah dilakukan.



## BAB 4 ANALISIS KEBUTUHAN SISTEM

Analisis kebutuhan merupakan kegiatan yang bertujuan untuk mendapatkan dan menganalisis daftar kebutuhan sistem informasi review *Smartphone*. Beberapa cara digunakan untuk pengumpulan data dan informasi diantaranya adalah dengan wawancara terhadap stakeholder dan melakukan observasi terhadap proses konsultasi antara konsumen dan petugas mengenai sebuah *smartphone* pada TNT Cell Bojonegoro.

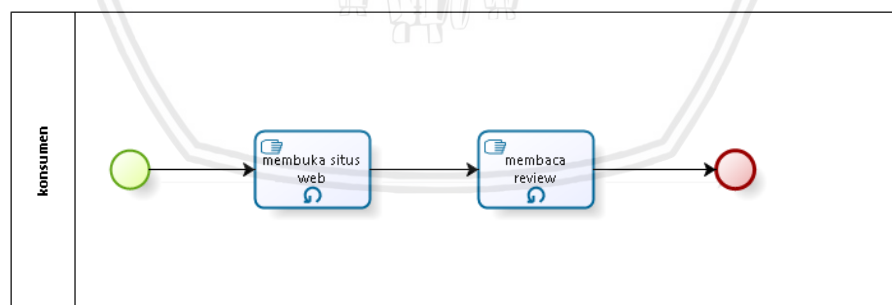
### 4.1 Profil TNT Cell Bojonegoro

TNT Cell Bojonegoro adalah salah satu *counter* atau toko penjual HP di Bojonegoro. TNT Cell beralamat di Jl.Mastrip, Ledok Wetan, Kecamatan Bojonegoro, Jawa Timur. TNT Cell merupakan salah satu *counter* yang menjual berbagai merk *smartphone*. *Counter* ini berdiri sejak tahun 2012. Salah satu proses bisnis yang ada pada TNT Cell adalah proses konsultasi mengenai *smartphone* antara calon konsumen dengan petugas.

### 4.2 Analisis Proses Bisnis

#### 4.2.1 Identifikasi Proses Bisnis Saat Ini

Proses bisnis mencari dan membaca *review smartphone* yang berjalan pada saat ini adalah konsumen membuka satu persatu dari situs web yang mengandung *review smartphone* yang diingikannya. Hal ini cukup menyulitkan konsumen karena harus membuka satu persatu dari situs web yang ada. Berikut ini merupakan proses bisnis dari mencari dan membaca *review smartphone* yang berjalan pada saat ini .



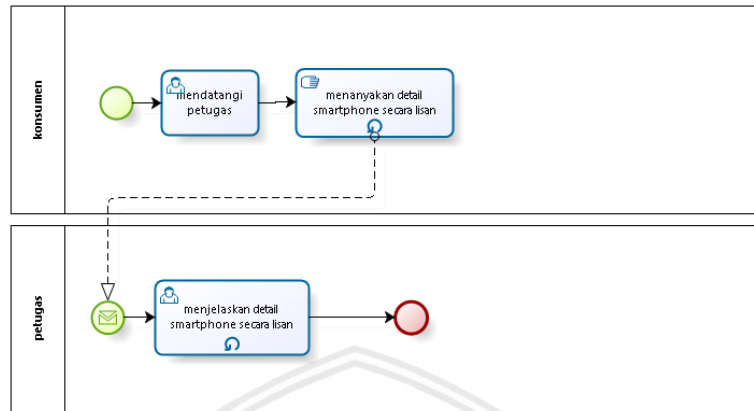
**Gambar 4.1 Proses Bisnis Mencari Dan Membaca Review Smartphone Saat Ini**

Berikut ini merupakan alur proses bisnis mencari dan membaca *review smartphone* saat ini :

1. Konsumen membuka satu persatu situs web
2. Kemudian konsumen dapat membaca *review* yang ada pada situs tersebut

Kemudian proses bisnis konsultasi mengenai *smartphone* pada TNT Cell yang saat ini berjalan melibatkan petugas dan konsumen. Pada proses ini petugas menjelaskan mengenai fitur yang terdapat pada *smartphone*. Berikut ini

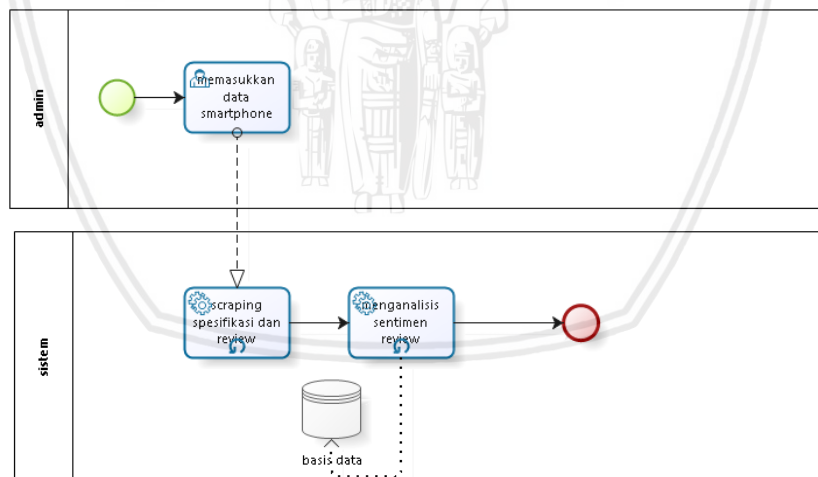
merupakan alur proses bisnis mencari dari konsultasi mengenai *smartphone* yang saat ini berjalan di TNT Cell.



**Gambar 4.2** Proses Bisnis Konsultasi *Smartphone* Saat Ini

#### 4.2.2 Proses Bisnis Yang Diusulkan

Pada proses bisnis yang diusulkan ini untuk membaca *review* dan mengetahui detail dari *smartphone* terdapat dua aktivitas, yaitu pertama admin memasukkan data *smartphone* pada sistem dan berikutnya adalah konsumen membaca *review* pada sistem. Alur proses bisnis memasukkan data *smartphone* pada sistem ditunjukkan pada Gambar 4.3.

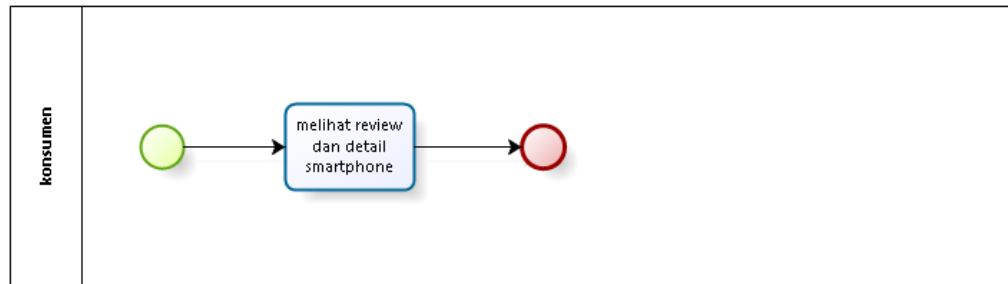


**Gambar 4.3** Proses Bisnis Memasukkan Data *Smartphone*

Proses bisnis yang diusulkan melibatkan sistem informasi *review smartphone*. Admin disini merupakan salah satu petugas dari TNT CELL. Berikut penjelasan alur proses bisnis memasukkan data *smartphone* :

1. Admin memasukkan data *smartphone* pada sistem
2. Selanjutnya sistem akan menscraping (mengambil spesifikasi dan *review*) dan melakukan analisis sentimen atau klasifikasi pada *review* kemudian menyimpan data pada basis data.

Dengan adanya sistem, konsumen dapat membaca *review* dari beberapa situs sekaligus. Berikut ini merupakan proses bisnis dari konsumen membaca *review* yang diusulkan.



**Gambar 4.4 Proses Bisnis Membaca *Review* Yang Diusulkan**

Penjelasan dari Gambar 4.4 antara lain adalah sebagai berikut ini :

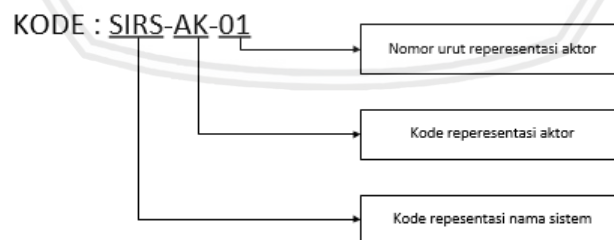
1. Konsumen membaca *review* pada sistem informasi *review smartphone*. Konsumen juga dapat mengetahui apakah *review* tersebut merupakan *review* positif atau *review* negatif.

### 4.3 Analisis Aktor

Bagian ini menjelaskan mengenai orang-orang yang terlibat dengan sistem yang akan dibangun. Orang – orang yang terlibat dengan sistem pada penelitian ini disebut sebagai aktor atau user.

#### 4.3.1 Aturan Penomoran Aktor

Pemberian nomor bertujuan untuk mempermudah pengguna atau pengembang sistem dalam mengidentifikasi aktor yang terlibat di dalam sistem yang akan dibangun. Aturan penomoran aktor pada sistem informasi *review Smartphone* ditunjukkan pada Gambar 4.5.



**Gambar 4.5 Aturan Penomoran Kode Aktor**

#### 4.3.2 Identifikasi Aktor

Proses identifikasi aktor dilakukan berdasarkan proses wawancara terhadap pemilik TNT Cell. Terdapat dua aktor yang memiliki kepentingan dalam menggunakan sistem informasi *review Smartphone* yaitu admin dan konsumen. Pada tabel dibawah ini merupakan identifikasi aktor yang terlibat di dalam sistem. Identifikasi aktor ditunjukkan pada Tabel 4.1.



Tabel 4.1 Identifikasi Aktor

No.	Kode Aktor	Nama Aktor	Deskripsi Aktor
1.	SIRS-AK-01	Admin	Admin merupakan orang yang dapat melakukan <i>update</i> data pada sistem.
2.	SIRS-AK-02	Konsumen	Konsumen merupakan calon pembeli melihat data <i>Smartphone</i> dan review dari <i>Smartphone</i> .

#### 4.4 Analisis Permasalahan

Analisis permasalahan merupakan proses identifikasi masalah yang terjadi pada saat ini sehingga dibutuhkan suatu Sistem Informasi sebagai jawaban dari masalah tersebut. Identifikasi dari permasalahan yang terjadi pada saat ini ditunjukkan pada Tabel 4.2.

Tabel 4.2 Identifikasi Masalah

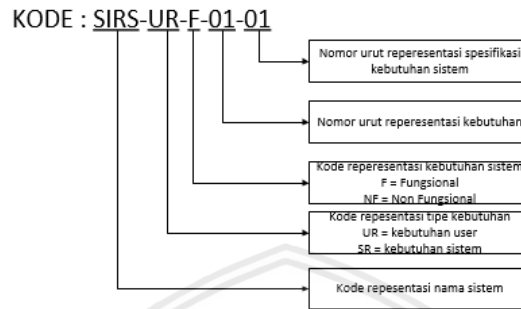
<b>Masalah</b>	Konsumen hanya ingin kesimpulan apakah <i>smartphone</i> yang dinginkannya bagus atau tidak tanpa mengetahui parameternya Untuk melayani satu konsumen, petugas membutuhkan waktu kurang lebih 30 menit
<b>Mempengaruhi</b>	Jumlah konsumen yang dapat dilayani oleh petugas
<b>Akibat dari permasalahan</b>	Konsumen sulit dalam membuat keputusan Konsumen harus menunggu cukup lama untuk dilayani ketika <i>counter</i> ramai karena keterbatasan petugas
<b>Solusi yang diharapkan dapat</b>	Mempermudah konsumen dalam membuat keputusan. Mengurangi waktu tunggu konsumen.

#### 4.5 Analisis Kebutuhan

Bagian ini merupakan pendefinisian kebutuhan-kebutuhan baik fungsional maupun non fungsional dari sistem informasi *review smartphone* yang akan dibangun. Kebutuhan didapatkan dari hasil wawancara dengan pihak terkait yaitu TNT Cell dan hasil dari perancangan kebutuhan divalidasi kembali oleh pihak TNT Cell sebelum masuk kedalam proses perancangan.

### 4.5.1 Aturan Penomoran Kebutuhan

Pemberian kode pada analisis kebutuhan bertujuan untuk mempermudah pengembang sistem maupun pengguna dalam mengidentifikasi kebutuhan dari sistem yang akan dibangun. Aturan kode sistem informasi *review smartphone* produk *smartphone* ditunjukkan pada Gambar 4.6.



**Gambar 4.6 Aturan Penomoran Kode Kebutuhan**

### 4.5.2 Analisis Kebutuhan User

Kebutuhan user merupakan hasil dari analisis permasalahan dan kebutuhan dari sistem yang harus dipenuhi oleh sistem yang akan dibangun. Kebutuhan dibagi menjadi dua yang kebutuhan user dan kebutuhan sistem. Kebutuhan user merupakan kebutuhan layanan sistem yang diharapkan oleh user. Sedangkan kebutuhan sistem adalah kebutuhan yang mendeskripsikan layanan dari sistem untuk memenuhi kebutuhan user.

#### 4.5.2.1 Kebutuhan User (*User Requirement*)

Tabel 4.3 menunjukkan hasil identifikasi kebutuhan user yang didapat dari proses wawancara dan observasi dengan pihak TNT Cell.

**Tabel 4.3 Identifikasi Kebutuhan User**

No.	[SIRS-AK-01] Admin	
	Kode UR	Kebutuhan
1	SIRS-UR-01	Menambahkan data <i>Smartphone</i> baru. Data tersebut berisi jenis tipe, merk, spesifikasi, gambar, harga, dan review <i>online</i> mengenai <i>Smartphone</i> tersebut.
2	SIRS-UR-02	Mengelola data <i>Smartphone</i> yang sudah ada sebelumnya. Data tersebut berisi jenis tipe, merk, spesifikasi, gambar, harga, dan review <i>online</i> mengenai <i>Smartphone</i> tersebut.

3	SIRS-UR-03	Melihat deskripsi <i>Smartphone</i> . Deskripsi terdiri dari jenis tipe, merk, spesifikasi, gambar dan harga <i>Smartphone</i> .
4	SIRS-UR-04	Melihat review <i>Smartphone</i> beserta sentimen dari review yang ada.
5	SIRS-UR-05	Melihat nilai dari aspek <i>Smartphone</i>
	[SIRS-AK-02] Konsumen	
	Kode UR	Kebutuhan
6	SIRS-UR-06	Melihat deskripsi <i>Smartphone</i> . Deskripsi terdiri dari jenis tipe, merk, spesifikasi, gambar dan harga <i>Smartphone</i> .
7	SIRS-UR-07	Melihat review <i>Smartphone</i> beserta sentimen dari review yang ada.
8	SIRS-UR-08	Melihat nilai dari aspek <i>Smartphone</i>

#### 4.5.2.2 Kebutuhan Sistem (*System Requirement*)

Kebutuhan sistem merupakan hasil dari analisis kebutuhan user yang nantinya akan dijadikan sebagai fitur atau layanan yang ada pada sistem untuk memenuhi kebutuhan user yang sudah diidentifikasi sebelumnya.

**Tabel 4.4 Identifikasi Kebutuhan Sistem**

No.	Kode SR	Kebutuhan
1	SIRS-SR-01	Fungsi untuk menambah data <i>Smartphone</i> baru
2	SIRS-SR-02	Fungsi untuk mengelola data <i>Smartphone</i> yang sudah ada
3	SIRS-SR-03	Fungsi untuk autentifikasi pengguna sistem
4	SIRS-SR-04	Fungsi untuk menampilkan deskripsi <i>Smartphone</i>
5	SIRS-SR-05	Fungsi untuk menampilkan review <i>Smartphone</i> beserta sentimennya
6	SIRS-SR-06	Fungsi untuk menampilkan nilai aspek dari <i>Smartphone</i>
7	SIRS-SR-07	Fungsi untuk <i>scraping</i> review dan spesifikasi dari situs yang sudah ditentukan

8	SIRS-SR-08	Fungsi melakukan analisis sentimen review
---	------------	---

Tabel 4.4 menjelaskan hasil identifikasi kebutuhan sistem yang diperlukan untuk memenuhi kebutuhan user yang ada. Dari 8 kebutuhan user di dapatkan kebutuhan sistem sebanyak 8 kebutuhan.

### 4.5.3 Analisis Spesifikasi Kebutuhan Perangkat Lunak

Pada bagian ini, sistem informasi *review smartphone* di representasikan dengan kode SIRS. Spesifikasi ini merupakan deskripsi detail mengenai fungsi atau layanan dari sistem yang di dapat dari hasil analisis kebutuhan. Pada spesifikasi kebutuhan perangkat lunak, kebutuhan dibagi menjadi kebutuhan fungsional dan *non fungsional*.

#### 4.5.3.1 Kebutuhan Fungsional (*Functional Requirement*)

Kebutuhan fungsional merupakan kebutuhan sistem yang menggambarkan fungsi atau layanan yang harus tersedia pada sistem untuk memenuhi kebutuhan user yang ada. Tabel 4.5 menunjukkan hasil identifikasi kebutuhan fungsional.

**Tabel 4.5 Identifikasi Spesifikasi Kebutuhan Fungsional**

No.	[SIRS-AK-01] Admin			
	Kode UR	Kode Fungsi	Nama Fungsional	Deskripsi
1	SIRS-UR-01	SIRS-F-01-01	Tambah Data <i>Smartphone</i>	Sistem dapat digunakan untuk menambahkan data <i>smartphone</i> baru
		SIRS-F-01-02	<i>Scraping</i> Spesifikasi dan <i>Review</i>	Sistem dapat digunakan untuk mengambil spesifikasi dan <i>review smartphone</i> dari situs web yang sudah ditentukan
2	SIRS-UR-02	SIRS-F-02-01	Proses <i>Review</i>	Sistem dapat memproses <i>review</i> yaitu mengklasifikasikan <i>review</i> ke dalam klasifikasi positif atau negatif
		SIRS-F-02-02	Rubah Data <i>Smartphone</i>	Sistem dapat digunakan untuk merubah data <i>smartphone</i>
		SIRS-F-02-03	Hapus <i>Data Smartphone</i>	Sistem dapat digunakan untuk menghapus data <i>smartphone</i>



3	SIRS-UR-03	SIRS-F-03-01	Melihat <i>detail smartphone</i>	Sistem dapat digunakan untuk melihat <i>detail smartphone</i>
		SIRS-F-03-02	Cari data <i>smartphone</i>	Sistem dapat digunakan untuk mencari data <i>smartphone</i> berdasarkan merk dan tipe
		SIRS-F-03-03	Lihat <i>smartphone</i> terbaik	Sistem dapat digunakan untuk melihat daftar lima <i>smartphone</i> terbaik
4	SIRS-UR-04	SIRS-F-04-01	Lihat <i>Review</i>	Sistem dapat digunakan untuk melihat <i>review smartphone</i> dan klasifikasi dari <i>review</i>
5	-	SIRS-F-05-01	Tambah <i>user</i>	Sistem dapat digunakan untuk menambah pengguna sistem
		SIRS-F-05-02	<i>Login</i>	Sistem dapat mengautentifikasi hak ases pengguna
		SIRS-F-05-03	<i>Logout</i>	sistem memutuskan hak ases pengguna dari sistem
[SIRS-AK-02] Konsumen				
6	SIRS-UR-06	SIRS-F-03-01	Lihat <i>detail smartphone</i>	Sistem dapat digunakan untuk melihat <i>detail smartphone</i>
		SIRS-F-03-02	Cari data <i>smartphone</i>	Sistem dapat digunakan untuk mencari data <i>smartphone</i> berdasarkan merk dan tipe
		SIRS-F-03-03	Lihat <i>smartphone</i> terbaik	Sistem dapat digunakan untuk melihat daftar lima <i>smartphone</i> terbaik
7	SIRS-UR-07	SIRS-F-04-01	Lihat <i>Review</i>	Sistem dapat digunakan untuk melihat <i>review smartphone</i> dan klasifikasi dari <i>review</i>

#### 4.5.3.2 Kebutuhan *Non Fungsional*

Kebutuhan *non fungsional* merupakan kebutuhan sistem yang dapat mendukung agar fungsionalitas sistem dapat berjalan dengan baik. Tabel 4.6 menunjukkan hasil identifikasi spesifikasi kebutuhan *non fungsional* yang dihasilkan dari wawancara dengan pihak TNT Cell.

**Tabel 4.6 Identifikasi Spesifikasi Kebutuhan *Non Fungsional***

No.	Kode SRS	Spesifikasi Kebutuhan Perangkat Lunak
1	SIRS-NF-01	Sistem harus mudah digunakan oleh berbagai pengguna

#### 4.6 Analisis Hasil Rekayasa Kebutuhan

Rekayasa kebutuhan pada sistem informasi *review Smartphone* menghasilkan 7 kebutuhan user dan 8 kebutuhan sistem yang harus dipenuhi oleh sistem. Dengan jumlah kebutuhan fungsional sebanyak 12 kebutuhan fungsional dan 1 kebutuhan non fungsional. Jumlah kebutuhan fungsional 12 dikarenakan terdapat kebutuhan fungsional yang sama, dimana setiap kebutuhan yang sama dianggap sebagai satu kebutuhan. Hasil tersebut akan digunakan sebagai dasar untuk fase selanjutnya yaitu fase perancangan sistem. Sebelum memasuki fase perancangan, hasil analisis kebutuhan akan divalidasi kembali oleh pihak dari TNT Cell.

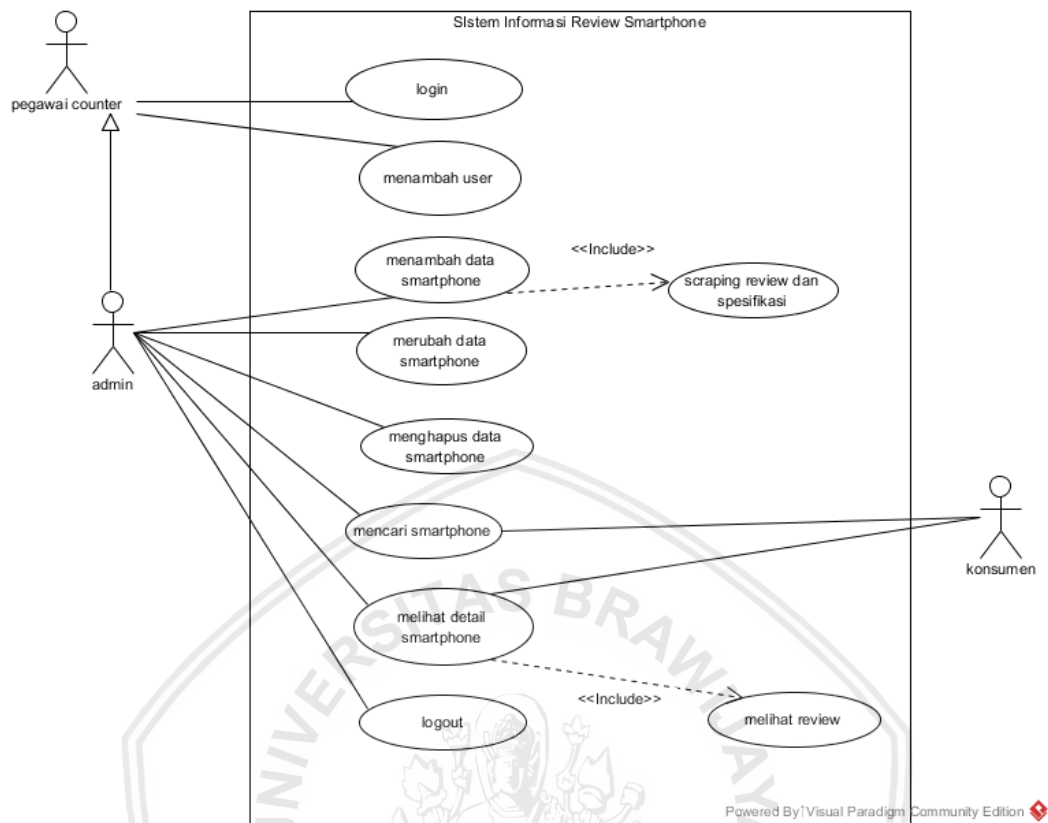
#### 4.7 Pemodelan Kebutuhan

Tahap ini merupakan tahap memodelkan kebutuhan hasil dari analisis kebutuhan pada sebelumnya. Pada tahap ini kebutuhan sistem di modelkan ke dalam pemodelan *Unified Modeling Language* (UML). Pemodelan UML yang digunakan pada tahap ini adalah *Use Case diagram*.

##### 4.7.1 Pemodelan *Use Case Diagram*

*Use Case diagram* adalah pemodelan yang menggambarkan sistem yang dilihat dari sudut pandang pengguna sistem. Pada *Use Case diagram* ini digambarkan fungsi-fungsi apa saja yang ada pada sistem yang akan dibangun. Gambar 4.7 menunjukkan diagram *Use Case* dari sistem informasi *review smartphone* pada TNT Cell Bojonegoro.





**Gambar 4.7 Use Case Diagram Sistem Review**

Gambar 4.7 menjelaskan diagram *Use Case* beserta aktor yang telah diidentifikasi pada bab analisis kebutuhan sistem. *Use Case* di dapatkan dari kebutuhan fungsional sedangkan aktor merupakan hasil dari analisis aktor. Dari diagram *Use Case* diatas di dapatkan bahwa sistem yang akan dikembangkan ini terdapat 10 *Use Case* dengan terdapat 2 aktor yang akan menggunakan sistem ini.

#### 4.7.2 Pemodelan *Use Case Scenario*

Pemodelan *Use Case Scenario* berisi detail penjelasan mengenai suatu *Use Case* yang mencakup deskripsi beserta alur dari masing-masing *Use Case*.

##### 4.7.2.1 *Use Case Scenario Login*

**Tabel 4.7 Use Case Scenario Login**

<i>Use Case Name</i>	Login
<i>Actor:</i>	Pegawai
<i>Description:</i>	Use case ini menjelaskan bagaimana pegawai masuk ke dalam sistem informasi review <i>Smartphone</i>
<i>Usecase Goal:</i>	Pegawai terautentifikasi sebagai admin dan berhasil masuk pada sistem

<i>Preconditions:</i>	Pegawai mengakses halaman untuk login
<i>Main Flow:</i>	<ol style="list-style-type: none"> <li>1. Pegawai memasukkan <i>username</i> dan <i>password</i></li> <li>2. Sistem mengautentifikasi <i>username</i> dan <i>password</i> yang dimasukkan oleh admin</li> </ol>
<i>Alternative Flow - 1:</i>	Proses login gagal, jika <i>username</i> dan <i>password</i> tidak sesuai dengan yang terdapat dalam sistem maka sistem akan menampilkan notifikasi login gagal
<i>Postconditions:</i>	Sistem menampilkan halaman data <i>Smartphone</i>

#### 4.7.2.2 Use Case Scenario Menambah Data Smartphone

Tabel 4.8 Use Scenario Menambah Data Smartphone

<i>Use Case Name</i>	Menambah data <i>smartphone</i>
<i>Actor:</i>	Admin
<i>Description:</i>	Use case ini menjelaskan bagaimana admin menambahkan data <i>Smartphone</i> baru ke dalam sistem
<i>Usecase Goal:</i>	Data berhasil di tambahkan ke dalam sistem
<i>Preconditions:</i>	Admin sudah berhasil masuk ke dalam sistem
<i>Main Flow:</i>	<ol style="list-style-type: none"> <li>1. Admin memilih tambah <i>smartphone</i></li> <li>2. Sistem akan menampilkan halaman tambah <i>smartphone</i></li> <li>3. Admin memasukkan data terkait <i>Smartphone</i>. Data tersebut berupa merk, tipe, harga, dan gambar dari <i>Smartphone</i> tersebut.</li> <li>4. Admin mengklik tombol <b>cari spesifikasi + review</b> untuk mengambil spesifikasi dan review tentang <i>Smartphone</i> tersebut</li> <li>5. Sistem menampilkan spesifikasi dan review yang berhasil di <i>scraping</i></li> <li>6. Admin mengklik tombol <b>simpan</b></li> <li>7. Sistem menyimpan data</li> </ol>
<i>Alternative Flow - 1:</i>	Apabila review <i>Smartphone</i> tidak tersedia pada <i>website</i> yang sudah ditentukan, maka saat admin mengklik tombol <b>cari</b>

	<b>spesifikasi + review</b> maka sistem tidak akan menampilkan review
<i>Postconditions:</i>	Sistem menampilkan halaman data <i>Smartphone</i>

#### 4.7.2.3 Use Case Scenario Merubah Data Smartphone

**Tabel 4.9 Use Case Scenario Merubah Data Smartphone**

<i>Use Case Name</i>	Merubah Data <i>Smartphone</i>
<i>Actor:</i>	Admin
<i>Description:</i>	Use case ini menjelaskan bagaimana admin merubah data <i>Smartphone</i>
<i>Usecase Goal:</i>	Data berhasil dirubah
<i>Preconditions:</i>	Sistem menampilkan halaman data <i>Smartphone</i>
<i>Main Flow:</i>	<ol style="list-style-type: none"> <li>1. Admin memilih <i>edit</i></li> <li>2. Sistem menampilkan form edit data <i>Smartphone</i></li> <li>3. Admin merubah data pada form</li> <li>4. Admin memilih <i>simpan</i></li> </ol> <p>Sistem merubah data</p>
<i>Alternative Flow - 1:</i>	-
<i>Postconditions:</i>	Sistem menampilkan halaman data <i>Smartphone</i>

#### 4.7.2.4 Use Case Scenario Menghapus Data Smartphone

**Tabel 4.10 Use Case Scenario Menghapus Smartphone**

<i>Use Case Name</i>	Menghapus data <i>Smartphone</i>
<i>Actor:</i>	Admin
<i>Description:</i>	Use case ini menjelaskan bagaimana admin dapat menghapus data <i>Smartphone</i>
<i>Usecase Goal:</i>	Data terhapus
<i>Preconditions:</i>	Sistem menampilkan halaman data <i>Smartphone</i>
<i>Main Flow:</i>	<ol style="list-style-type: none"> <li>1. Admin memilih <i>delete</i></li> <li>2. Sistem menampilkan notifikasi konfirmasi apakah benar akan menghapus data</li> <li>3. Admin memilih OK</li> </ol>

	4. Sistem menghapus data
<i>Alternative Flow - 1:</i>	Apabila admin memilih Cancel maka kembali pada halaman data <i>Smartphone</i>
<i>Alternative Flow - 2:</i>	-
<i>Postconditions:</i>	Sistem menampilkan halaman data <i>Smartphone</i>

#### 4.7.2.5 Use Case Scenario *Memproses Review*

**Tabel 4.11 Use Case Scenario Memproses Review**

<i>Use Case Name</i>	Memproses Review
<i>Actor:</i>	Admin
<i>Description:</i>	Use case ini menjelaskan bagaimana admin memproses review, sehingga review dapat diklasifikasikan menjadi positif dan negatif
<i>Usecase Goal:</i>	Review berhasil diklasifikasikan menjadi positif dan negatif
<i>Preconditions:</i>	Sistem menampilkan halaman deskripsi <i>Smartphone</i>
<i>Main Flow:</i>	<ol style="list-style-type: none"> <li>1. Memilih review</li> <li>2. Sistem menampilkan halaman review</li> <li>3. Memilih <b><i>process review</i></b></li> <li>4. Sistem memproses review</li> </ol>
<i>Alternative Flow - 1:</i>	<ol style="list-style-type: none"> <li>1. Pada halaman data <i>Smartphone</i>, admin memilih <b><i>process</i></b></li> </ol> <p>Sistem memproses review</p>
<i>Postconditions:</i>	Sistem menampilkan halaman <i>review</i> beserta review yang sudah di klasifikasikan

#### 4.7.2.6 Use Case Scenario *Mencari Data Smartphone*

**Tabel 4.12 Use Case Scenario Mencari Data Smartphone**

<i>Use Case Name</i>	Mencari Data <i>Smartphone</i>
<i>Actor:</i>	Admin, Konsumen
<i>Description:</i>	Use case ini menjelaskan bagaimana admin atau konsumen mencari data <i>smartphone</i> yang diinginkan
<i>Usecase Goal:</i>	Sistem menampilkan data sesuai yang diinginkan aktor
<i>Preconditions:</i>	Sistem menampilkan halaman home

<i>Main Flow:</i>	<ol style="list-style-type: none"> <li>1. <i>Actor</i> memasukkan kata kunci pencarian (<i>keyword</i>) pada kolom <i>search</i></li> <li>2. <i>Actor</i> menekan <i>ENTER</i> atau <i>icon search</i></li> <li>3. Sistem menampilkan halaman data <i>smartphone</i> sesuai dengan kata kunci pencarian</li> </ol>
<i>Alternative Flow - 1:</i>	-
<i>Postconditions:</i>	-

#### 4.7.2.7 Use Case Scenario *Melihat Detail Smartphone*

**Tabel 4.13 Use Case Scenario *Melihat Detail Smartphone***

<i>Use Case Name</i>	Melihat Detail <i>Smartphone</i>
<i>Actor:</i>	Admin, Konsumen
<i>Description:</i>	Use case ini menjelaskan bagaimana admin, pegawai dan konsumen dapat melihat detail <i>Smartphone</i>
<i>Usecase Goal:</i>	Sistem menampilkan detail data <i>Smartphone</i>
<i>Preconditions:</i>	Sistem menampilkan halaman home
<i>Main Flow:</i>	<ol style="list-style-type: none"> <li>1. <i>Actor</i> memilih salah satu gambar <i>Smartphone</i> pada halaman home</li> </ol> <p>Sistem menampilkan halaman detail <i>Smartphone</i></p>
<i>Alternative Flow - 1:</i>	Apabila review <i>Smartphone</i> tidak terdapat pada sistem maka pada tabel review tidak ada review
<i>Postconditions:</i>	-

#### 4.7.2.8 Use Case Scenario *Scraping Spesifikasi dan Review*

**Tabel 4.14 Use Case Scenario *Scraping Spesifikasi dan Review***

<i>Use Case Name</i>	<i>Scraping Spesifikasi dan Review</i>
<i>Actor:</i>	Admin
<i>Description:</i>	Use case ini menjelaskan bagaimana admin mengambil data ( <i>scraping</i> ) spesifikasi dan <i>review smartphone</i> dari situs web yang sudah ditentukan
<i>Usecase Goal:</i>	Jumlah <i>people ask</i> bertambah
<i>Preconditions:</i>	Sistem menampilkan halaman tambah data <i>smartphone</i>
<i>Main Flow:</i>	<ol style="list-style-type: none"> <li>1. Admin memilih <b>cari spesifikasi + review</b></li> </ol>

	2. Sistem mengambil data dari situs web
<i>Alternative Flow - 1:</i>	-
<i>Postconditions:</i>	Sistem menampilkan spesifikasi dan review yang di dapat proses <i>scraping</i> pada form tambah data <i>smartphone</i>

#### 4.7.2.9 Use Case Scenario Melihat Review

**Tabel 4.15 Use Case Scenario Melihat Review**

<i>Use Case Name</i>	Melihat Review
<i>Actor:</i>	Admin,Konsumen
<i>Description:</i>	Use case ini menjelaskan bagaimana admin atau konsumen dapat melihat review dari <i>smartphone</i>
<i>Usecase Goal:</i>	Sistem menampilkan <i>review smartphone</i>
<i>Preconditions:</i>	Sistem menampilkan halaman detail <i>smartphone</i>
<i>Main Flow:</i>	1. Sistem menampilkan <i>review smartphone</i> pada tabel <i>review</i>
<i>Alternative Flow - 1:</i>	- Jika tidak ada <i>review</i> , maka tabel <i>review</i> kosong
<i>Postconditions:</i>	-

#### 4.7.2.10 Use Case Scenario Logout

**Tabel 4.16 Use Case Scenario Logout**

Use Case Name	Logout
Actor:	Admin
Description:	Use case ini menjelaskan bagaimana admin keluar dari sistem informasi review <i>Smartphone</i>
Usecase Goal:	Aktor berhasil keluar dari sistem
Preconditions:	Login berhasil
Main Flow:	1. Aktor memilih fungsi logout pada sistem 2. Keluar dari sistem
Alternative Flow - 1:	-
Postconditions:	Sistem menampilkan halaman home



#### 4.7.2.11 Use Case Scenario *Menambah User*

Tabel 4.17 *Use Case Scenario Menambah User*

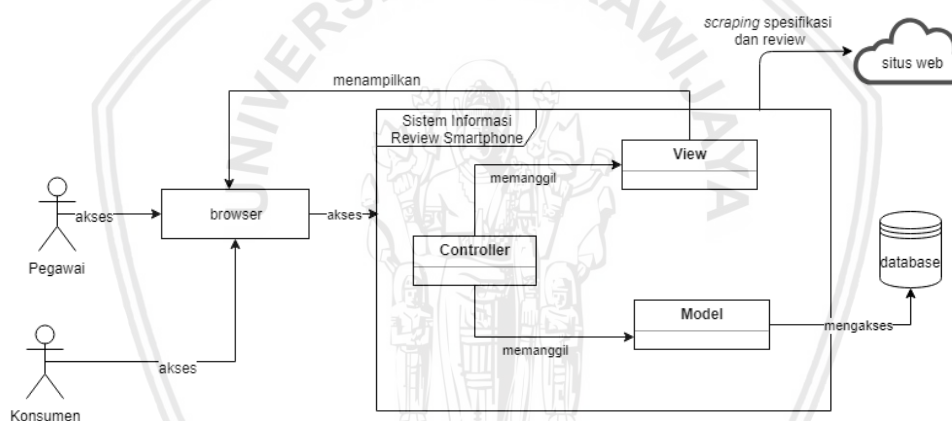
<i>Use Case Name</i>	Menambah User
<i>Actor:</i>	Admin
<i>Description:</i>	Use case ini menjelaskan bagaimana admin atau petugas dapat menambah user sistem
<i>Usecase Goal:</i>	Aktor berhasil keluar dari sistem
<i>Preconditions:</i>	Sistem menampilkan halaman login
<i>Main Flow:</i>	<ol style="list-style-type: none"><li>1. Aktor memilih sign up</li><li>2. Sistem menampilkan halaman add user</li><li>3. Aktor mengisi data pada form</li><li>4. Aktor memilih submit</li></ol>
<i>Alternative Flow - 1:</i>	- Jika aktor memilih cancel, sistem kembali menampilkan halaman login
<i>Alternative Flow - 2:</i>	-
<i>Postconditions:</i>	Sistem menampilkan halaman login

## BAB 5 PERANCANGAN

Perancangan sistem merupakan tahap pemodelan dari kebutuhan sistem yang telah di analisis sebelumnya. Pemodelan sistem pada tahap ini menggunakan pemodelan UML. Model diagram yang digunakan pada tahap perancangan sistem adalah *Sequence diagram* dan *Class diagram*. Tahap ini juga mencakup proses perancangan arsitektur sistem, perancangan skema basis data, perancangan antar muka, dan perancangan algoritme.

### 5.1 Perancangan Arsitektur Sistem

Tahap pertama dari perancangan sebuah sistem adalah tahap perancangan arsitektur sistem. Tahap ini menggambarkan komponen-komponen beserta hubungan interaksi antar komponen pada sistem yang akan dibangun. Pada bagian ini akan proses perancangan arsitektur sistem bertujuan untuk menggambarkan interaksi setiap komponen dari Sistem Informasi *Review Smartphone*.



**Gambar 5.1 Rancangan Arsitektur Diagram Sistem Informasi *Review Smartphone***

Pada Gambar 5.1 menunjukan komponen beserta interaksi antar komponen pada sistem yang akan dibangun. Gambar tersebut menjelaskan bahwa baik pengguna maupun konsumen dapat mengakses sistem melalui *browser* dikarenakan Sistem Informasi *Review Smartphone* merupakan sistem berbasis *website*. *Website* Sistem Informasi *Review Smartphone* dikembangkan dengan menerapkan pola perancangan *MVC (Model-View-Controller)*, oleh karena itu dalam proses pengembangan sistem menggunakan *framework CodeIgniter*.

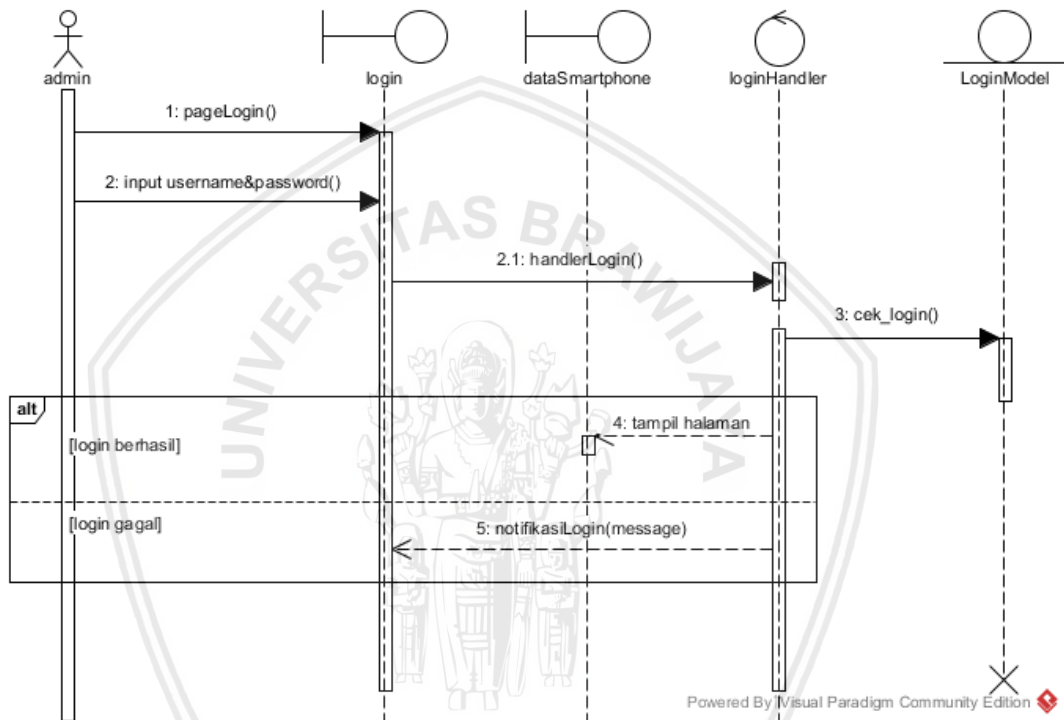
### 5.2 Perancangan *Sequence Diagram*

Pada bagian ini alur proses yang terjadi pada Sistem Informasi *Review Smartphone* akan digambarkan dalam bentuk *sequence diagram*. *Sequence diagram* digunakan karena sesuai untuk menggambarkan alur dari sebuah sistem yang dikembangkan menggunakan pemrograman berorientasi objek. *Sequence*

diagram dibuat dengan mengacu *Use Case diagram* yang sudah di gambarkan pada tahap pemodelan kebutuhan.

### 5.2.1 Sequence Diagram Login

*Sequence diagram login* menggambarkan alur dari proses autentikasi *user* yang akan masuk ke dalam sistem. Alur di mulai dari sistem menampilkan halaman login, lalu aktor pegawai akan memasukkan *username* dan *password* pada form login. Selanjutnya halaman login akan memanggil fungsi *handlerLogin()* pada *controller LoginHandler*. Setelah itu *LoginHandler* akan memanggil fungsi *cek\_user()* dari *LoginModel*.



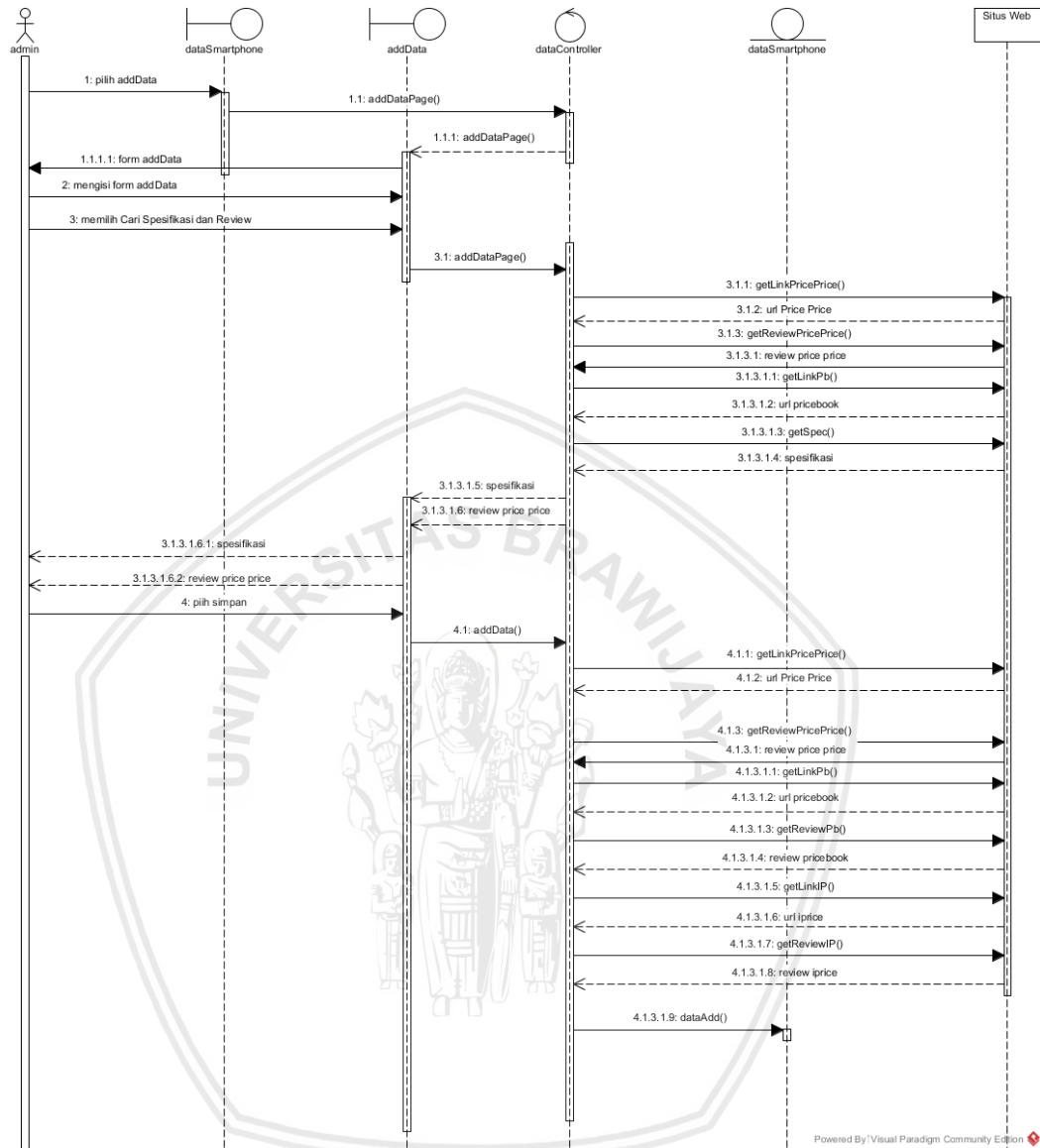
**Gambar 5.2 Sequence Diagram Login**

Pada Gambar 5.2 juga menunjukkan alur alternatif jika login gagal maka *controller LoginHandler* akan mengirimkan notifikasi login gagal.

### 5.2.2 Sequence Diagram Menambah Data Smartphone

Diagram *Sequence* menambah data *smartphone* dimulai dengan admin memilih menu tambah data *smartphone*, lalu *boundary dataSmartphone* akan menjalankan fungsi *addDataPage()*, dengan begitu *dataController* akan menampilkan *addData*. Admin kemudian mengisi form pada *addData* dan memilih *Cari Spesifikasi dan Review*, *addData* akan memanggil *addDataPage()*. Kemudian *dataController* memanggil fungsi *getLinkPricePrice()*, *getReviewPrice()*, *getLinkPb()*, dan *getSpec()* secara berurutan. Kemudian *dataController* mengirimkan spesifikasi dan review. Setelah itu admin memilih *simpan*, dan fungsi *addData()* di jalankan. *dataController* akan kembali menjalankan fungsi *getLinkPricePrice()*, *getReviewPrice()*, *getLinkPb()*, *getReviewPb()*, *getLinkIP()*,

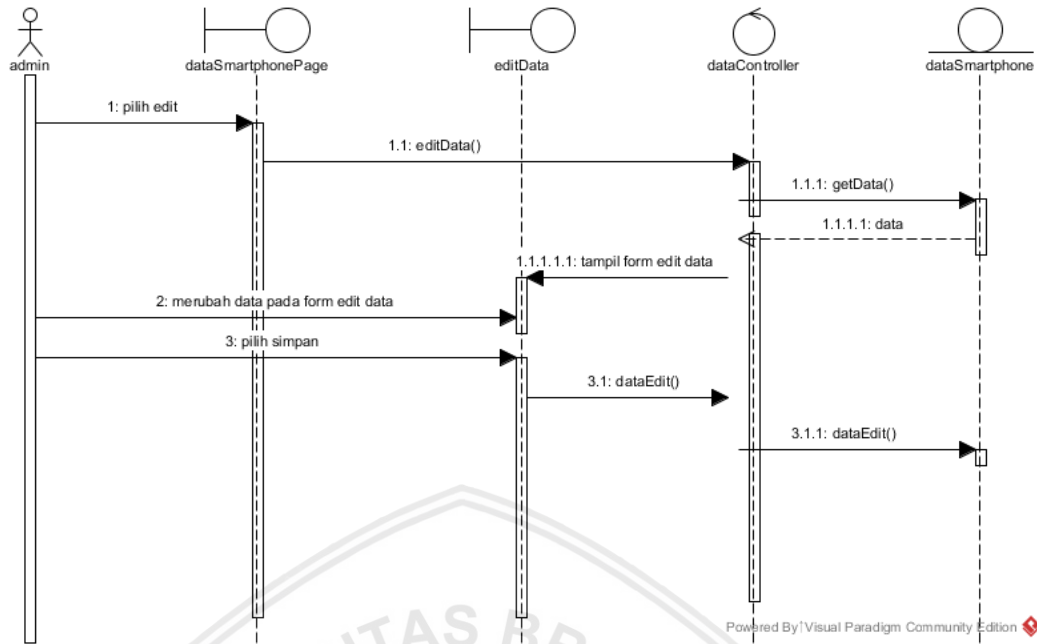
getReviewIP(), getSpec() secara berurutan. Setelah itu dataController memanggil fungsi dataAdd() pada dataSmartphone.



Gambar 5.3 Sequence Diagram Menambah Data

### 5.2.3 Sequence Diagram Mengubah Data Smartphone

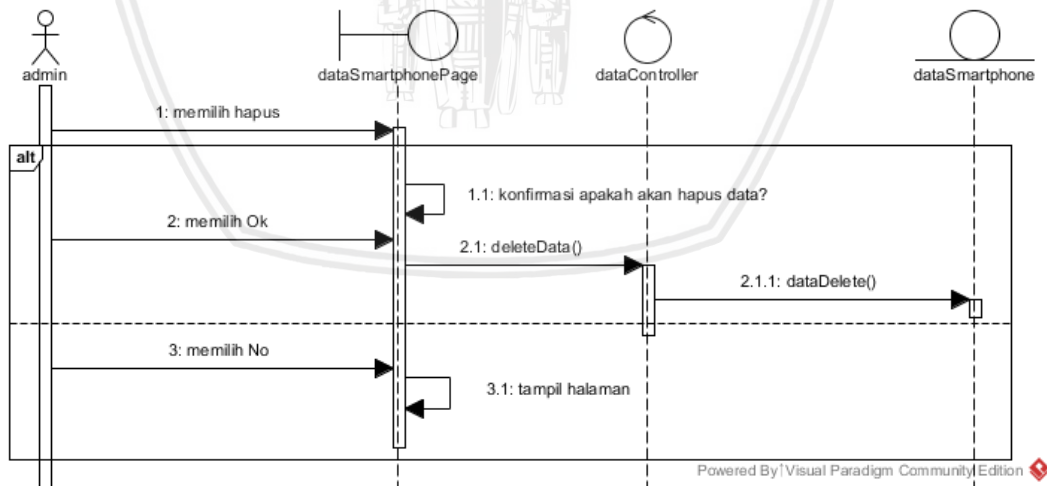
Sequence diagram mengubah data smartphone mengacu pada Use Case diagram mengubah data smartphone. Alur sequence diagram mengubah data smartphone dimulai dengan admin memilih menu edit kemudian dataController() menjalankan editData(). Kemudian akan tampil form pada editData, selanjutnya admin mengubah data pada form. Setelah itu, fungsi editData() dijalankan dan fungsi dataEdit() dijalankan juga.



Gambar 5.4 Sequence Diagram Mengubah Data Smartphone

### 5.2.4 Sequence Diagram Menghapus Data Smartphone

Sequence diagram menghapus data smartphone mengacu pada Use Case diagram menghapus data smartphone. Dimulai dari admin yang memilih menu hapus, kemudian dataSmartphone mengirimkan konfirmasi. Jika admin memilih Ok maka dataSmartphone memanggil fungsi deleteData(), kemudian dataController akan memanggil fungsi dataDelete().

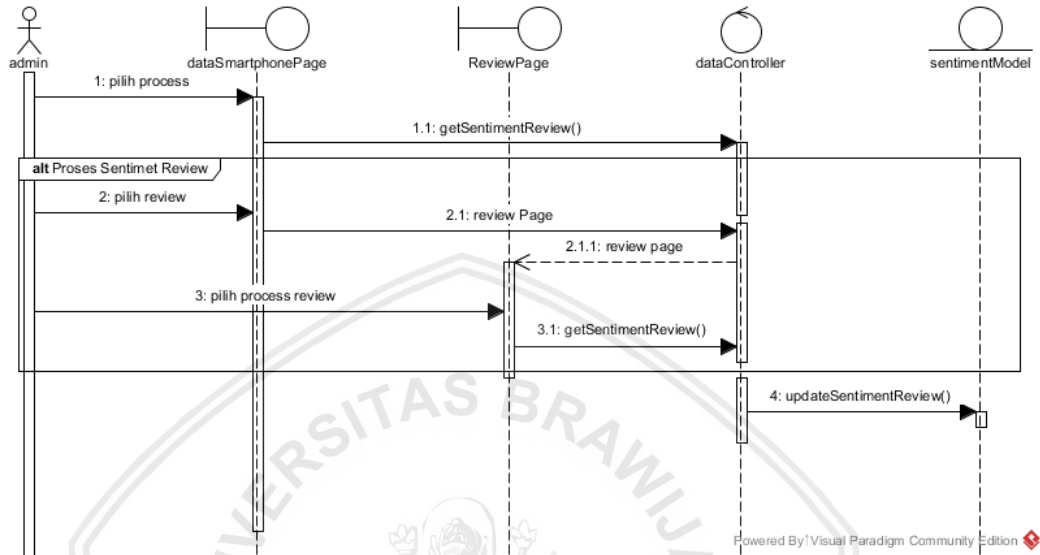


Gambar 5.5 Sequence Diagram Menghapus Data Smartphone

Gambar 5.5 menunjukkan alternatif jika admin memilih No pada konfirmasi maka dataSmartphone akan menampilkan halaman

### 5.2.5 Sequence Diagram Memproses Review

Sequence diagram memproses review mengacu pada Use Case diagram memproses review. Sequence dimulai dari admin memilih process kemudian dataSmartphone memanggil fungsi getSentimentReview(). Setelah itu dataController memanggil fungsi updateSentimentReview().



Gambar 5.6 Sequence Diagram Memproses Review

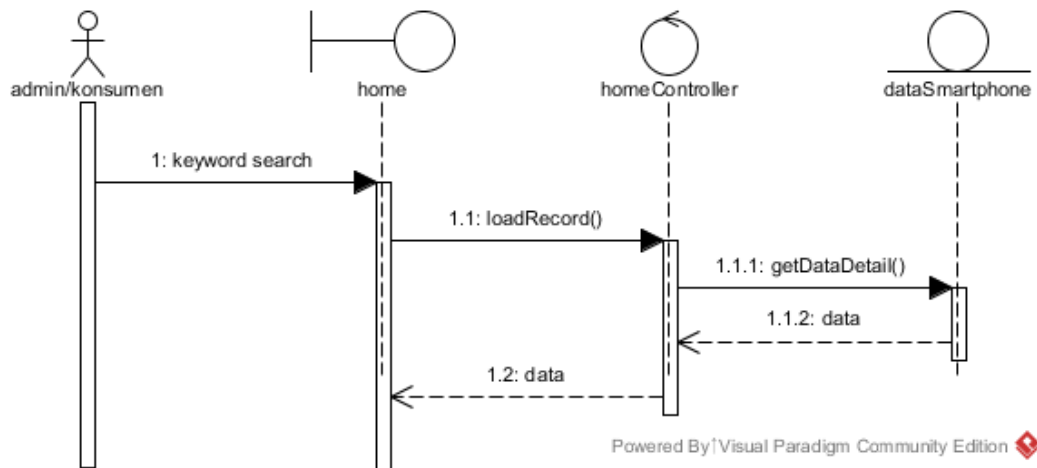
Gambar 5.6 juga dijelaskan alternatif jika admin memilih review pada dataSmartphone. Setelah itu dataSmartphone memanggil reviewPage(). Kemudian admin memilih process review. Selanjutnya ReviewPage memanggil getSentimentReview(), dan kemudian dataController memanggil fungsi updateSentimentReview().

### 5.2.6 Sequence Diagram Mencari Data Smartphone

Sequence diagram mencari data smartphone diawali oleh aktor yaitu admin atau konsumen memasukkan kata kunci pencarian (keyword search), kemudian home memanggil fungsi loadRecord() dan dilanjutkan homeController dengan memanggil fungsi getDataDetail() yang mana kemudian datanya di tampilkan pada home. Gambar 5.6 menunjukkan Sequence diagram mencari data smartphone.



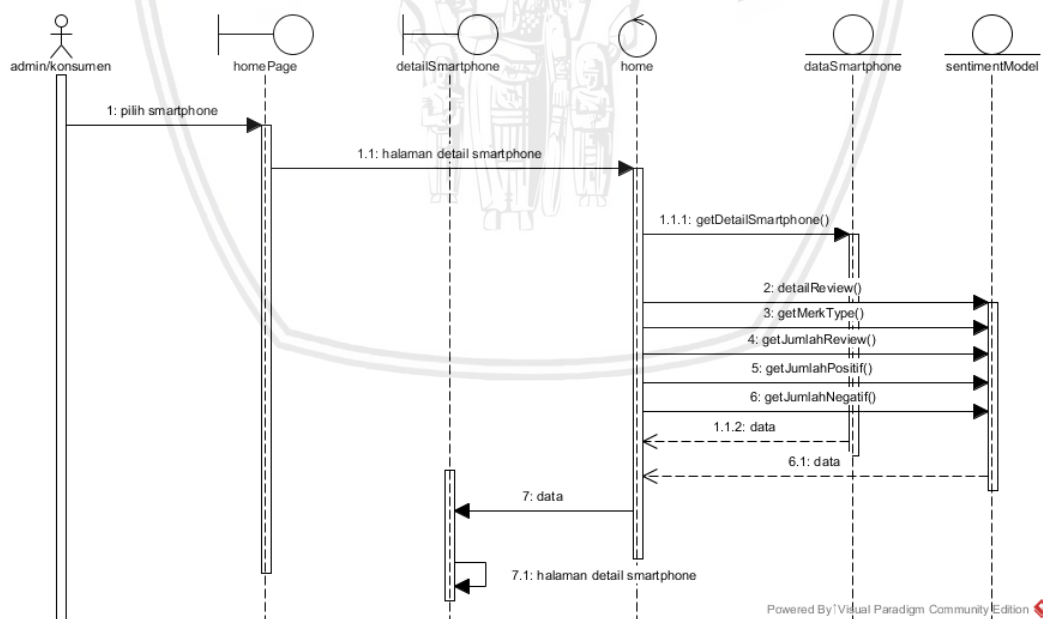




Gambar 5.7 Sequence Diagram Mencari Data Smartphone

### 5.2.7 Sequence Diagram Melihat Detail Smartphone

Alur pada *sequence diagram* melihat detail *smartphone* diawali dengan aktor baik itu admin, petugas ataupun konsumen memilih *smartphone*. Selanjutnya *home* memanggil fungsi *detailSmartphone()*, dan kemudian *homeController* memanggil fungsi *getDetailSmartphone* dari *dataSmartphone*, *detailReview()*, *getMerkType()*, *getJumlahReview()*, *getJumlahPositif()*, *getJumlahNegatif()* dari *SentimentModel* secara berurutan. Lalu dilanjutkan memuat *detailSmartphone* yang memuat data detail *smartphone*.

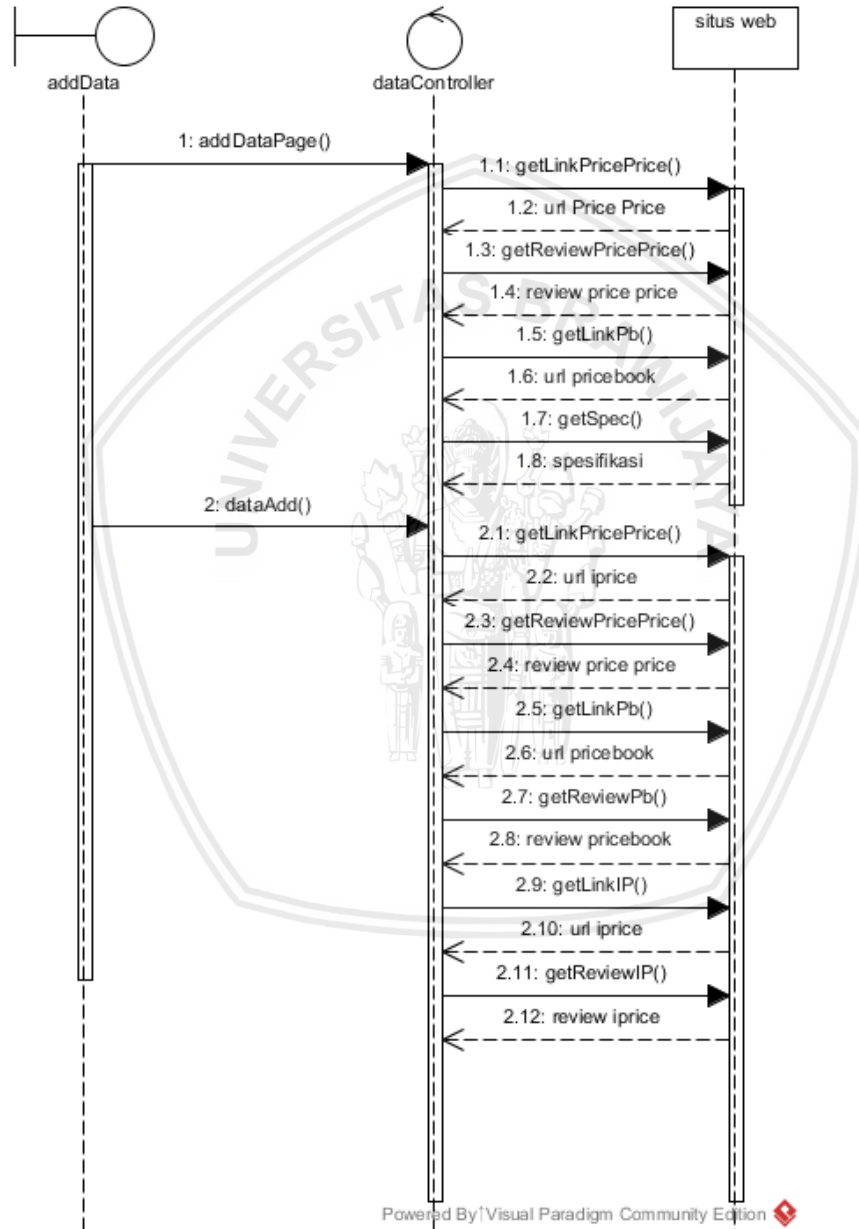


Gambar 5.8 Sequence Diagram Melihat Detail Smartphone

### 5.2.8 Sequence Diagram Scraping Spesifikasi dan Review

*Sequence diagram scraping* spesifikasi dan *review* menggambarkan alur sistem dalam mendapatkan spesifikasi dan *review* dari situs web yang sudah ditentukan. Proses ini merupakan salah satu bagian proses dari menambah data *smartphone*.

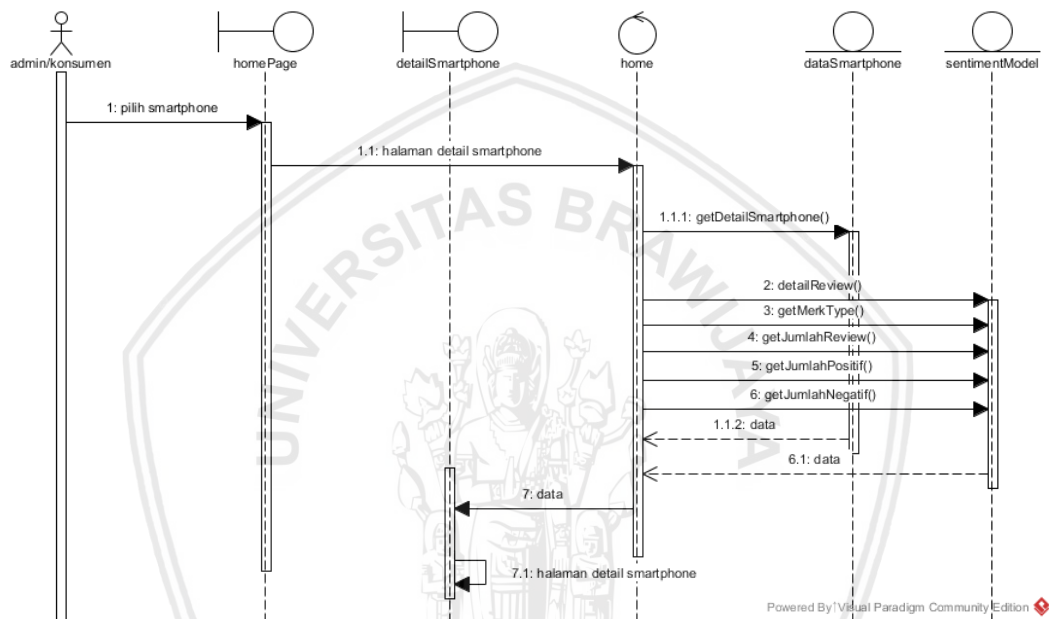
Sequence *scraping* spesifikasi dan *review* dimulai dari `addData` memanggil fungsi `addDataPage()`. Kemudian `dataController` melanjutkan dengan memanggil fungsi `getLinkPricePrice()`, `getReviewPricePrice()`, `getLinkPb()`, dan `getSpec()` secara berurutan, selanjutnya `addData` memanggil `dataAdd()`. `dataController` kembali melanjutkan dengan memanggil fungsi `getLinkPricePrice()`, `getReviewPricePrice()`, `getLinkPb()`, `getReviewPb()`, `getLinkIP()`, dan `getReviewIP()` secara berurutan. Dimana dari fungsi tersebut akan didapatkan spesifikasi dan *review* dari ketiga situs web yang sudah di tentukan.



Gambar 5.9 Sequence Diagram Scraping Spesifikasi dan Review

### 5.2.9 Sequence Diagram Melihat Review

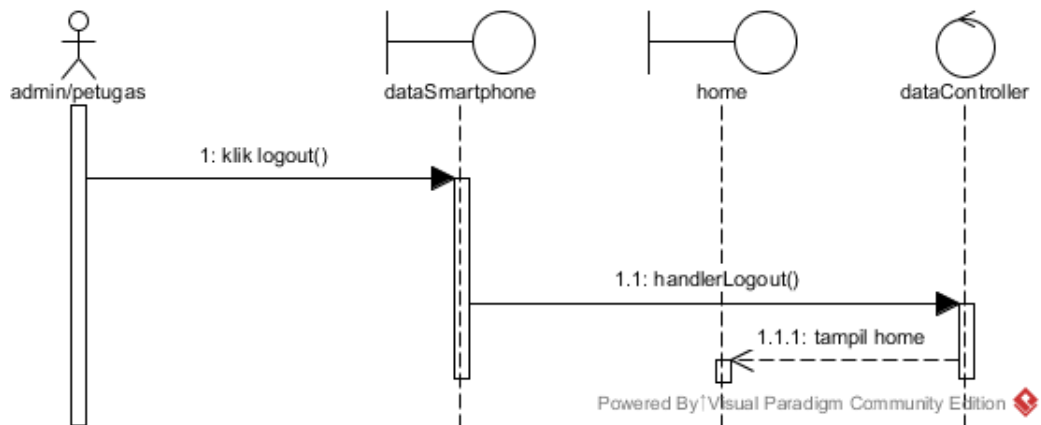
Sequence diagram melihat review mengacu pada use case diagram melihat review yang include pada use case melihat detail smartphone. Sequence melihat review dimulai dengan aktor yaitu baik admin atau konsumen memilih smartphone pada home. Kemudian home akan memanggil fungsi detailSmartphone() dari dataController, selanjutnya untuk mendapatkan review dataController memanggil fungsi getReview(), getMerkType(), getJumlahReview(), getJumlahPositif(), getJumlahNegatif() secara berurutan dari sentimentModel. Dari pemanggilan fungsi – fungsi tersebut di dapatkan data review yang ditampilkan pada home\_detailSmartphone.



Gambar 5.10 Sequence Diagram Melihat Review

### 5.2.10 Sequence Diagram Logout

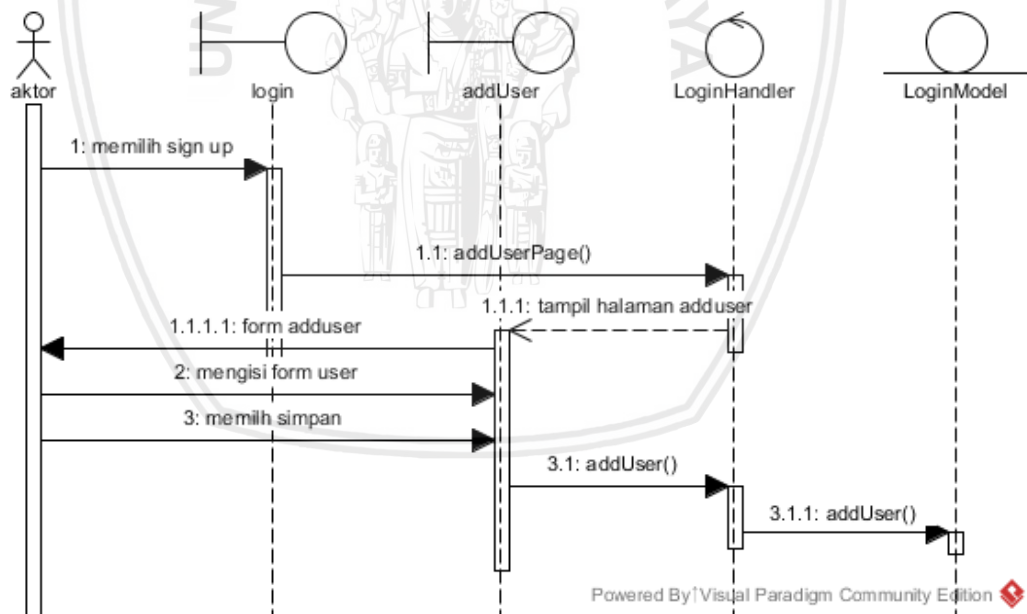
Sequence diagram logout mengacu pada use case diagram logout. Sequence diagram ini menggambarkan alur aktor baik admin atau petugas keluar dari sistem. Dimulai dari aktor memilih logout, kemudian home memanggil fungsi HandlerLogout() yang akan langsung menampilkan home setelah aktor keluar dari sistem. Gambar 5.11 menunjukkan Sequence diagram logout.



Gambar 5.11 Sequence Diagram Logout

### 5.2.11 Sequence Diagram Menambah User

Mengacu pada *use case diagram* menambah *user*, *sequence diagram* menambah *user* diawali dengan aktor pegawai memilih *sign up*, kemudian dilanjutkan login memanggil fungsi *addUserPage()* yang akan menampilkan *addUser*. Selanjutnya aktor pegawai mengisi form dan memilih *simpan*, berikutnya *addUser* akan memanggil fungsi *addUser()* dan dilanjutkan kembali oleh *LoginHandler* yang memanggil fungsi *addUser()* dari *LoginModel*.



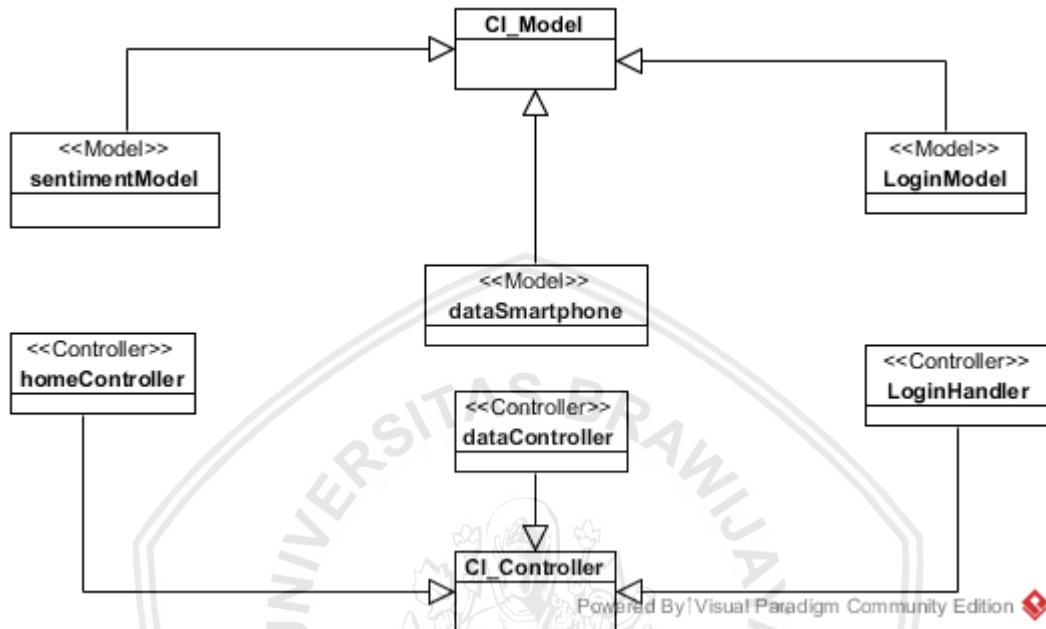
Gambar 5.12 Sequence Diagram Menambah User

### 5.3 Pemodelan Class Diagram

Bagian ini menjelaskan gambaran *class-class* atau *object-object* apa saja yang saling berhubungan dalam Sistem Informasi *Review Smartphone*. Pada pemodelan *Class diagram* ini juga akan menggambarkan *Class diagram* yang akan dijadikan sebagai acuan *model* dalam pemodelan skema basis data.

### 5.3.1 Diagram Relasi Antar Kelas

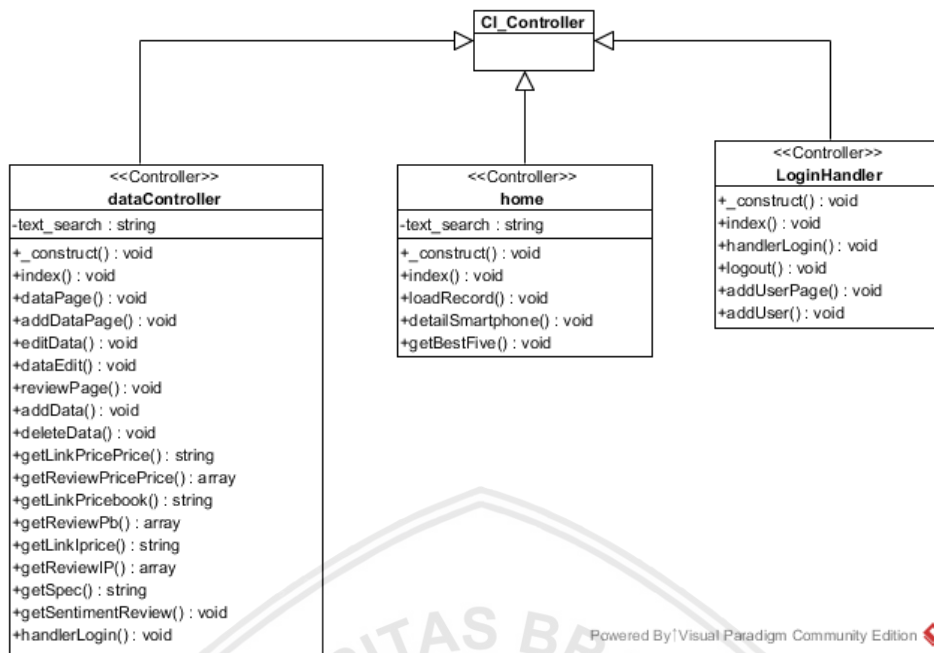
Pada bagian ini menggambarkan relasi atau hubungan antar *class* pada Sistem Informasi *Review Smartphone*. Gambar 5.13 menunjukkan terdapat dua *class generalisasi* yaitu *CI\_Controller* dan *CI\_Model*, dimana setiap *class* tersebut merupakan *generalisasi* dari tiga *class*.



Gambar 5.13 Class Diagram Relasi Antar Kelas

### 5.3.2 Diagram Relasi Antar Kelas Controller

Bagian ini merupakan penggambaran relasi atau hubungan antar *class controller* yang terdapat pada Sistem Informasi *Review Smartphone*. Terdapat tiga kelas *controller* pada sistem ini yaitu *homeController*, *dataController*, dan *LoginHandler*. Gambar 5.14 menunjukkan gambaran dari diagram relasi antar kelas *controller*.



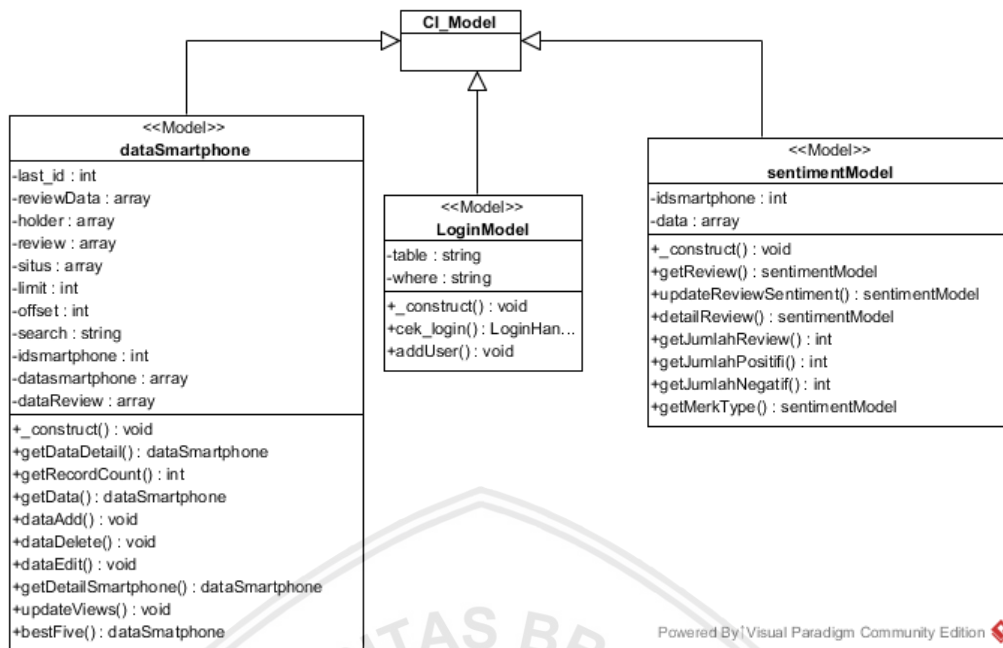
**Gambar 5.14 Diagram Relasi Antar Kelas Controller**

Pada Gambar 5.14 menjelaskan relasi antar *class controller*, dimana terdapat kelas *dataController* yang merupakan kelas *controller* untuk menampilkan keseluruhan informasi data *smartphone* dan digunakan operasi-operasi yang berhubungan dengan admin seperti proses menambah data, merubah data, maupun menghapus *smartphone*. Pada kelas *dataController* juga terdapat operasi untuk mendapatkan (scraping) data spesifikasi dan review dari situs web yang ditentukan. Selain itu kelas tersebut juga merupakan kelas yang digunakan untuk operasi proses sentimen dari *review*. Selanjutnya kelas *home* merupakan kelas yang menampilkan segala informasi *smartphone* pada halaman *home* dan halaman *home\_detailSmartphone*. Kelas *LoginHandler* merupakan kelas *controller* yang digunakan untuk segala operasi autentifikasi seperti login atau logout. Kelas *LoginHandler* juga digunakan untuk operasi proses menambah *user*.

### 5.3.3 Diagram Relasi Antar Kelas Model

Pada bagian ini menggambarkan hubungan atau relasi antara kelas model yang terdapat pada Sistem Informasi *Review Smartphone*. Terdapat tiga kelas model yang digunakan pada Sistem Informasi *Review Smartphone* yaitu *dataSmartphone*, *LoginModel*, dan *SentimentModel*. Gambar 5.15 menunjukkan gambaran dari diagram relasi antar kelas model pada Sistem Informasi *Review Smartphone*.





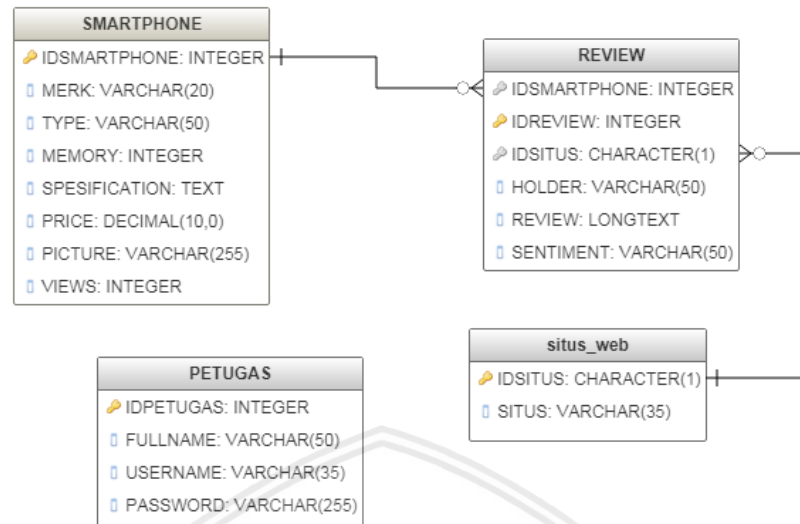
Gambar 5.15 Diagram Relasi Antar Kelas Model

Gambar 5.15 menunjukkan gambaran hubungan atau relasi antar kelas model. Kelas *dataSmartphone* merupakan kelas model yang digunakan untuk mengolah operasi-operasi yang berhubungan dengan data-data mengenai *smartphone*. Seperti menambah data, merubah data, menghapus, atau menampilkan data *smartphone*. Selanjutnya kelas *LoginModel* merupakan kelas model yang digunakan untuk segala operasi mengenai *user*, seperti autentikasi user yaitu proses login atau operasi menambah user. Kelas *SentimentModel* merupakan kelas yang digunakan untuk operasi-operasi yang berhubungan dengan *review* seperti melakukan perubahan nilai sentimen dari review, menampilkan jumlah *review*, jumlah *review* positif, dan jumlah *review* negatif.

#### 5.4 Perancangan Skema Basis Data

Bagian ini menggambarkan skema basis data yang akan digunakan sebagai acuan dalam pembuatan basis data untuk Sistem Informasi *Review Smartphone*. Skema basis data dimodelkan dengan menggunakan pemodelan *physical data model* (PDM). Perancangan skema basis data pada penelitian ini menggunakan aplikasi perancangan yaitu *genmiMyModel*, yang mana pada aplikasi tersebut menyediakan persegi sebagai gambaran dari sebuah tabel. Penggambaran *Primary Keys* pada aplikasi ini digambarkan dengan ikon kunci berwarna biru, sedangkan *Foreign Keys* digambarkan dengan menggunakan ikon kunci berwarna abu-abu.





**Gambar 5.16 Physical Data Model Sistem Informasi Review Smartphone**

Pada Gambar 5.16 diatas merupakan *physical data model* dari Sistem Informasi *Review Smartphone* yang mengacu pada kelas diagram relasi antar model. Dimana didapatkan empat tabel yaitu *smartphone*, *review*, *situs\_web* dan *petugas*. Tabel *review* memiliki relasi dengan tabel *smartphone*, dengan kolom *IDSMARTPHONE* tabel *review* sebagai *foreign keys*.

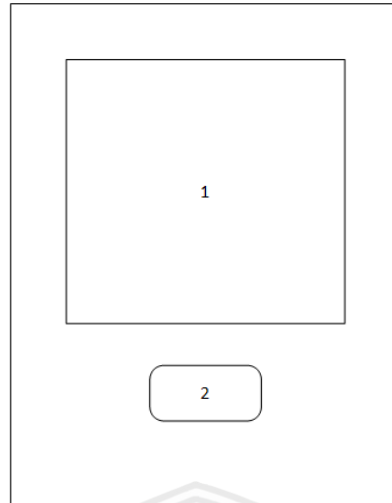
## 5.5 Perancangan Antar Muka Pengguna

Pada bagian ini menunjukkan perancangan antar muka pengguna untuk menggambarkan tampilan dari sistem yang akan dibangun.

### 5.5.1 Rancangan Antar Muka Halaman Login

Gambar 5.17 merupakan gambar rancangan untuk halaman *login*. Keterangan dari gambar 5.17 antara lain :

1. Form *Login*
2. Tombol *Login*

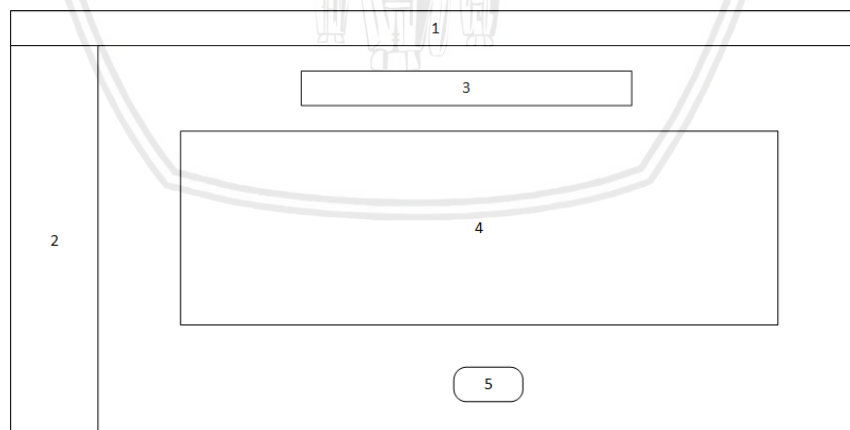


Gambar 5.17 Rancangan Antar Muka *Login*

### 5.5.2 Rancangan Antar Muka Halaman Tambah dan Rubah *Data Smartphone*

Rancangan antar muka halaman menambah dan merubah data *smartphone* digambarkan pada Gambar 5.18, dengan keterangan sebagai berikut :

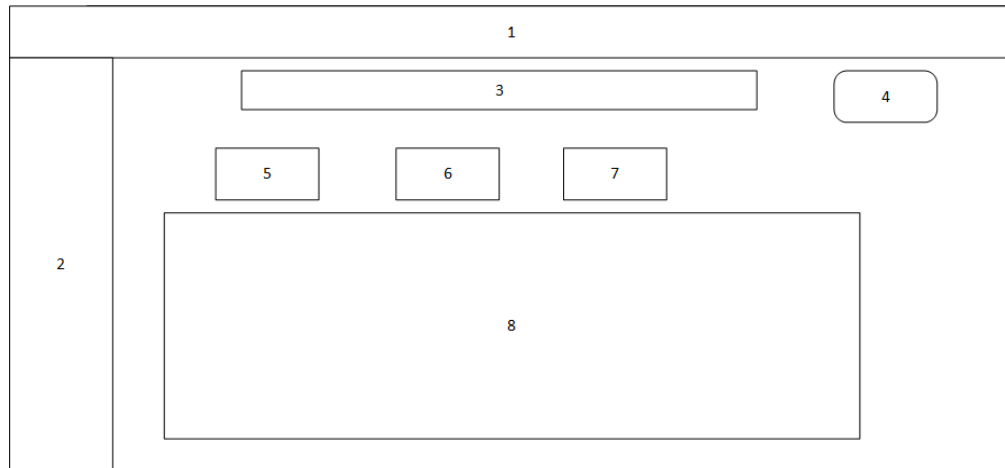
1. Header Autentifikasi User
2. Menu Nagivasi
3. Judul Halaman
4. Form Menambah atau Merubah Data *Smratphone*
5. Tombol Simpan



Gambar 5.18 Rancangan Antar Muka Halaman Menambah dan Merubah *Data Smartphone*

### 5.5.3 Rancangan Antar Muka Halaman Review

Halaman review merupakan halaman untuk admin melihat dan memproses *review* untuk mendapatkan nilai *sentiment review*. Rancangan antar muka halaman *review* ditunjukkan pada Gambar 5.19.



**Gambar 5.19 Rancangan Antar Muka Halaman Review**

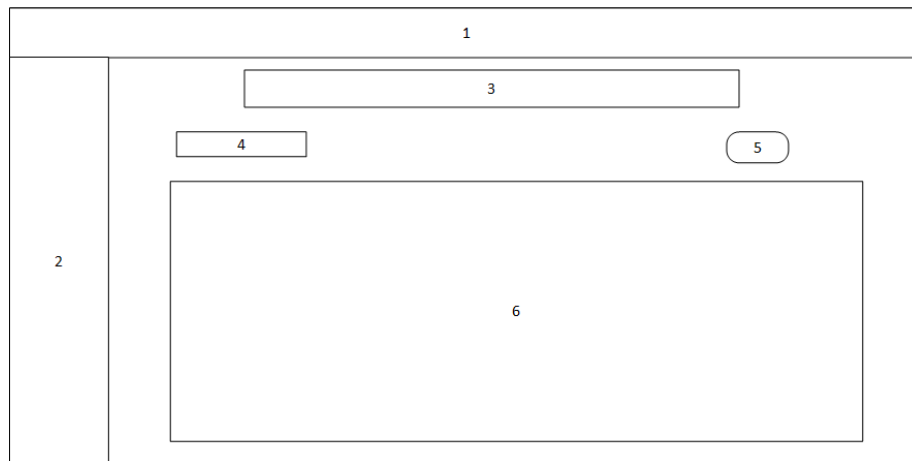
Berikut ini merupakan keterangan untuk menjelaskan komponen-komponen yang terdapat pada Gambar 5.19 :

1. Header Autentifikasi *User*
2. Menu Navigasi
3. Judul Halaman
4. Tombol *Process Review*
5. Jumlah Review
6. Jumlah Review Positif
7. Jumlah Review Negatif
8. Tabel *Review*

#### **5.5.4 Rancangan Antar Muka Halaman Daftar *Smartphone***

Gambar 5.20 menunjukkan rancangan antar muka dari halaman daftar *smartphone*, dimana halaman daftar *smartphone* merupakan halaman yang menampilkan daftar *smartphone*. Keterangan untuk Gambar 5.20 antara lain:

1. Header Autentifikasi *User*
2. Menu Navigasi
3. Judul Halaman
4. Kolom *Search*
5. Tombol Tambah Data *Smartphone*
6. Tabel Daftar *Smartphone*

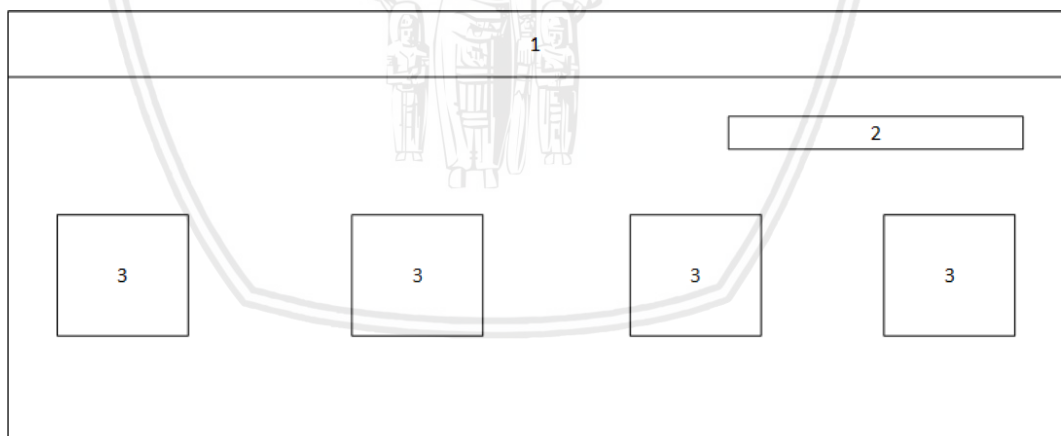


**Gambar 5.20 Rancangan Antar Muka Halaman Daftar *Smartphone***

### 5.5.5 Rancangan Antar Muka Halaman *Home*

Halaman *home* merupakan halaman utama dari Sistem Informasi *Review Smartphone*. Halaman ini menampilkan daftar *smartphone* yang dapat dilihat oleh admin maupun konsumen. Rancangan antar muka halaman *home* digambarkan pada Gambar 5.21, dengan keterangan sebagai berikut :

1. Menu Navigasi
2. Kolom *Search*
3. Form *Smartphone*



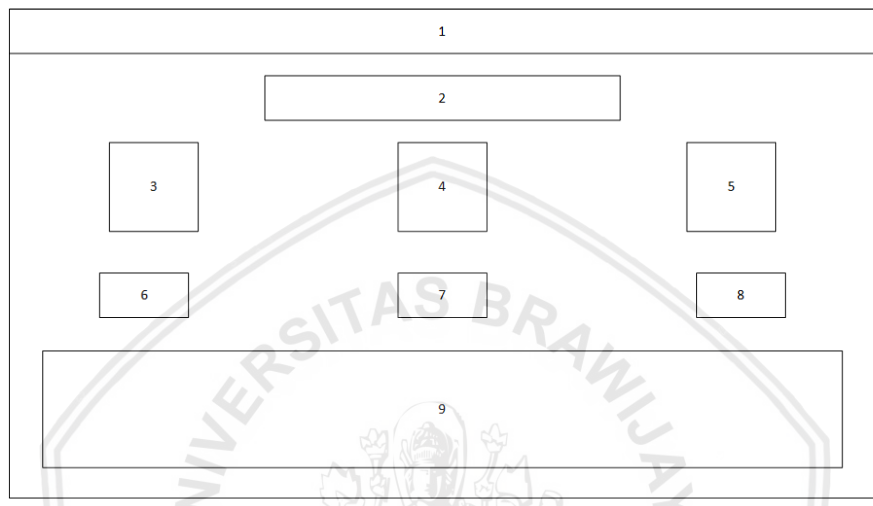
**Gambar 5.21 Rancangan Antar Muka Halaman *Home***

### 5.5.6 Rancangan Antar Muka Halaman *Detail Smartphone*

Rancangan antar muka halaman *detail smartphone* ditunjukkan pada Gambar 5.22. Halaman *detail smartphone* merupakan halaman yang menampilkan *detail* informasi mengenai *smartphone*. Keterangan Gambar 5.22 adalah sebagai berikut:

1. Menu Navigasi
2. Judul Halaman
3. Gambar *Smartphone*

- 4. Tabel *Detail Smartphone*
- 5. Tabel Spesifikasi *Smartphone*
- 6. Jumlah *Review*
- 7. Jumlah *Review* Positif
- 8. Jumlah *Review* Negatif
- 9. Tabel Daftar *Review*

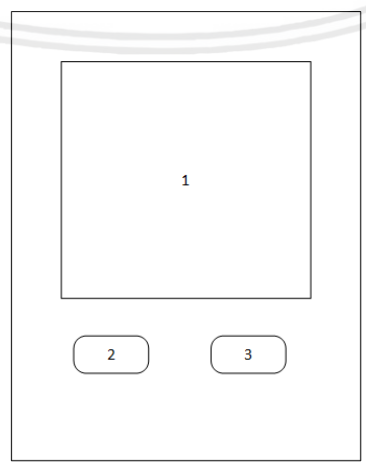


Gambar 5.22 Rancangan Antar Muka Halaman *Detail Smartphone*

### 5.5.7 Rancangan Antar Muka Halaman Tambah *User*

Gambar 2.23 menunjukkan rancangan antar muka halaman tambah *user*, dengan keterangan sebagai berikut :

- 1. *Form* Tambah *User*
- 2. Tombol *Submit*
- 3. Tombol *Cancel*



Gambar 5.23 Rancangan Antar Muka Halaman Tambah *User*



## 5.6 Perancangan Algoritme

Bagian ini menjelaskan algoritme yang akan diterapkan pada tahap implementasi Sistem Informasi *Review Smartphone*. Pada bagian perancangan algoritme ini akan menjelaskan algoritme dari proses mendapatkan *spesifikasi* dan *review* dari situs *web*, menambah data *smartphone*, merubah data *smartphone*, menghapus data *smartphone*, memproses *review*, dan melihat detail *smartphone*.

### 5.6.1 Algoritme *Scraping* Spesifikasi dan *Review*

Proses *scraping* spesifikasi dan *review* adalah proses untuk mendapatkan spesifikasi dan *review* dari situs *web* yang sudah di tentukan. Beberapa langkah proses ini dilakukan oleh sistem, Tabel 5.1 menunjukan perancangan algoritme dari proses *scraping* spesifikasi dan *review*.

**Tabel 5.1 Perancangan Algoritme *Scraping* Spesifikasi dan *Review***

No	Pseudocode
1	Mulai
2	Muat <i>library</i> kelas DOM ke dalam <i>controller</i>
3	Buat <i>object</i> dari kelas DOM
4	Simpan <i>object</i> ke dalam variabel lokal
5	Lakukan pemanggilan fungsi mendapatkan <i>html</i> dari kelas DOM
6	Temukan <i>url</i> halaman <i>review</i> dari situs <i>web</i>
7	Temukan <i>review</i> pada halaman <i>review</i> situs <i>web</i>
8	Simpan <i>review</i> ke dalam variabel lokal
9	Temukan spesifikasi <i>smartphone</i> pada situs <i>web</i>
10	Simpan spesifikasi <i>smartphone</i> ke dalam variabel lokal
11	Selesai

### 5.6.2 Algoritme Menambah Data *Smartphone*

Dalam proses menambah data *smartphone* terdapat beberapa langkah – langkah proses yang dilakukan oleh sistem. Pada proses menambah data *smartphone* juga dijalankan proses *scraping* spesifikasi dan *review*. Perancangan algoritme menambah data *smartphone* di jelaskan pada Tabel 5.2.

**Tabel 5.2 Menambah Data *Smartphone***

No	Pseudocode
1	Mulai
2	Jalankan proses <i>scraping</i> spesifikasi dan <i>review</i>
3	Simpan data ke ke dalam variabel lokal <i>controller</i>
4	Jalankan fungsi tambah data <i>smartphone</i>
5	Simpan data ke dalam basis data melalui kelas
6	Model

No	Pseudocode
7	Arahkan ke halaman detail <i>smartphone</i>
8	Selesai

### 5.6.3 Algoritme Memproses Review

Memproses *review* merupakan proses analisis sentimen pada *review*. Beberapa tahap pada proses ini dilakukan oleh sistem. Perancangan algoritme memproses *review* di jelaskan pada Tabel 5.3.

**Tabel 5.3 Perancangan Algoritme Memproses Review**

No	Pseudocode
1	Mulai
2	Lakukan request <i>get review</i> ke kelas <i>controller</i> untuk mendapatkan <i>review</i> dari kelas Model
3	Simpan <i>review</i> ke dalam variabel lokal
4	Muat <i>library</i> kelas proses <i>sentiment</i> pada kelas <i>controller</i>
5	Buat <i>object</i> dari kelas proses <i>sentiment</i> pada kelas <i>controller</i>
6	Simpan <i>object</i> ke dalam variabel lokal
7	Lakukan pemanggilan fungsi dari kelas proses <i>sentiment</i>
8	Lakukan pemanggilan fungsi SVM dari kelas proses <i>sentiment</i>
9	Ulangi sampai seluruh <i>sentiment review</i>
10	simpan <i>sentiment review</i> ke dalam variabel lokal
11	Selesai perulangan
12	Simpan <i>sentiment review</i> ke dalam basis data melalui kelas Model
13	Selesai

### 5.6.4 Algoritme Merubah Data Smartphone

Terdapat beberapa tahap yang dilakukan oleh sistem dalam proses merubah data *smartphone*. Tabel 5.4 menunjukkan perancangan algoritme dari proses merubah data *smartphone*.

**Tabel 5.4 Perancangan Algoritme Merubah Data Smartphone**

No	Pseudocode
1	Mulai
2	Lakukan request <i>get data</i> ke kelas <i>controller</i> untuk mendapatkan data <i>smartphone</i>



3	Simpan data <i>smartphone</i> ke dalam variabel lokal
4	Simpan perubahan data <i>smartphone</i> ke basis data melalui model
5	Arahkan ke halaman detail <i>smartphone</i>
6	Selesai

### 5.6.5 Algoritme Menghapus Data *Smartphone*

Beberapa langkah proses menghapus data dilakukan oleh sistem. Tabel 5.5 menunjukkan perancangan algoritme menghapus data *smartphone*.

**Tabel 5.5 Perancangan Algoritme Menghapus Data *Smartphone***

No	Pseudocode
1	Mulai
2	Lakukan <i>request get id</i> data ke kelas <i>controller</i> untuk mendapatkan data dari kelas model
3	Simpan id data ke dalam variabel lokal
4	Tampilkan <i>alert</i> konfirmasi apakah data akan dihapus
5	Jika konfirmasi Ok
6	hapus data pada basis data melalui kelas model
7	arahkan ke halaman detail data <i>smartphone</i>
8	Keluar dari kondisi
9	Selesai

### 5.6.6 Algoritme Menampilkan Detail *Smartphone*

Tabel 5.6 menunjukkan perancangan algoritme proses menampilkan detail *smartphone*. Beberapa tahap proses detail *smartphone* dilakukan oleh sistem.

Tabel 5.6 Perancangan Algoritme Menampilkan Detail *Smartphone*

No	Pseudocode
1	Mulai
2	Lakukan <i>request get</i> data <i>smartphone</i> dan <i>review</i> ke kelas <i>controller</i>
3	Lakukan <i>request get</i> jumlah <i>review</i> , jumlah <i>review</i> positif, jumlah <i>review</i> negatif ke kelas <i>controller</i>
4	Simpan data, jumlah <i>review</i> , jumlah <i>review</i> positif, jumlah <i>review</i> negatif ke dalam variabel lokal
5	Tampilkan data dan jumlah <i>review</i> , jumlah <i>review</i> positif, jumlah <i>review</i> negatif pada halaman detail <i>smartphone</i>
6	Selesai



## BAB 6 IMPLEMENTASI

Implementasi merupakan tahap proses direalisasikannya kebutuhan *user*, kebutuhan sistem, dan perancangan kedalam bentuk program dan basis data sehingga terbentuklah aplikasi perangkat lunak konkrit. Bab ini merupakan representasi dari tahapan implementasi pada *Waterfall* model. Pembahasan tahapan implementasi pada bab ini diuraikan menjadi tiga bagian yaitu batasan implementasi, implementasi program, dan implementasi antarmuka.

### 6.1 Batasan Implementasi

Berikut ini merupakan batasan-batasan dalam proses implementasi Sistem Informasi *Review Smartphone* :

1. Sistem Informasi *Review Smartphone* merupakan sistem yang berbasis *website*.
2. Proses pembangunan Sistem Informasi *Review Smartphone* menggunakan bahasa pemrograman PHP dengan menerapkan *framework CodeIgniter*.
3. Manajemen basis data dibangun dengan menggunakan *Relational Database Manajemen System (RDBMS) MySQL*.

### 6.2 Implementasi Program

Bagian ini berisi mengenai implementasi kode-kode program dari Sistem Informasi *Review Smartphone*. Pembahasan pada bagian ini diuraikan menjadi dua yaitu implementasi *website* dan implementasi basis data.

#### 6.2.1 Implementasi *Website*

Pada bagian ini berisi kode-kode implementasi *website* Sistem Informasi *Review Smartphone*. Implementasi pada bagian ini merupakan realisasi dari perancangan yang telah dilakukan yaitu *Sequence diagram* dan perancangan algoritme.

##### 6.2.1.1 Implementasi *Scraping Spesifikasi Dan Review*

Bagian ini merupakan penjelasan mengenai implementasi proses *scraping* spesifikasi dan *review*. Proses ini merupakan proses mendapatkan spesifikasi dan *review* dari situs *web* yang telah ditentukan. Implementasi proses *scraping* spesifikasi dan *review* merupakan realisasi dari perancangan *sequence diagram* *scraping* spesifikasi dan *review* perancangan algoritme *scraping* spesifikasi dan *review*. Pada proses implementasi, fungsi *scraping* spesifikasi dan *review* berada pada *controller* dataController. Tabel 6.1 menunjukkan kode-kode program dari implementasi proses *scraping* spesifikasi dan *review*.

Tabel 6.1 Implementasi *Scraping* Spesifikasi dan *Review*

No	dataController.php
1	public function getLinkPricePrice(){
2	\$data['select_merk'] = \$this->input->post('select_merk');
3	\$data['text_type'] = \$this->input->post('text_type');
4	\$data['select_memory'] = \$this->input->post('select_memory');
5	if(\$data['select_merk'] == 'IPHONE'){
6	\$main_url = "http://id.priceprice.com/search/?keyword="
7	."iphone+" . \$data['text_type'] .
8	"&lid=search_suggest_keyword";
9	\$dom = new DomDocument();
10	@\$dom->loadHTML(file_get_contents(\$main_url));
11	\$dom->preserveWhiteSpace = false;
12	\$urls = \$dom->getElementsByTagName('a');
13	foreach(\$urls as \$value){
14	\$pages = \$value->attributes;
15	\$arr_page = \$pages->
16	getNamedItem('href')->nodeValue;
17	\$keyword = striestr(\$arr_page,
18	\$data['select_merk']."-". "iPhone-"
19	.\$data["text_type"]);
20	if(\$keyword !== false)
21	\$url_review[] = \$keyword;
22	}
23	return \$url_review[0];
24	}else if(\$data['select_merk'] == 'SAMSUNG'){
25	\$main_url =
26	"http://id.priceprice.com/search/?keyword="
27	.\$data['select_merk'] ."GALAXY+" . \$data['text_type']
28	."+"."&lid=search_suggest_keyword";
29	\$dom = new DomDocument();
30	@\$dom->loadHTML(file_get_contents(\$main_url));
31	\$dom->preserveWhiteSpace = false;
32	\$urls = \$dom->getElementsByTagName('a');
33	foreach(\$urls as \$value){
34	\$pages = \$value->attributes;
35	\$arr_page = \$pages->
36	getNamedItem('href')->nodeValue;
37	\$keyword = striestr(\$arr_page,
38	\$data['select_merk']."-galaxy-"
39	.\$data['text_type']);
40	if(\$keyword !== false)
41	\$url_review[] = \$keyword;
42	}
43	return \$url_review[0];
44	}else{
45	\$main_url =
46	"http://id.priceprice.com/search/?keyword="
47	.\$data['select_merk'] ."+" . \$data['text_type']
48	."+"."&lid=search_suggest_keyword";
49	\$dom = new DomDocument();



No	dataController.php
36	<code>@\$dom-&gt;loadHTML(file_get_contents(\$main_url));</code>
37	<code>\$dom-&gt;preserveWhiteSpace = false;</code>
38	<code>\$urls = \$dom-&gt;getElementsByTagName('a');</code>
39	<code>foreach(\$urls as \$value){</code>
40	<code>  \$pages = \$value-&gt;attributes;</code>
41	<code>  \$arr_page = \$pages-&gt;</code>
42	<code>  getNamedItem('href')-&gt;nodeValue;</code>
43	<code>  \$keyword = strstr(\$arr_page,</code>
44	<code>  \$data['select_merk']."-".\$data['text_type']);</code>
45	<code>  if(\$keyword !== false)</code>
46	<code>    \$url_review[] = \$keyword;</code>
47	<code>  }</code>
48	<code>  return \$url_review[0];</code>
49	<code>}</code>
50	<code>public function getReviewPricePrice(){</code>
51	<code>  \$review_url= file_get_html</code>
52	<code>  ("http://id.priceprice.com/" .</code>
53	<code>  \$this-&gt;getLinkPricePrice()."reviews");</code>
54	<code>  foreach (\$review_url-&gt;find("div.itemReview")</code>
55	<code>  as \$keyreview =&gt; \$review) {</code>
56	<code>    \$reviewProfile[] = \$review-&gt;</code>
57	<code>    find("li.nameInfo",0)-&gt;plaintext;</code>
58	<code>    \$reviewIsi[] = \$review-&gt;</code>
59	<code>    find("div.reviewText p", 0)-&gt;plaintext;</code>
60	<code>  }</code>
61	<code>  \$reviews = [</code>
62	<code>    'holder' =&gt; \$reviewProfile,</code>
63	<code>    'review' =&gt; \$reviewIsi</code>
64	<code>  ];</code>
65	<code>  return \$reviews;</code>
66	<code>}</code>
67	<code>function getLinkPricebook(){</code>
68	<code>  \$select_merk = \$this-&gt;input-&gt;post('select_merk');</code>
69	<code>  \$text_type = \$this-&gt;input-&gt;post('text_type');</code>
70	<code>  \$main_url = "";</code>
71	<code>  if(\$select_merk == 'SAMSUNG'){</code>
72	<code>    \$main_url =</code>
73	<code>    "https://www.pricebook.co.id/search?keyword="</code>
74	<code>    .\$select_merk."+GALAXY+"."+"</code>
75	<code>    .\$text_type."&amp;product=all";</code>
76	<code>  }else{</code>
77	<code>    \$main_url =</code>
78	<code>    "https://www.pricebook.co.id/search?keyword="</code>
79	<code>    .\$select_merk+".".\$text_type."&amp;product=all";</code>
80	<code>  }</code>
81	<code>  \$dom = new DomDocument();</code>
82	<code>  @\$dom-&gt;loadHTML(file_get_contents(\$main_url));</code>
83	<code>  \$dom-&gt;preserveWhiteSpace = false;</code>

No	dataController.php
73	<code>\$page = \$dom-&gt;getElementsByTagName('a');</code>
74	<code>foreach(\$page as \$value){</code>
75	<code>\$links = \$value-&gt;getAttribute('href');</code>
76	<code>\$str = strstr(\$links,\$select_merk."-".\$text_type);</code>
77	<code>if(\$str !== FALSE)</code>
78	<code>    \$arr_url[] = \$str;</code>
79	<code>}</code>
80	<code>return \$arr_url[0];</code>
81	<code>}</code>
82	<code>public function getReviewPb(){</code>
83	<code>    \$dom = new DomDocument();</code>
84	<code>    \$url = \$this-&gt;getLinkPricebook();</code>
85	<code>    \$pagesurl =</code>
	<code>        "https://www.pricebook.co.id/".\$url."/review";</code>
86	<code>    \$html = file_get_html(\$pagesurl);</code>
87	<code>    \$reviews=[];</code>
88	<code>        foreach (\$html -&gt;</code>
	<code>            find("div.caption") as \$review) {</code>
89	<code>            \$reviewprofile[] = \$review-&gt;</code>
	<code>                find("span.userprof-name",0)-&gt;plaintext;</code>
90	<code>            \$reviewisi[] = substr(\$review-&gt;</code>
	<code>                find("div.comment",0)-&gt;plaintext,0, 200);</code>
91	<code>        }</code>
92	<code>    \$reviewpb =[</code>
93	<code>        'holder' =&gt; \$reviewprofile,</code>
94	<code>        'review' =&gt; \$reviewisi,</code>
95	<code>    ];</code>
96	<code>    return \$reviewpb;</code>
97	<code>}</code>
98	<code>public function getLinkIprice(){</code>
99	<code>    \$merk = \$this-&gt;input-&gt;post('select_merk');</code>
100	<code>    \$type = \$this-&gt;input-&gt;post('text_type');</code>
101	<code>    \$main_url =</code>
	<code>        "https://iprice.co.id/search/?term="</code>
	<code>        .\$merk."+".\$type;</code>
102	<code>    \$dom = new DomDocument();</code>
103	<code>    @\$dom-&gt;loadHTML(</code>
	<code>        file_get_contents(\$main_url));</code>
104	<code>    \$dom-&gt;preserveWhiteSpace = false;</code>
105	<code>    \$urls = \$dom-&gt;getElementsByTagName('a');</code>
106	<code>    \$keyword = "";</code>
107	<code>    \$arr_url = array();</code>
108	<code>    foreach(\$urls as \$value){</code>
109	<code>        \$pages = \$value-&gt;attributes;</code>
110	<code>        \$arr_page =</code>
	<code>            \$value-&gt;getAttribute('href');</code>
111	<code>        if(\$merk == 'IPHONE'){</code>

No	dataController.php
112	<code>\$keyword = striestr(\$arr_page, "Ahalle-".\$merk."-".\$stype);</code>
113	<code>}elseif(\$merk=='samsung'){</code>
114	<code>    \$keyword = striestr(\$arr_page,\$merk."-galaxy-".\$stype);</code>
115	<code>}else{</code>
116	<code>    \$keyword = striestr(\$arr_page,\$merk."-".\$stype);</code>
117	<code>}</code>
118	<code>if(\$keyword !== false)</code>
119	<code>    \$arr_url[] = \$keyword;</code>
120	<code>}</code>
121	<code>return \$arr_url[0];</code>
122	<code>}</code>
123	<code>public function getReviewIP(){</code>
124	<code>    \$dom = new DomDocument();</code>
125	<code>    \$url = \$this-&gt;getLinkIprice();</code>
126	<code>    \$urls = "https://iprice.co.id/harga/".\$url;</code>
127	<code>    \$review_url = file_get_html(\$urls);</code>
128	<code>    \$pages = \$review_url-&gt;find("div[class=review dt w-100]");</code>
129	<code>    \$countdata = count(\$pages);</code>
130	<code>    if(\$countdata !== 0){</code>
131	<code>        foreach(\$pages as \$review){</code>
132	<code>            \$reviewProfile[] = \$review-&gt;find("div.[class=rating-header</code>
133	<code>            \$reviewIsi [] = substr(\$review-&gt;find("div.review-text",0)-</code>
134	<code>            &gt;plaintext,0,250);</code>
135	<code>        }  </code>
136	<code>        \$reviews = [</code>
137	<code>            'holder' =&gt; \$reviewProfile,</code>
138	<code>            'review' =&gt; \$reviewIsi,</code>
139	<code>        ];</code>
140	<code>    }else{</code>
141	<code>        \$reviews = [</code>
142	<code>            'holder' =&gt; "",</code>
143	<code>            'review' =&gt; "",</code>
144	<code>        ];</code>
145	<code>    }  </code>
146	<code>    return \$reviews;</code>
147	<code>}</code>
148	<code>public function getSpec(){</code>
149	<code>    \$dom = new DomDocument();</code>
150	<code>    \$url = \$this-&gt;getLinkPricebook();</code>
151	<code>    \$pagesurl = "https://www.pricebook.co.id/".\$url."/review";</code>
152	<code>    \$html = file_get_html(\$pagesurl);</code>
153	<code>    \$spec = \$html-&gt;find('div.value-spec',0)-&gt;</code>
154	<code>    innertext;</code>

No	dataController.php
153	return \$spec;
154	}

Pada Tabel 6.1 merupakan kode program dari proses *scraping* spesifikasi dan *review*. Kode tersebut berada pada kelas *controller* yaitu *dataController*. Fungsi-fungsi pada proses *scraping* ini menggunakan *library* kelas DOM, *library* DOM digunakan untuk membantu dalam mengambil data dari ketiga situs web yang sudah ditentukan yaitu *priceprice.com*, *pricebook.co.id*, dan *iprice.co.id*. Kode program baris 1-48 merupakan fungsi untuk mendapatkan *url* halaman *smartphone* dari situs *priceprice.com*. Dimana kode pada baris ke 5,19, dan 33 merupakan seleksi untuk menentukan nilai dari variabel *main\_url* berdasarkan merek dari *smartphone*. Kode pada baris ke 48-60 merupakan fungsi untuk mendapatkan *review* dari situs *priceprice.com*. Pada baris kode 61-79 adalah fungsi untuk mendapatkan *url* halaman *smartphone* dari situs *pricebook.co.id*. Kode program pada baris ke 62-97 merupakan fungsi untuk mendapatkan *review* dari situs *pricebook.co.id*. Kode program fungsi untuk mendapatkan *url* data *smartphone* dari situs *lprice.com* berada pada baris 98 – 122, sedangkan kode program untuk mendapatkan *review* dari situs *lprice* berada pada baris ke 123-146. Baris ke 130 merupakan seleksi kondisi dimana jika jumlah *review* pada situs *iprice* tidak bernilai nol maka nilai dari variabel *reviews* merupakan *review* yang berada pada situs web *iprice*. Kode program pada baris ke 147-154 merupakan fungsi untuk mendapatkan spesifikasi *smartphone*, dengan nilai dari variabel *url* di dapatkan dari nilai kembalian fungsi *getLinkPricebook*. Spesifikasi didapatkan dari situs web *pricebook.com*. Kode program baris ke 7, 21, 30, 70, 83, 102, 124, 148 merupakan pembuatan *object* dari *library* kelas DOM.

### 6.2.1.2 Implementasi Menambah Data Smartphone

Bagian ini menjelaskan mengenai kode-kode program dalam implementasi proses menambah data. Proses implementasi menambah data merupakan realisasi dari perancangan *use case* diagram dan perancangan algoritme menambah data *smartphone*. Fungsi menambah data diimplementasikan pada kelas *controller* *dataController* dan kelas model *datasmartphone*. Tabel 6.2 merupakan kode-kode program fungsi menambah data pada kelas *dataController*.

**Tabel 6.2 Implementasi Controller DataController Menambah Data Smartphone**

No	dataController.php
1	public function addData(){
2	\$merk = \$this->input->post('select_merk');
3	\$tipe = \$this->input->post('text_type');
4	\$memori = \$this->input->post('select_memory');
5	\$harga = \$this->input->post('text_harga');
6	\$gambar = \$this->image_upload();
7	\$Spesifikasi = \$this->input->post('text_spesifikasi');
8	\$revIp = \$this->getReviewIP();
9	\$revPrice = \$this->getReviewPricePrice();
10	\$revPb = \$this->getReviewPb();



No	dataController.php
11	if(\$revIp != array()){
12	\$holder = array_merge(\$revPrice['holder'], \$revPb['holder']);
13	\$review = array_merge(\$revPrice['review'], \$revPb['review']);
14	}
15	else{
16	\$holder = array_merge(\$revPrice['holder'],
17	\$revPb['holder'], \$revIp['holder']);
18	\$review = array_merge(\$revPrice['review'], \$revPb['review'],
19	\$revIp['review']);
20	}
21	\$dataReview['HOLDER'] = \$holder;
22	\$dataReview['REVIEW'] = \$review;
23	\$dataSmartphone = array(
24	'IDSMARTPHONE'=>' ',
25	'MERK'=>\$merk,
26	'TYPE'=>\$tipe,
27	'SPECIFICATION'=>\$spesifikasi,
28	'PRICE'=>\$harga,
29	'PICTURE'=>\$gambar,
30	);
31	\$insertData = \$this->datasmartphone-
32	->dataAdd(\$dataSmartphone, \$dataReview);
33	}

Tabel 6.2 menjelaskan fungsi `addData` yang merupakan fungsi untuk menambah data *smartphone* pada kelas `dataController`. Pada kode program baris ke 8, 9, 10 merupakan proses pengambilan data *review* dari situs yang sudah ditentukan. Baris ke 11 merupakan seleksi kondisi jika nilai dari *review* yang didapatkan dari situs *iprice* bukan sebuah array maka *review* hanya didapatkan dari dua situs yaitu *pricebook* dan *priceprice*. `array_merge` digunakan untuk menggabungkan *review* yang didapat dari ketiga situs dan menyimpannya menjadi ke dalam satu variabel bernilai array. Kode program ke 187 merupakan proses pemanggilan fungsi `dataAdd` pada kelas model `datasmartphone` untuk menyimpan data ke dalam basis data. Tabel 6.3 dibawah ini menjelaskan kode program dari fungsi `dataAdd` pada kelas model `datasmartphone`.

**Tabel 6.3 Implementasi Model *DataSmartphone* Menambah Data *Smartphone***

No	datasmartphone.php
1	public function dataAdd(\$datasmartphone, \$dataReview) {
2	\$this->db->insert('smartphone', \$datasmartphone);
3	\$holder = \$dataReview['HOLDER'];
4	\$review = \$dataReview['REVIEW'];
5	for(\$i=0; \$i<count(\$holder); \$i++) {
6	\$last_id = \$this->db->insert_id();
7	\$reviewData[] = array(
8	'IDSMARTPHONE' => \$last_id,
9	'HOLDER' => \$holder[\$i],

No	datasmartphone.php
10	REVIEW' => \$review[\$i],
11	);
12	}
13	\$this->db->insert_batch('review', \$reviewData);
14	}

Tabel 6.3 menjelaskan kode program fungsi `addData` yang merupakan fungsi untuk menambah data *smartphone* dan menyimpannya ke dalam basis data. Kode baris 2 merupakan proses menyimpan data ke dalam basis data tabel *smartphone*. Pada baris ke 5-12 merupakan perulangan untuk membuat array dari *review* dengan `IDSMARTPHONE` yang merupakan *foreign keys* pada basis data tabel *review* nilainya di dapat dari id yang terakhir di masukkan ke dalam tabel *smartphone*. Pada baris 13 merupakan proses menyimpan *review* ke dalam tabel *review*, `insert_batch` digunakan agar proses memasukkan banyak data *review* dapat dilakukan secara sekaligus.

### 6.2.1.3 Implementasi Memproses Review

Pada bagian ini menjelaskan kode-kode program dari memproses *review*. Proses ini merupakan proses pengidentifikasian *review* dan pengelompokkan *review* berdasarkan sentimennya. Kode-kode program ini merupakan realisasi dari perancangan *sequence* diagram dan perancangan algoritme memproses *review*. Pada implementasinya, fungsi memproses *review* berada pada kelas *controller* `dataController` dan kelas model `SentimentModel`. Tabel 6.4 merupakan kode-kode program dari memproses *review* pada kelas *controller* `dataController`.

**Tabel 6.4 Implementasi Controller DataController Memproses Review**

No	dataController.php
1	public function getSentimentReview(){
2	\$SentimentProcess = new SentimentProcess();
3	\$idsmartphone = \$this->uri->segment(3);
4	\$source_data = \$this->sentimentModel->getReview(\$idsmartphone);
5	\$data_review = \$SentimentProcess->document(\$source_data);
6	\$reviewID = \$data_review[0];
7	\$review = \$data_review[1];
8	\$labeling = \$SentimentProcess->labeling(\$review);
9	\$SVM = \$SentimentProcess->svm(\$labeling);
10	foreach(\$SVM as \$value_SVM){
11	foreach(\$value_SVM as \$value_sentiment){
12	\$sentiment[] = \$value_sentiment;
13	}
14	}
15	for(\$i=0; \$i < count(\$sentiment); \$i++){
16	\$data[] = ['IDREVIEW' => \$reviewID[\$i],
17	'SENTIMENT' => \$sentiment[\$i]];





No	dataController.php
18	}
19	\$sql = \$this->sentimentModel->updateReviewSentiment(\$data);
20	\$data['semua'] = \$this->sentimentModel->detailReview(\$idsmartphone);
21	\$data['merktype'] = \$this->sentimentModel->getMerkType(\$idsmartphone);
22	\$data['jmlreview'] = \$this->sentimentModel->getJumlahReview(\$idsmartphone);
23	\$data['jmlpositif'] = \$this->sentimentModel->getJumlahPositif(\$idsmartphone);
24	\$data['jmlnegatif'] = \$this->sentimentModel->getJumlahNegatif(\$idsmartphone);
25	\$header['css'][] = base_url('assets/vendor/bootstrap-4.1/bootstrap.min.css');
26	\$header['css'][] = base_url('assets/css/theme.css');
27	\$this->load->view('admin/dashboard_header');
28	\$this->load->view('admin/ReviewPage', \$data);
29	\$this->load->view('admin/dashboard_footer');
30	}

Tabel 6.4 menjelaskan kode-kode program dari implementasi proses memproses *review* pada kelas *controller* dataController. Pada kelas dataController fungsi *getSentimentReview* merupakan fungsi untuk memproses *review*. Pada kelas ini dibutuhkan *library* kelas *SentimentProses* untuk mempermudah proses analisis sentimen dari *review*. Pada baris ke 2 merupakan pembuatan *object* dari kelas *library* *SentimentProses*. Baris ke 4 menjelaskan proses untuk mendapatkan data *review* dari basis data dengan cara memanggil fungsi *getReview* dari kelas model *SentimentModel*. Kode baris ke 5 merupakan proses pemrosesan teks sebagai proses awal dari analisis sentimen, proses ini dilakukan dengan memanggil fungsi *document* dari kelas *library* *SentimentProses*. Pada baris ke 8 memanggil fungsi *labeling* dari kelas *library* *SentimentProses*, kode program ini merupakan salah satu tahap dari analisis sentimen yaitu proses pemberian label pada *review*. Tahap selanjutnya dari proses analisis sentimen adalah proses pengkelompokan *review* ke dalam nilai positif atau negatif, pada tabel 6.4 ini proses pengelompokan *review* di tunjukkan kode program pada baris ke 9. Proses ini memanggil fungsi *svm* yang merupakan fungsi pengklasifikasian *review* dengan menggunakan metode klasifikasi *Support vector machine*. Kode program pada baris ke 19 merupakan proses menyimpan hasil proses *review* ke dalam basis data dengan memanggil fungsi *updateReviewSentiment* dari kelas model *SentimentModel*. Tabel 6.5 dibawah ini menjelaskan kode program dari kelas model *SentimentModel*.

**Tabel 6.5 Implementasi Model SentimentModel Memproses Review**

No	sentimentModel.php
1	public function updateReviewSentiment(\$data) {
2	\$this->db->update_batch('REVIEW',\$data, 'IDREVIEW');
3	}

Pada tabel 6.5 menjelaskan fungsi *updateReviewSentiment* yang merupakan fungsi untuk proses menyimpan nilai sentimen *review* kedalam basis data. Kode program baris ke 2 menunjukkan proses menyimpan nilai sentimen *review* ke dalam tabel *review*. Fungsi *update\_batch* digunakan karena nilai awal sentimen pada basis data adalah *null*, sehingga fungsi *update* digunakan untuk mengubah nilai *null* menjadi nilai dari hasil proses pengklasifikasian sebelumnya.

#### 6.2.1.4 Implementasi Merubah Data Smartphone

Pada bagian ini menjelaskan kode-kode program implementasi proses merubah data *smartphone*. Fungsi untuk merubah data berada pada *controller* dataController dan kelas model *datasmartphone*. Tabel 6.6 merupakan kode program dari merubah data.

**Tabel 6.6 Implementasi Controller DataController Merubah Data Smartphone**

No	dataController.php
1	public function dataEdit() {
2	\$idsmartphone= \$this->uri->segment(3);
3	\$data['MERK'] = \$this->input->post('select_merk');
4	\$data['TYPE'] = \$this->input->post('text_type');
5	\$data['MEMORY'] = \$this->input->post('select_memory');
6	\$data['SPECIFICATION'] = \$this->input->post('text_spesifikasi');
7	\$data['PRICE'] = floatval(\$this->input->post('text_harga'));
8	\$data['PICTURE'] = \$this->image_upload();
9	\$this->datasmartphone->dataEdit(\$idsmartphone,\$data);
10	redirect(base_url('dataController'));
11	}

Tabel 6.6 merupakan kode-kode program dari implementasi proses merubah data dari kelas dataController. Pada kode baris ke 2 menjelaskan bahwa nilai dari variabel *idsmartphone* sama dengan nilai *url* pada segment ke 3. Untuk melakukan perubahan data pada basis data, fungsi *dataEdit* memanggil fungsi *dataEdit* dari kelas model *datasmartphone*. Kode program dari proses merubah data *smartphone* pada kelas *datasmartphone* ditunjukkan pada Tabel 6.7.

**Tabel 6.7 Implementasi Model DataSmartphone Merubah Data Smartphone**

No	datasmartphone.php
1	public function dataEdit(\$idsmartphone,\$datasmartphone) {
2	\$this->db->where('IDSMARTPHONE',\$idsmartphone);
3	\$this->db->update('SMARTPHONE', \$datasmartphone);
4	}



Pada Tabel 6.7 merupakan fungsi untuk melakukan perubahan data pada basis data. Fungsi `dataEdit` melakukan perubahan data pada tabel *smartphone*.

### 6.2.1.5 Implementasi Menghapus Data Smartphone

Bagian menunjukkan kode-kode program dari implementasi proses menghapus data *smartphone*. Fungsi proses ini pada kelas *controller* `dataController` ditunjukkan pada Tabel 6.8.

**Tabel 6.8 Implementasi Controller DataController Menghapus Data Smartphone**

No	<code>dataController.php</code>
1	<code>public function deleteData(){</code>
2	<code>    \$id = \$this-&gt;uri-&gt;segment(3);</code>
3	<code>    \$sql = \$this-&gt;datasmartphone-&gt;dataDelete(\$id);</code>
4	<code>    redirect(base_url('DataController/'));</code>
5	<code>}</code>

Pada Tabel 6.8 merupakan kode program dari fungsi `deleteData`, dimana fungsi tersebut adalah fungsi untuk proses menghapus data *smartphone*. Pada baris ke 3 merupakan proses pemanggilan fungsi `dataDelete` dari kelas model *datasmartphone* sebagai jembatan untuk menghapus data dari basis data. Kode program fungsi `dataDelete` pada kelas *datasmartphone* di jelaskan pada tabel 6.9 dibawah ini.

**Tabel 6.9 Implentasi Model DataSmartphone Menghapus Data Smartphone**

No	<code>datasmartphone.php</code>
1	<code>public function deleteData(){</code>
2	<code>    \$id = \$this-&gt;uri-&gt;segment(3);</code>
3	<code>    \$sql = \$this-&gt;datasmartphone-&gt;dataDelete(\$id);</code>
4	<code>    redirect(base_url('DataController/'));</code>
5	<code>}</code>

### 6.2.1.6 Implementasi Menampilkan Detail Smartphone

Pada bagian ini menjelaskan mengenai kode program implementasi untuk proses menampilkan *detail smartphone*. Proses ini adalah proses sistem untuk menampilkan keluruhan informasi dari sebuah *smartphone*. Implementasi proses ini mengacu pada perancangan diagram *sequence* dan perancangan algoritme menampilkan *detail smartphone*. Tabel 6.10 merupakan kode program dari kelas *controller* `home` untuk proses menampilkan detail *smartphone*.

**Tabel 6.10 Implementasi Controller Home Menampilkan Detail Smartphone**

No	<code>home</code>
1	<code>public function detailSmartphone(){</code>
2	<code>    \$idsmartphone = \$this-&gt;uri-&gt;segment(3);</code>
3	<code>    \$this-&gt;datasmartphone-&gt;updateViews(\$idsmartphone);</code>

No	home
4	<code>\$data['detailsmartphone'] = \$this-&gt;datasmartphone-&gt;getDetailSmartphone(\$idsmartphone);</code>
5	<code>\$data['semua'] = \$this-&gt;sentimentModel-&gt;detailReview(\$idsmartphone);</code>
6	<code>\$data['merktype'] = \$this-&gt;sentimentModel-&gt;getMerkType(\$idsmartphone);</code>
7	<code>\$data['jmlreview'] = \$this-&gt;sentimentModel-&gt;getJumlahReview(\$idsmartphone);</code>
8	<code>\$data['jmlpositif'] = \$this-&gt;sentimentModel-&gt;getJumlahPositif(\$idsmartphone);</code>
9	<code>\$data['jmlnegatif'] = \$this-&gt;sentimentModel-&gt;getJumlahNegatif(\$idsmartphone);</code>
10	<code>\$this-&gt;load-&gt;view('home_header');</code>
11	<code>\$this-&gt;load-&gt;view('home_detailSmartphone', \$data);</code>
12	<code>\$this-&gt;load-&gt;view('home_footer');</code>
13	<code>}</code>

Tabel 6.10 menjelaskan fungsi `getDetailSmartphone` sebagai fungsi untuk menampilkan detail *smartphone*. Dalam fungsi ini untuk mendapatkan data dari basis data, fungsi ini memanggil beberapa fungsi sekaligus dari kelas model *datasmartphone* dan kelas model *sentimentmodel*. Pada baris ke 3 merupakan proses untuk melakukan *update* jumlah views dari sebuah *smartphone*. Baris ke 7 – 9 merupakan proses untuk mendapatkan jumlah review, jumlah review positif, jumlah review negatif secara berurutan dari kelas model *sentiment model*. Semua data yang di dapatkan dari basis data ditampilkan pada kelas *view home\_detailSmartphone* seperti pada kode program baris ke 11.

Kode-kode program dari setiap fungsi yang dipanggil oleh fungsi `getDetailSmartphone` kelas model *datasmartphone* dan kelas model *sentimentmodel* di jabarkan pada tabel-tabel di bawah ini.

**Tabel 6.11 Implementasi Model DataSmartphone Fungsi GetDetailSmartphone**

No	<code>datasmartphone.php</code>
1	<code>public function getDetailSmartphone(\$idsmartphone) {</code>
2	<code>    \$value = \$this-&gt;db-&gt;query("SELECT S. IDSMARTPHONE, S.MERK, S.TYPE, S. SPECIFICATION, S.PRICE, S.PICTURE, S.MEMORY, S.VIEWS FROM SMARTPHONE S where idsmartphone='".\$idsmartphone);</code>
3	<code>    \$result = \$value-&gt;result_array();</code>
4	<code>    return \$result;</code>
5	<code>}</code>



**Tabel 6.12 Implementasi Model DataSmartphone Fungsi updateViews**

No	dataSmartphone.php
1	function updateViews(\$idsmartphone) {
2	\$this->db->where('IDSMARTPHONE', \$idsmartphone);
3	\$this->db->select('VIEWS');
4	\$count = \$this->db->get('smartphone')->row();
5	\$this->db->where('IDSMARTPHONE', \$idsmartphone);
6	\$this->db->set('VIEWS', (\$count->VIEWS+1));
7	\$this->db->update('smartphone');
8	}

**Tabel 6.13 Implementasi Model SentimentModel**

No	sentimentModel.php
1	public function detailReview(\$idsmartphone) {
2	\$value = \$this->db->query("SELECT S.MERK, S.TYPE, S.SPECIFICATION, R.HOLDER, R.IDREVIEW, R.REVIEW, R.SENTIMENT FROM SMARTPHONE S RIGHT JOIN REVIEW R ON S.IDSMARTPHONE = R.IDSMARTPHONE WHERE R.IDSMARTPHONE=".\$idsmartphone );
3	\$result = \$value->result_array();
4	return \$result;
5	}
6	
7	public function getJumlahReview(\$idsmartphone) {
8	\$value = \$this->db->query("SELECT SENTIMENT FROM REVIEW WHERE IDSMARTPHONE = ".\$idsmartphone);
9	return \$value->num_rows();
10	}
11	
12	public function getJumlahPositif(\$idsmartphone) {
13	\$this->SENTIMENT = 'positif';
14	\$value = \$this->db->query("SELECT SENTIMENT FROM REVIEW WHERE SENTIMENT= '\$this->SENTIMENT' AND IDSMARTPHONE = ".\$idsmartphone);
15	return \$value->num_rows();
16	}
17	
18	public function getJumlahNegatif(\$idsmartphone) {
19	\$this->SENTIMENT = 'negatif';
20	\$value = \$this->db->query("SELECT SENTIMENT FROM REVIEW WHERE SENTIMENT= '\$this->SENTIMENT' AND IDSMARTPHONE = ".\$idsmartphone);
21	return \$value->num_rows();
22	}

### 6.2.2 Implementasi Data Definition Language

Bagian ini merupakan bagian dimana skema basis data yang telah di buat sebelumnya pada perancangan, diimplementasikan menjadi sebuah basis data. Proses pengeimplementasian skema basis data menjadi sebuah basis data menggunakan perintah SQL yaitu *Data Definition Language* (DDL). Pada implementasi DDL ini pada tabel-tabel dibawah ini dijelaskan mengenai *query*



yang digunakan untuk membuat tabel pada basis data untuk Sistem Informasi Review *Smartphone*. Tabel 6.14 menunjukkan implementasi DDL untuk tabel *smartphone* yang digunakan untuk menyimpan data *smartphone*.

**Tabel 6.14 Implementasi DDL Tabel *Smartphone***

NO	<i>SMARTPHONE</i>
1	CREATE TABLE `SMARTPHONE` ( `IDSMARTPHONE` INT(9)
2	NOT NULL, `MERK` CHAR(9) NOT NULL, `TYPE` VARCHAR(50)
3	DEFAULT NULL, `MEMORY` INT(3) DEFAULT NULL,
4	`SPECIFICATION` LONGTEXT, `PRICE` DECIMAL(10,0) NULL,
5	`PICTURE` TEXT, `VIEWS` INT(11));

Selanjutnya Tabel 6.15 merupakan *query* DDL untuk implementasi tabel *review* yang digunakan untuk menyimpan data *review*.

**Tabel 6.15 Implementasi DDL Tabel *Review***

No	<i>review</i>
1	CREATE TABLE `REVIEW` ( `IDSMARTPHONE` INT(9) NOT NULL FOREIGN KEY
2	REFERENCES <i>SMARTPHONE</i> (IDSMARTPHONE),
3	`IDREVIEW` INT(11) NOT NULL,
4	`IDSITUS` CHAR(1) NOT NULL FOREIGN KEY REFERENCES
5	<i>SITUS_WEB</i> (IDSITUS),
6	`HOLDER` VARCHAR(100) NOT NULL,
7	`REVIEW` LONGTEXT,
8	`SENTIMENT` VARCHAR(50));

Tabel 6.16 dibawah ini merupakan *query* DDL dari implementasi tabel *petugas* yang digunakan untuk menyimpan data pengguna sistem seperti *admin*.

**Tabel 6.16 Implementasi DDL Tabel *Review***

No	<i>petugas</i>
1	CREATE TABLE `PETUGAS` ( `IDPETUGAS` INT(9) NOT NULL,
2	`FULLNAME` VARCHAR(50) NOT NULL,
3	`USERNAME` VARCHAR(35) NOT NULL,
4	`PASSWORD` VARCHAR(255));

Tabel 6.17 dibawah ini merupakan *query* DDL dari implementasi tabel *situs\_web* yang digunakan untuk menyimpan data alamat situs.

**Tabel 6.17 Implementasi DDL Tabel *Situs\_Web***

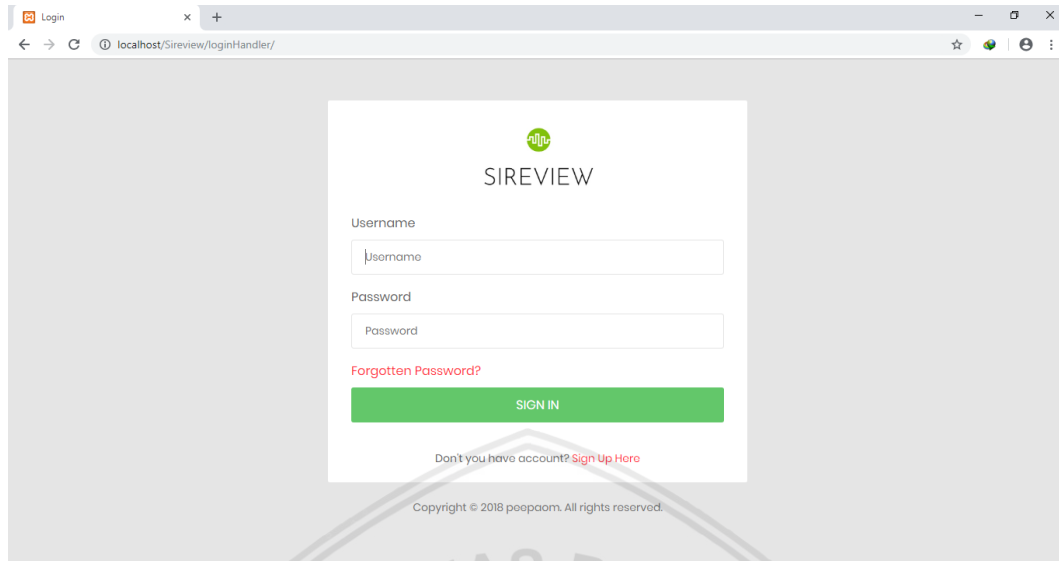
No	<i>petugas</i>
1	CREATE TABLE `PETUGAS` ( `IDSITUS` CHAR(1) NOT NULL PRIMARY KEY,
2	`SITUS` VARCHAR(35) NOT NULL));

### 6.2.3 Implementasi Antarmuka

Bagian ini merupakan bagian dimana rancangan antar muka yang sudah di rancang sebelumnya diimplementasikan menjadi antarmuka dari Sistem Informasi *Review Smartphone*.



### 6.2.3.1 Implementasi Antarmuka Halaman Login

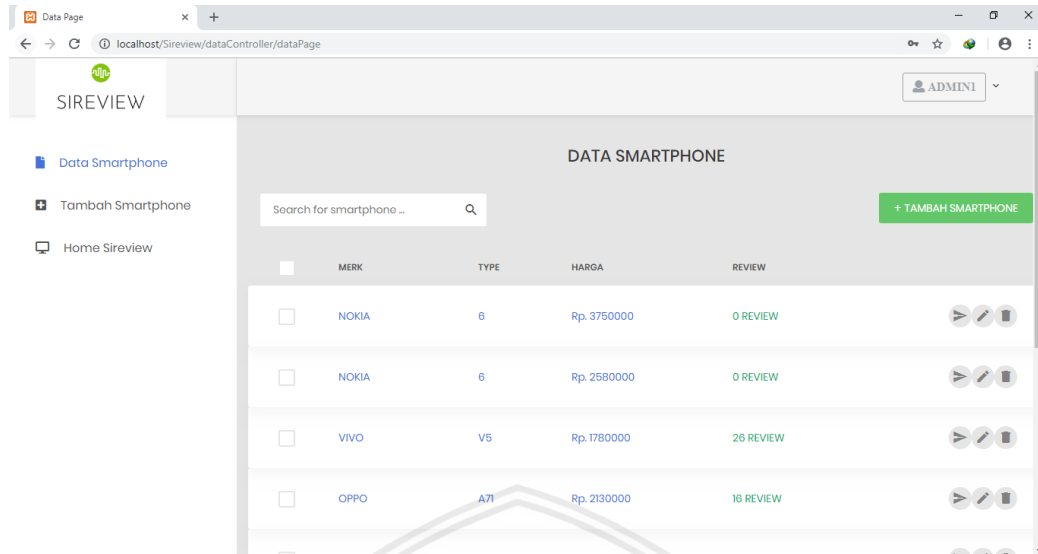


**Gambar 6.1 Implementasi Antarmuka Halaman Login**

Gambar 6.1 di atas merupakan antarmuka halaman login dari Sistem Informasi *Review Smartphone*. Pada halaman *login* terdapat *form login* yang terdiri dari kolom untuk memasukkan *username*, kolom untuk memasukkan *password*, tombol *sign in* dan sebuah *link* untuk *sign up*.

### 6.2.3.2 Implementasi Antarmuka Halaman DataSmartphonePage

Halaman *datasmartphonepage* adalah halaman yang menampilkan daftar *smartphone* yang terdapat pada basis data. Halaman ini juga merupakan halaman *admin* dapat melakukan proses pengolahan data seperti memproses *review*, mengubah data dan/atau menghapus data. Gambar 6.2 merupakan antarmuka dari halaman *datasmartphonepage*.

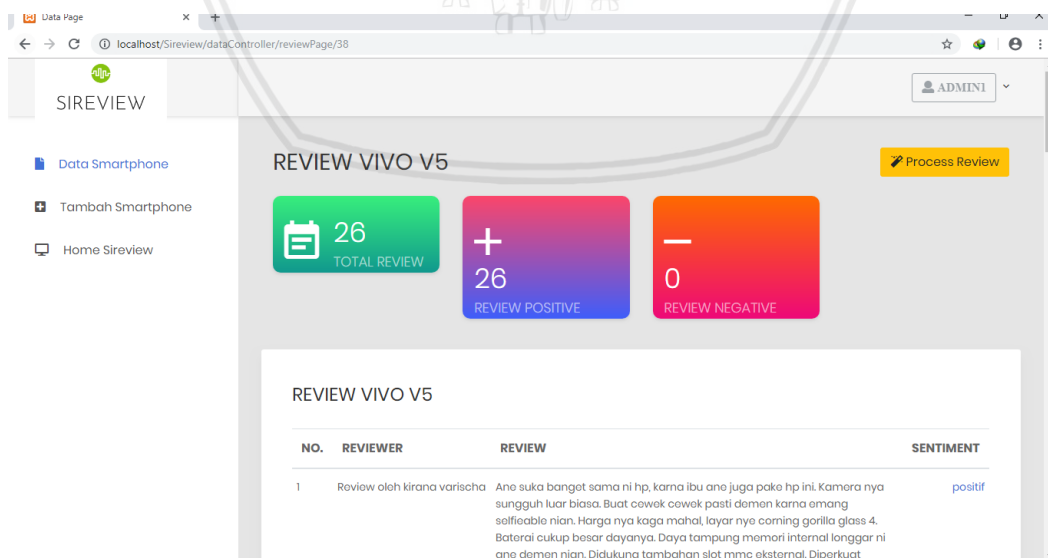


**Gambar 6.2 Implementasi Antarmuka Halaman DataSmartphonePage**

Pada gambar 6.2 menjelaskan pada halaman *datasmartphonepage* menampilkan tabel daftar data *smartphone* yang tersimpan dalam basis data sistem. Pada halaman ini juga terdapat tombol tambah *smartphone*, tombol untuk merubah data, tombol untuk memproses data, tombol untuk menghapus data, dan kolom pencarian untuk mencari data. Pada navigasi terdapat pilihan menuju halaman data *smartphone*, tambah *smartphone* dan halaman *home* dari Sistem Informasi *Review*.

### 6.2.3.3 Implementasi Antarmuka Halaman ReviewPage

Halaman *reviewpage* merupakan halaman yang menampilkan *review* beserta nilai sentimennya. Pada halaman ini admin dapat melakukan proses memproses *review*. Antarmuka halaman *reviewpage* ditunjukkan pada Gambar 6.3.



**Gambar 6.3 Implementasi Antarmuka Halaman ReviewPage**

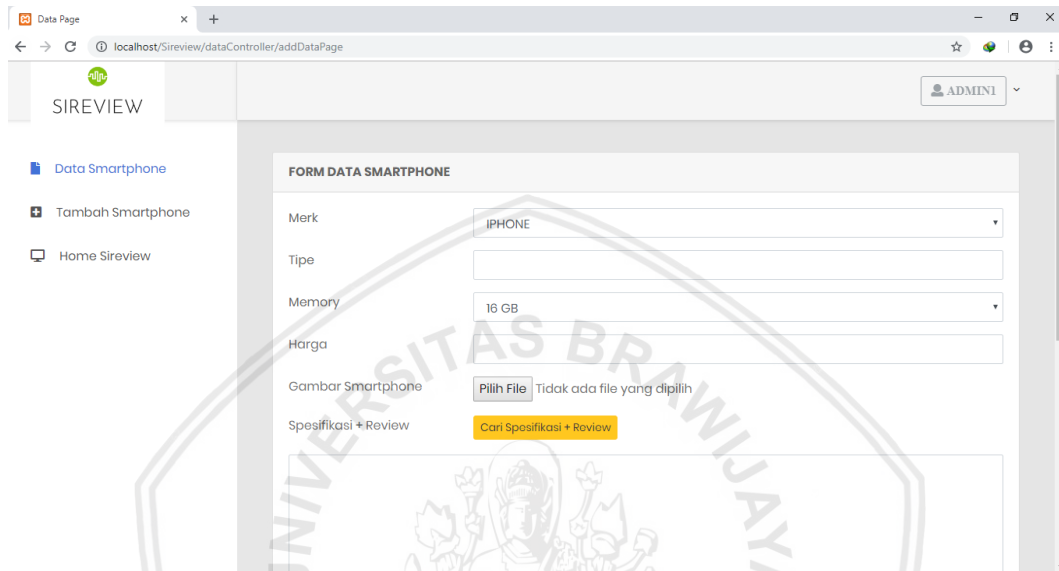
Gambar 6.3 menunjukkan pada halaman *reviewpage* menampilkan tabel dari *review* sebuah *smartphone* dengan klasifikasi sentimennya. Pada halaman ini juga

repository.ub.ac.id

ditampilkan jumlah *review* berdasarkan klasifikasi sentimennya. Terdapat juga tombol *process review*.

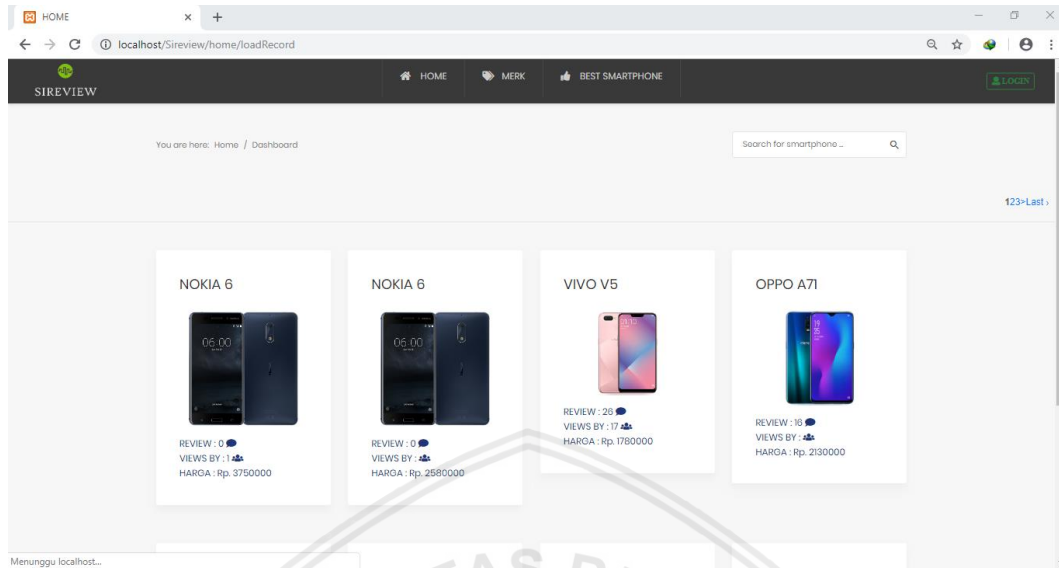
#### 6.2.3.4 Implementasi Antarmuka Halaman EditDataPage

Halaman editdatapage adalah halaman untuk proses merubah data. Gambar 6.4 menunjukkan antarmuka dari halaman editdatapage. Pada halaman tersebut terdapat *form* untuk melakukan perubahan data *smartphone*.



Gambar 6.4 Implementasi Antarmuka Halaman EditDataPage

### 6.2.3.5 Implementasi Antarmuka Halaman Home



**Gambar 6.5 Implementasi Antarmuka Halaman Home**

Gambar 6.5 merupakan antarmuka dari halaman home. Dimana pada halaman home ditampilkan daftar *smartphone* yang terdapat pada sistem beserta gambar dan keterangan dari *smartphone* seperti jumlah *review smartphone*, jumlah *views* dari *smartphone* tersebut, dan harga dari *smartphone* tersebut.

### 6.2.3.6 Implementasi Antarmuka Halaman DetailSmartphone

Pada halaman *detailsmartphone* ditampilkan informasi lengkap mulai dari gambar *smartphone*, detail dari *smartphone* seperti jumlah memori yang dimiliki, dan nilai dari setiap aspek *spesifikasi smartphone*. Pada halaman ini juga ditampilkan daftar *review* dari *smartphone* beserta dengan nama yang memberikan *review* dan klasifikasi sentimen dari *review*. Halaman *detailsmartphone* juga menampilkan jumlah *review*, jumlah *review* positif, jumlah *review* negatif, dan jumlah *views* dari sebuah *smartphone*. Gambar 6.6 menunjukkan antarmuka dari halaman *detailsmartphone*.

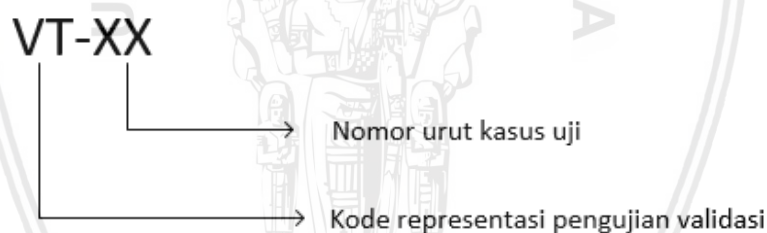


## BAB 7 PENGUJIAN DAN ANALISIS HASIL

Tahap pengujian merupakan salah satu tahap pengembangan perangkat lunak pada SDLC *Waterfall model*, tahapan ini bertujuan untuk memeriksa apakah solusi perangkat lunak yang telah dikembangkan telah sesuai dengan spesifikasi dan persyaratan yang telah disepakati dan apakah dapat memenuhi tujuannya. Pada bab ini menjelaskan mengenai proses pengujian dan analisis hasil pengujian dari Sistem Informasi *Review Smartphone*.

### 7.1 Pengujian *Black Box*

Pengujian *black box* merupakan pengujian yang dilakukan pada antar muka sistem yang berfokus terhadap kebutuhan fungsional sistem. Pengujian ini merupakan proses menjalankan sistem untuk menemukan kesalahan fungsi, kesalahan struktur, kesalahan kinerja dan/atau kesalahan antarmuka sistem. Proses pengujian dilakukan dengan membuat kasus uji yang merujuk pada *Use Case Scenario* yang telah di deskripsikan pada Tahap Analisis Kebutuhan. Kemudian hasil dari pengujian dibandingkan dengan hasil yang telah didefinisikan pada *Use Case Scenario*. Pada penelitian ini pengujian *Black Box* dilakukan dengan menggunakan teknik pengujian validasi. Gambar 7.1 menunjukkan aturan penomoran untuk pengujian validasi.



Gambar 7.1 Aturan penomoran pengujian validasi

#### 7.1.1 Pengujian Validasi *Scraping Spesifikasi dan Review*

Pengujian ini bermaksud untuk melakukan validasi terhadap proses *scraping* spesifikasi dan *review*, dimana proses ini adalah proses yang dilakukan untuk mendapatkan spesifikasi dan *review* dari situs yang telah ditentukan. Kasus uji pada pengujian merujuk pada *Use Case Scenario Scraping Spesifikasi dan Review*. Tabel 7.1 menjelaskan mengenai kasus uji proses *scraping* spesifikasi dan *review* dengan kode uji VT-01.

Tabel 7.1 Kasus Uji Pengujian Validasi *Scraping Spesifikasi dan Review*

<b>Kode Pengujian</b>	VT-01
<b><i>Use Case Scenario</i></b>	<i>Scraping Spesifikasi dan Review</i>



<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan bahwa sistem dapat melakukan mengambil ( <i>scraping</i> ) data spesifikasi dan <i>review</i> dari situs web yang sudah ditentukan
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji telah diidentifikasi sebagai admin</li> <li>2. Penguji memilih menu tambah <i>smartphone</i></li> <li>3. Penguji memilih fungsi Cari Spesifikasi + <i>Review</i></li> </ol>
<b>Hasil yang Diharapkan</b>	Sistem berhasil mendapatkan spesifikasi dan <i>review</i> dari situs web, dan kembali di arahkan ke halaman tambah <i>smartphone</i> dengan kolom spesifikasi dan <i>review</i> telah terisi oleh spesifikasi dan <i>review</i> dari hasil proses <i>scraping</i>
<b>Hasil Pengujian</b>	Sistem berhasil mendapatkan spesifikasi dan <i>review</i> dari situs web, dan kembali di arahkan ke halaman tambah <i>smartphone</i> dengan kolom spesifikasi dan <i>review</i> telah terisi oleh spesifikasi dan <i>review</i> dari hasil proses <i>scraping</i>
<b>Status Pengujian</b>	<i>Valid</i>

### 7.1.2 Pengujian Validasi Menambah Data *Smartphone*

Pada pengujian ini kasus uji merujuk pada *Use Case Scenario* Menambah Data *Smartphone*. Pengujian ini bertujuan untuk melakukan validasi terhadap fungsi menambah data *smartphone* pada sistem. Tabel 7.2 menjelaskan kasus uji menambah data *smartphone* dengan kode uji VT-02.

**Tabel 7.2 Kasus Uji Pengujian Validasi Menambah Data *Smartphone***

<b>Kode Pengujian</b>	VT-02
<b><i>Use Case Scenario</i></b>	Menambah Data <i>Smartphone</i>
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan bahwa sistem dapat menambah data <i>smartphone</i> dan menyimpannya ke dalam basis data sistem
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji telah diidentifikasi sebagai admin</li> <li>2. Penguji memilih menu tambah <i>smartphone</i></li> <li>3. Penguji mengisi <i>form</i> yang diperlukan untuk menambah data <i>smartphone</i></li> <li>4. Penguji memilih fungsi Cari Spesifikasi + <i>Review</i></li> </ol>

	5. Penguji memilih fungsi simpan
<b>Hasil yang Diharapkan</b>	Sistem berhasil menambahkan data <i>smartphone</i> dan menyimpannya ke dalam basis data, kemudian sistem mengarahkan ke halaman data <i>smartphone</i>
<b>Hasil Pengujian</b>	Sistem berhasil menambahkan data <i>smartphone</i> dan menyimpannya ke dalam basis data, kemudian sistem mengarahkan ke halaman data <i>smartphone</i>
<b>Status Pengujian</b>	<i>Valid</i>

### 7.1.3 Pengujian Validasi Memproses Review

Memproses *review* merupakan proses analisis sentimen pada *review*, yaitu proses untuk melakukan klasifikasi pada *review* yang telah di dapat dari hasil *scraping review*. Pada pengujian ini kasus uji merujuk pada *Use Case Scenario Memproses Review*, dengan kode pengujian VT-03. Tabel 7.3 menjelaskan kasus uji dari memproses *review*.

**Tabel 7.3 Kasus Uji Pengujian Validasi Memproses Review**

<b>Kode Pengujian</b>	VT-03
<b>Use Case Scenario</b>	Memproses <i>Review</i>
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan bahwa sistem dapat memproses <i>review</i> yaitu mengklasifikasikan <i>review</i> berdasarkan nilai sentimennya.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji telah diidentifikasi sebagai admin</li> <li>2. Penguji berada pada halaman <i>review</i></li> <li>3. Penguji memilih fungsi <i>process review</i></li> </ol>
<b>Hasil yang Diharapkan</b>	Sistem berhasil memproses <i>review</i> yaitu mengklasifikasikan <i>review</i> berdasarkan nilai <i>sentimen</i> , dan kembali diarahkan ke halaman <i>review</i> dengan kolom <i>sentimen</i> telah terisi nilai <i>sentimen</i> dari setiap <i>review</i>
<b>Hasil Pengujian</b>	Sistem berhasil memproses <i>review</i> yaitu mengklasifikasikan <i>review</i> berdasarkan nilai <i>sentimen</i> , dan kembali diarahkan ke halaman <i>review</i> dengan kolom <i>sentimen</i> telah terisi nilai <i>sentimen</i> dari setiap <i>review</i>
<b>Status Pengujian</b>	<i>Valid</i>

#### 7.1.4 Pengujian Validasi Merubah Data *Smartphone*

Pengujian ini bertujuan untuk melakukan validasi terhadap proses merubah data *smartphone* dengan kasus uji merujuk pada *Use Case Scenario* Merubah Data *Smartphone*. Tabel 7.4 menjelaskan kasus uji dari pengujian validasi merubah data *smartphone* dengan kode pengujian VT-04.

**Tabel 7.4 Kasus Uji Pengujian Validasi Merubah Data *Smartphone***

<b>Kode Pengujian</b>	VT-04
<b>Use Case Scenario</b>	Merubah Data <i>Smartphone</i>
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan bahwa sistem dapat merubah data <i>smartphone</i> yang ada pada sistem dan menyimpannya ke dalam basis data sistem.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji telah diidentifikasi sebagai admin</li> <li>2. Penguji berada pada halaman data <i>smartphone</i></li> <li>3. Penguji memilih menu edit</li> <li>4. Penguji merubah data pada <i>form</i> yang diperlukan untuk merubah data</li> <li>5. Penguji memilih fungsi simpan</li> </ol>
<b>Hasil yang Diharapkan</b>	Sistem berhasil merubah dan menyimpan data hasil perubahan pada basis data sistem. Sistem mengarahkan kembali ke halaman data <i>smartphone</i>
<b>Hasil Pengujian</b>	Sistem berhasil merubah dan menyimpan data hasil perubahan pada basis data sistem. Sistem mengarahkan kembali ke halaman data <i>smartphone</i>
<b>Status Pengujian</b>	<i>Valid</i>

### 7.1.5 Pengujian Validasi Menghapus Data *Smartphone*

Pengujian dilakukan dengan membuat kasus uji yang merujuk pada *Use Case Scenario* Menghapus Data *Smartphone*. Pengujian ini bertujuan untuk melakukan validasi pada fungsi hapus data pada sistem. Dengan kode pengujian VT-05, Tabel 7.5 menjelaskan kasus uji proses menghapus data.

**Tabel 7.5 Kasus Uji Pengujian Validasi Menghapus Data *Smartphone***

<b>Kode Pengujian</b>	VT-05
<b>Use Case Scenario</b>	Menghapus Data <i>Smartphone</i>
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan bahwa sistem dapat menghapus data <i>smartphone</i> yang tersimpan pada basis <i>data</i> .
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji telah diidentifikasi sebagai admin</li> <li>2. Penguji berada pada halaman data <i>smartphone</i></li> <li>3. Penguji memilih menu delete</li> <li>4. Penguji memilih Ok untuk konfirmasi hapus data</li> </ol>
<b>Hasil yang Diharapkan</b>	Sistem berhasil menghapus data dari sistem dan basis data sistem, dan sistem kembali diarahkan ke halaman data <i>smartphone</i>
<b>Hasil Pengujian</b>	Sistem berhasil menghapus data dari sistem dan basis data sistem, dan sistem kembali diarahkan ke halaman data <i>smartphone</i>
<b>Status Pengujian</b>	<i>Valid</i>

### 7.1.6 Pengujian Validasi Melihat *Detail Smartphone*

Proses melihat *detail smartphone* adalah proses dimana pengguna sistem baik petugas *counter* maupun konsumen dapat melihat informasi lengkap mengenai *smartphone* yaitu diantaranya adalah informasi mengenai gambar *smartphone*, spesifikasi *smartphone* dan *review* beserta nilai *sentimen* dari setiap *review*. Pada pengujian ini kasus uji dibuat dengan merujuk pada *Use Case Scenario* Melihat Detail *Smartphone*. Pada Tabel 7.6 menjelaskan mengenai kasus uji melihat *detail smartphone*, dengan nomor pengujian VT-06.

**Tabel 7.6 Kasus Uji Pengujian Validasi Melihat *Detail Smartphone***

<b>Kode Pengujian</b>	VT-06
<b>Use Case Scenario</b>	Melihat <i>Detail Smartphone</i>
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan bahwa sistem menampilkan informasi lengkap dari <i>smartphone</i> seperti gambar <i>smartphone</i> , spesifikasi <i>smartphone</i> , <i>review smartphone</i> beserta nilai <i>sentimen</i> dari setiap <i>review</i> .
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji berada pada halaman home</li> <li>2. Penguji memilih salah satu <i>smartphone</i></li> </ol>
<b>Hasil yang Diharapkan</b>	Sistem berhasil diarahkan ke halaman <i>detail smartphone</i> dan menampilkan informasi lengkap <i>smartphone</i> seperti gambar <i>smartphone</i> , <i>detail smartphone</i> , spesifikasi <i>smartphone</i> , jumlah <i>review smartphone</i> , dan <i>review</i> beserta nilai <i>sentimen</i> dari setiap <i>review</i> .
<b>Hasil Pengujian</b>	Sistem berhasil diarahkan ke halaman <i>detail smartphone</i> dan menampilkan informasi lengkap <i>smartphone</i> seperti gambar <i>smartphone</i> , <i>detail smartphone</i> , spesifikasi <i>smartphone</i> , jumlah <i>review smartphone</i> , dan <i>review</i> beserta nilai <i>sentimen</i> dari setiap <i>review</i> .
<b>Status Pengujian</b>	<i>Valid</i>

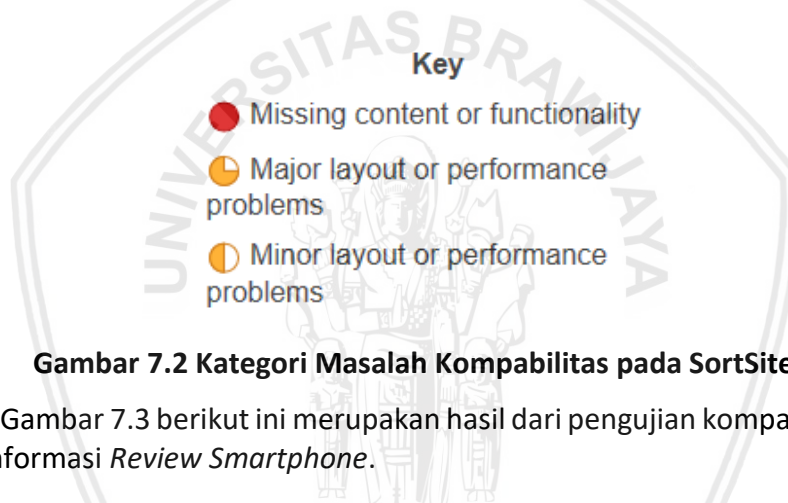
Berdasarkan dari pengujian validasi yang telah dilakukan, dari hasil yang di dapatkan dari kasus uji dengan kode nomor pengujian VT-01 sampai VT-06 adalah keseluruhan kasus uji yang dilakukan berstatus valid. Dengan demikian maka dapat disimpulkan bahwa sistem perangkat lunak yang dikembangkan dapat berjalan dan telah sesuai dengan hasil yang di harapkan. Berikut pada Tabel 7.7 menunjukkan rangkuman dari hasil pengujian validasi yang telah dilakukan pada kasus uji dengan nomor kode pengujian VT-01 sampai VT-06.

**Tabel 7.7 Tabel Hasil Pengujian Validasi Sistem Informasi *Review Smartphone***

Nomor Kode Pengujian	Hasil Pengujian
VT-01	<i>Valid</i>
VT-02	<i>Valid</i>
VT-03	<i>Valid</i>
VT-04	<i>Valid</i>
VT-05	<i>Valid</i>
VT-06	<i>Valid</i>

## 7.2 Pengujian *Compatibility*

Pengujian *compatibility* atau pengujian kompatibilitas pada penelitian ini dilakukan dengan menggunakan bantuan *tools* SortSite 5. Pada SortSite terdapat kategori masalah kompabilitas yang dapat diketahui, Gambar 7.2 menjelaskan mengenai kategori masalah tersebut.



**Gambar 7.2 Kategori Masalah Kompabilitas pada SortSite**

Pada Gambar 7.3 berikut ini merupakan hasil dari pengujian kompatibilitas dari Sistem Informasi *Review Smartphone*.

Browser	IE	Edge	Firefox	Safari	Opera	Chrome	iOS			Android		
Version	11	17	63	12	55	70	≤ 10	11	12	≤ 3	4*	
Critical Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Major Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Minor Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

**Gambar 7.3 Hasil *Compatibility Testing***

Berdasarkan pada hasil *compatibility testing* yang ada pada Gambar 7.3 diketahui bahwa Sistem Informasi *Review Smartphone* yang dibuat dapat berjalan dengan baik pada keseluruhan *browser* yang diuji.

## 7.3 Pengujian *Usability*

Pengujian *usability* atau *usability testing* adalah pengujian yang melibatkan perwakilan dari pengguna perangkat lunak yang diuji. Pengujian dilakukan dengan memberikan serangkaian *task-task* kepada pengguna. Mengacu pada standar



metric *usability* ISO/IEC 9126-4, akan dilakukan perhitungan mengenai *effectiveness*, *efficiency* dan *satisfaction* setelah *task-task* diberikan kepada perwakilan dari pengguna. Pada bagian ini akan dibagi menjadi tiga bagian yaitu masing-masing untuk menghitung metrik *effectiveness*, *efficiency*, dan *satisfaction*.

### 7.3.1 Metrik *Effectiveness*

Pada bagian ini pengujian *usability* untuk mengukur metrik *effectiveness* dimulai dengan perwakilan dari pengguna diberikan serangkaian *task-task*. Setelah perwakilan dari pengguna menjalankan *task-task* yang telah diberikan, dilakukan perhitungan nilai *effectiveness* dari *task-task* yang berhasil dijalankan guna mengukur metrik *effectiveness*. Nilai *effectiveness* yang telah didapatkan, diolah kembali untuk dilakukan perhitungan rata-rata nilai dari seluruh perwakilan pengguna sehingga didapatkan nilai rata-rata *effectiveness* dari keseluruhan perwakilan pengguna. Pada bagian pengukuran metrik *effectiveness* ini akan dibagi ke dalam dua bagian yaitu pengukuran metrik *effectiveness* pengguna admin dan pengukuran metrik *effectiveness* pengguna konsumen.

#### 7.3.1.1 Pengukuran Metrik *Effectiveness* Pengguna Admin

Pada Tabel 7.8 berikut ini merupakan tabel daftar *task* yang diberikan kepada perwakilan pengguna admin untuk pengujian *usability* metrik *effectiveness*. Tabel 7.8 ditandai dengan kode *task*, nama *task*, *Use Case*, dan skenario pengujian. Kode dari setiap *task* pada pengujian *usability* yang diberikan kepada pengguna admin akan diwakilkan dengan kode TA-XX. Dimana TA merupakan singkatan dari *Task Admin*, sedangkan XX merupakan penomoran dari kode *task* tersebut.

**Tabel 7.8 Daftar *Task Usability Testing* Pengguna Admin**

Kode Task	Nama Task	Use Case	Skenario Pengujian
TA-01	Menambah <i>user</i>	Menambah <i>user</i>	<ol style="list-style-type: none"> <li>1. Pengguna memilih pilihan menu <i>sign up</i> pada halaman login untuk membuka halaman menambah <i>user</i>.</li> <li>2. Pengguna mengisi semua kolom pada <i>form</i> yang diperlukan.</li> <li>3. Pengguna memilih tombol <i>submit</i> untuk menambah <i>user</i>.</li> </ol>
TA-02	Melakukan <i>Login</i>	<i>Login</i>	<ol style="list-style-type: none"> <li>1. Pengguna memasukkan <i>username</i> dan <i>password</i> pada <i>form</i> yang tersedia.</li> <li>2. Pengguna memilih tombol <i>login</i> untuk mendapatkan <i>autentifikasi login</i>.</li> </ol>

Kode Task	Nama Task	Use Case	Skenario Pengujian
TA-03	Menambah data <i>smartphone</i>	Menambah data <i>smartphone</i>	<ol style="list-style-type: none"> <li>1. Pengguna memilih menu tambah <i>smartphone</i>.</li> <li>2. Pengguna mengisi semua kolom pada <i>form</i> yang tersedia.</li> <li>3. Mengunggah gambar <i>smartphone</i>.</li> <li>4. Memilih tombol cari spesifikasi dan <i>review</i>.</li> <li>5. Memilih tombol simpan untuk menyimpan data <i>smartphone</i>.</li> </ol>
TA-04	Memproses <i>Review</i>	Memproses <i>review</i>	<ol style="list-style-type: none"> <li>1. Pengguna memilih <i>review</i> pada salah satu <i>smartphone</i>.</li> <li>2. Pengguna memilih tombol <i>process review</i> untuk memproses <i>review</i>.</li> </ol>
TA-05	Merubah data <i>smartphone</i>	Merubah data <i>smartphone</i>	<ol style="list-style-type: none"> <li>1. Pengguna memilih menu edit pada salah satu <i>smartphone</i>.</li> <li>2. Pengguna merubah data pada <i>form</i> rubah data yang tersedia</li> <li>3. Pengguna memilih tombol simpan untuk menyimpan data <i>smartphone</i> yang sudah dirubah.</li> </ol>
TA-06	Menghapus data <i>smartphone</i>	Menghapus data <i>smartphone</i>	<ol style="list-style-type: none"> <li>1. Pengguna memilih menu delete pada salah satu <i>smartphone</i>.</li> <li>2. Pengguna memilih ok pada <i>alert</i> konfirmasi untuk menghapus data <i>smartphone</i>.</li> </ol>
TA-07	Mencari <i>smartphone</i>	Mencari <i>smartphone</i>	<ol style="list-style-type: none"> <li>1. Pengguna memasukkan kata kunci pada kolom pencarian</li> <li>2. Pengguna menekan enter untuk mencari <i>smartphone</i></li> </ol>
TA-08	Melihat <i>detail smartphone</i>	Melihat <i>detail smartphone</i>	<ol style="list-style-type: none"> <li>1. Pengguna memilih menu home</li> <li>2. Pengguna memilih salah satu <i>smartphone</i> untuk melihat <i>detail smartphone</i></li> </ol>
TA-09	Melakukan <i>Logout</i>	<i>Logout</i>	<ol style="list-style-type: none"> <li>1. Pengguna memilih menu <i>logout</i> untuk keluar dari sistem</li> </ol>

Tabel 7.9 berikut ini merupakan hasil keberhasilan dari *task – task* pengujian *usability* yang telah diberikan kepada perwakilan pengguna admin. Hasil keberhasilan *task-task* digunakan untuk mengukur metrik *effectiveness*. Pada tabel 7.9 jika responden berhasil menyelesaikan suatu *task* maka pada kolom dari

daftar *task* akan diberikan nilai biner '1', sedangkan jika tidak dapat menyelesaikan maka akan diberikan nilai '0'. Sedangkan pada kolom *effectiveness* merupakan nilai persentase dari perhitungan *effectiveness* dengan menggunakan rumus 2-6.

**Tabel 7.9 Hasil Keberhasilan *Task* Pengguna Admin**

Pengguna	Kode Task								Effectiveness
	TA-01	TA-02	TA-03	TA-04	TA-05	TA-06	TA-07	TA-08	
Pengguna Admin	1	1	1	1	1	1	1	1	100%
<b>Rata-Rata</b>									100%

### 7.3.1.2 Pengukuran Metrik *Effectiveness* Pengguna Konsumen

Pada Tabel 7.10 berikut ini merupakan daftar *task* yang diberikan kepada perwakilan pengguna konsumen untuk pengujian *usability*. Tabel 7.10 ditandai dengan kode *task*, nama *task*, *Use Case*, dan skenario pengujian. Kode dari setiap *task* pada pengujian *usability* yang diberikan kepada pengguna konsumen akan diwakilkan dengan kode TK-XX. Dimana TK merupakan singkatan dari *Task* Konsumen, sedangkan XX merupakan penomoran dari kode *task* tersebut.

**Tabel 7.10 Daftar *Task Usability Testing* Pengguna Konsumen**

Kode Task	Nama Task	Use Case	Skenario Pengujian
TK-01	Melihat <i>detail smartphone</i>	Melihat <i>detail smartphone</i>	<ol style="list-style-type: none"> <li>1. Pengguna memilih menu home</li> <li>2. Pengguna memilih salah satu <i>smartphone</i> untuk melihat <i>detail smartphone</i></li> </ol>
TK-02	Mencari <i>smartphone</i>	Mencari <i>smartphone</i>	<ol style="list-style-type: none"> <li>1. Pengguna memasukkan kata kunci pada kolom pencarian</li> <li>2. Pengguna menekan enter untuk mencari <i>smartphone</i></li> </ol>

Tabel 7.11 berikut ini merupakan hasil keberhasilan dari *task – task* pengujian *usability* yang telah diberikan kepada perwakilan pengguna konsumen. Hasil keberhasilan *task-task* digunakan untuk mengukur metrik *effectiveness*. Pada Tabel 7.11 jika responden berhasil menyelesaikan suatu *task* maka pada kolom dari daftar *task* akan diberikan nilai biner '1', sedangkan jika tidak dapat menyelesaikan maka akan diberikan nilai '0'. Sedangkan pada kolom *effectiveness* merupakan nilai persentase dari perhitungan *effectiveness* dengan menggunakan rumus 2-6.

**Tabel 7.11 Hasil Keberhasilan *Task* Pengguna Konsumen**

Pengguna	Kode Task	Effectiveness
----------	-----------	---------------

	TK-01	TK-02	
A	1	1	100%
B	1	1	100%
C	1	1	100%
D	1	1	100%
E	1	1	100%
<b>Rata-Rata</b>			100%

Berdasarkan Tabel 7.11 didapatkan bahwa dari keseluruhan perwakilan pengguna konsumen berhasil menyelesaikan keseluruhan *task* yang diberikan.

### 7.3.2 Metrik *Efficiency*

Pengujian *usability* metrik *efficiency* dilakukan dengan mencatat waktu yang diperlukan oleh setiap perwakilan pengguna dalam menyelesaikan setiap *task-task* yang diberikan. Pengukuran metrik *efficiency* dilakukan dengan menghitung *time based efficiency* (TBE) dari setiap *task* yang diberikan kepada perwakilan pengguna, selanjutnya akan dihitung juga nilai TBE dari *task-task* untuk keseluruhan perwakilan pengguna. Pada pengujian *usability* metrik *efficiency* pada penelitian ini akan dibagi menjadi pengukuran metrik *efficiency* pengguna admin dan pengukuran metrik *efficiency* pengguna konsumen.

#### 7.3.2.1 Pengukuran Metrik *Efficiency* Pengguna Admin

Tabel 7.12 berikut ini merupakan hasil dari waktu yang diperlukan pengguna admin untuk mengerjakan setiap *task-task* yang diberikan, baik *task* itu selesai dikerjakan maupun tidak. Pada pengukuran metrik *efficiency* pengguna admin ini, kode dan setiap *task* yang diberikan sesuai dengan kode dan *task* yang ada pada pengukuran metrik *effectiveness* pengguna admin. Pada Tabel 7.12 nilai yang terdapat pada kolom dari setiap kode *task* merupakan waktu yang dibutuhkan oleh pengguna admin dalam mengerjakan *task* tersebut dalam satuan waktu detik. Sedangkan untuk nilai yang terdapat pada kolom TBE adalah nilai dari *task* yang selesai dikerjakan dalam satu detik, nilai tersebut di dapatkan dari perhitungan menggunakan Rumus 2-7.

**Tabel 7.12 Hasil Pengukuran Waktu *Task* Pengguna Admin**

Pengguna	Kode <i>Task</i>									TBE
	TA-01	TA-02	TA-03	TA-04	TA-05	TA-06	TA-07	TA-08	TA-09	
Pengguna Admin	26	7	73	33	29	4	3	7	3	0.132 <i>goals/sec</i>



### 7.3.2.2 Pengukuran Matrik Efficiency Pengguna Konsumen

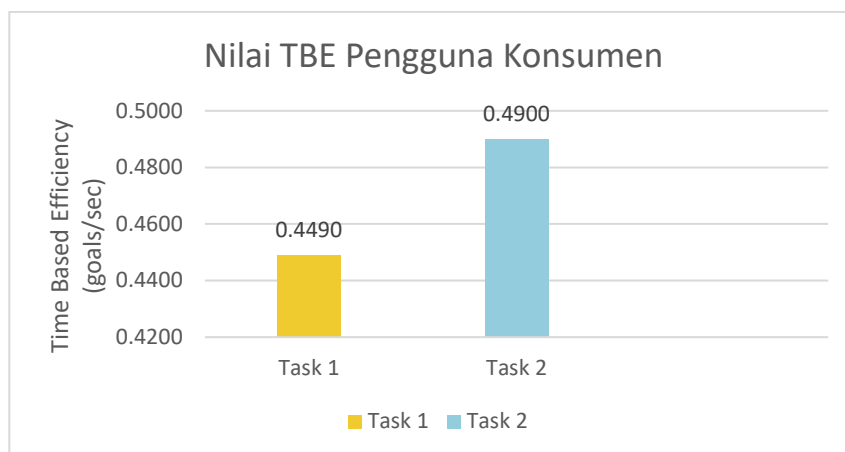
Tabel 7.13 berikut ini merupakan hasil dari waktu yang diperlukan pengguna konsumen untuk mengerjakan setiap *task-task* yang diberikan, baik *task* itu selesai dikerjakan maupun tidak. Pada pengukuran metrik *efficiency* pengguna konsumen ini, kode dan setiap *task* yang diberikan sesuai dengan kode dan *task* yang ada pada pengukuran metrik *effectiveness* pengguna konsumen. Pada Tabel 7.13 nilai yang terdapat pada kolom dari setiap kode *task* merupakan waktu yang dibutuhkan oleh pengguna konsumen dalam mengerjakan *task* tersebut dalam satuan waktu detik. Sedangkan untuk nilai yang terdapat pada kolom TBE adalah nilai dari *task* yang selesai dikerjakan dalam satu detik, nilai tersebut di dapatkan dari perhitungan menggunakan Rumus 2-7.

**Tabel 7.13 Perhitungan Metrik Efficiency Pengguna Konsumen**

Pengguna	Kode Task		TBE tiap Pengguna	TBE seluruh Pengguna
	TK-01	TK-02		
A	6	2	0.333 goals/sec	0.469 goals/sec
B	1	1	1 goals/sec	
C	4	4	0.25 goals/sec	
D	2	2	0.5 goals/sec	
E	3	5	0.266 goals/sec	
TBE tiap Task	0.449 goals/sec	0.49 goals/sec		
	0.469 goals/sec			

Untuk mempermudah dalam membaca hasil perhitungan metrik *efficiency* pada Tabel 7.13, maka berikut ini pada Gambar 7.4 disajikan grafik berupa grafik batang nilai TBE dari pengguna konsumen. Pada Gambar 7.4 sumbu X menjelaskan daftar *task* yang diberikan kepada pengguna konsumen, sedangkan sumbu Y menjelaskan nilai TBE dari masing-masing *task*.





Gambar 7.4 Grafik Batang Nilai TBE Pengguna Konsumen

### 7.3.3 Metrik *Satisfaction*

Pada bagian ini menjelaskan tentang pengukuran metrik *satisfaction* pada pengujian *usability* yang dimulai dengan memberikan kuisisioner *task level satisfaction* untuk perwakilan pengguna admin dan kuisisioner *test level satisfaction* untuk perwakilan pengguna konsumen. Pada pengujian *usability* pengukuran metrik *satisfaction* ini dibagi menjadi pengujian metrik *satisfaction* pengguna admin dan pengujian metrik *satisfaction* pengguna konsumen.

#### 7.3.3.1 Pengujian Metrik *Satisfaction* Pengguna Admin

Pada bagian ini pengujian *usability* untuk mengukur metrik *satisfaction* dimulai dengan memberikan kuisisioner *task level satisfaction* yang berupa *single ease question* (SEQ) kepada pengguna admin setelah setiap *task* selesai dikerjakan. Pertanyaan yang terdapat pada kuisisioner berupa pertanyaan mengenai seberapa sulit atau mudah sebuah *task* untuk dikerjakan. Pengguna dapat menjawabnya setiap pertanyaan tersebut dengan menggunakan poin-poin skala *likert* yang telah disediakan. Kemudian hasil dari jawaban pada kuisisioner *task level satisfaction* digunakan untuk mengukur metrik *satisfaction*.

Berikut ini pada Tabel 7.14 merupakan tabel hasil kuisisioner *task level satisfaction* yang diberikan kepada perwakilan pengguna admin. Tabel 7.14 ditandai dengan daftar pertanyaan, kode *task*, dan poin-poin skala *likert* sebagai jawaban dari setiap pertanyaan. Satu pertanyaan yang diberikan mengacu pada satu *task* yang diwakilkan oleh kode *task*. Untuk hasil kuisisioner *task level kuisisioner* dari responden admin akan dilampirkan pada bagian Lampiran B.



Tabel 7.14 Kuisiner *Task Level Satisfaction* Pengguna Admin

No.	Pertanyaan	Kode Task	Jawaban				
			Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
1	Apakah proses untuk mendaftarkan akun baru sebagai <i>administrator</i> mudah untuk dilakukan?	TA-01	0	0	0	1	0
2	Apakah proses untuk melakukan <i>login</i> sebagai administrator mudah untuk dilakukan?	TA-02	0	0	0	1	0
3	Apakah proses <i>scraping</i> spesifikasi dan <i>review</i> mudah untuk dilakukan?	TA-03	0	0	1	0	0
4	Apakah proses untuk menambahkan data <i>smartphone</i> mudah untuk dilakukan?	TA-04	0	0	0	1	0
5	Apakah proses untuk memproses <i>review</i> mudah untuk dilakukan?	TA-05	0	0	0	1	0
6	Apakah proses untuk merubah data <i>smartphone</i> mudah untuk dilakukan?	TA-06	0	0	0	1	0

No.	Pertanyaan	Kode Task	Jawaban				
			Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
7	Apakah proses untuk melihat <i>detail smartphone</i> mudah untuk dilakukan?	TA-07	0	0	0	1	0
8	Apakah proses untuk mencari <i>smartphone</i> mudah dilakukan?	TA-08	0	0	0	1	0
9	Apakah proses untuk melihat <i>review</i> mudah untuk dilakukan?	TA-09	0	0	1	0	0
10	Apakah proses untuk <i>logout</i> mudah untuk dilakukan?	TA-10	0	0	0	1	0
Total			0	0	2	8	0

Hasil dari kuisisioner *task level satisfaction* pengguna admin kemudian dilakukan perhitungan *skala likert* dengan menghitung total nilai berdasarkan pada jawaban dari responden. Perhitungan dilakukan dengan menggunakan rumus perhitungan *skala likert* seperti yang dijelaskan pada studi literatur.

$$A = 0 \times 5 = 0$$

$$B = 8 \times 4 = 32$$

$$C = 2 \times 3 = 6$$

$$D = 0 \times 2 = 0$$

$$E = 0 \times 1 = 0$$

$$\text{Total Nilai} = A + B + C + D + E$$

$$\text{Total Nilai} = 0 + 32 + 6 + 0 + 0 = 38$$

Kemudian dilanjutkan dengan menghitung nilai Y.

$$Y = \text{Jumlah Skala} \times \text{Jumlah Responden} \times \text{Jumlah Pertanyaan}$$

$$Y = 5 \times 1 \times 10 = 50$$

Selanjutnya dilakukan perhitungan nilai persentase dengan menggunakan rumus index seperti pada Rumus 2-12.

$$\text{Rumus Index} = (\text{Total Nilai} \div Y) \times 100\%$$

$$\text{Rumus Index} = (38 \div 50) \times 100\% = 76\% \text{ (Setuju)}$$

### 7.3.3.2 Pengujian Metrik Satisfaction Pengguna Konsumen

Pada bagian ini pengujian *usability* untuk mengukur metrik *satisfaction* dimulai dengan memberikan kuisisioner *test level satisfaction*, yaitu dengan menggunakan metode *System Usability Score (SUS)*. Dalam metode SUS kuisisioner terdiri dari sepuluh pertanyaan, pada penelitian ini daftar pertanyaan pada kuisisioner menggunakan pertanyaan template seperti yang sudah dijelaskan pada studi literatur. Pengguna dapat menjawab setiap pertanyaan tersebut dengan menggunakan poin-poin skala *likert* yang telah disediakan. Kemudian hasil dari jawaban pada kuisisioner *test level satisfaction* digunakan untuk mengukur metrik *satisfaction* pada pengguna admin.

Berikut ini pada tabel 7.15 merupakan tabel salah satu hasil kuisisioner *test level satisfaction* yang diberikan kepada perwakilan pengguna konsumen. Tabel 7.15 ditandai dengan daftar pertanyaandan poin-poin skala *likert* sebagai jawaban dari setiap pertanyaan. Untuk hasil kuisisioner *test level kuisisioner* dari responden konsumen akan dilampirkan pada bagian Lampiran B.

**Tabel 7.15 Hasil Kuisisioner Test Level Satisfaction Pengguna Konsumen**

No.	Pertanyaan	Jawaban				
		Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
1	Saya pikir saya akan sering untuk menggunakan sistem ini				V	
2	Saya menemukan sistem yang tidak rumit untuk digunakan				V	
3	Saya pikir sistemnya mudah untuk digunakan					V
4	Saya pikir saya akan membutuhkan dukungan dari	V				

No.	Pertanyaan	Jawaban				
		Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
	orang teknisi untuk dapat menggunakan sistem ini					
5	Saya menemukan berbagai fungsi dalam sistem ini terintegrasi dengan baik					V
6	Saya pikir ada terlalu banyak inkonsistensi dalam sistem ini	V				
7	Saya akan membayangkan bahwa kebanyakan orang akan belajar menggunakan sistem ini dengan sangat cepat.			V		
8	Saya menemukan sistem yang sangat rumit untuk digunakan	V				
9	Saya merasa sangat yakin menggunakan sistem ini.				V	
10	Saya perlu belajar banyak hal sebelum saya dapat			V		



No.	Pertanyaan	Jawaban				
		Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
	menggunakan sistem ini.					

Hasil dari kuisisioner *test level satisfaction* pengguna konsumen kemudian dilakukan perhitungan nilai SUS (*SUS Score*). Pada Tabel 7.15 untuk di menghitung nilai SUS, jawaban di konversikan ke dalam bentuk skala nilai angka yaitu dimulai dari 'sangat tidak setuju' bernilai 1 , kemudian untuk jawaban selanjutnya nilai bertambah 1 hingga untuk jawaban 'sangat setuju' bernilai 5.

Perhitungan nilai SUS dihitung dengan menggunakan rumus pada berikut ini

1. Untuk setiap pertanyaan pada nomor ganjil, nilai dari jawaban dikurangi satu.
2. Untuk setiap pertanyaan pada nomor genap, nilainya adalah 5 dikurangi nilai dari jawaban.
3. Kemudian jumlahkan keseluruhan nilai baru yang di dapatkan dan hasilnya dikalikan 2.5

Berikut ini perhitungan nilai SUS dari Tabel 7.15, perhitungan ini dilakukan kepada setiap hasil *test level satisfaction* dari setiap responden.

1. Pada pertanyaan nomor 1 dipilih jawaban setuju yang memiliki skala nilai 4. Maka skor untuk pertanyaan nomor 1 adalah 3.
2. Pada pertanyaan nomor 2 dipilih setuju yang memiliki skala nilai 4. Maka skor untuk pertanyaan nomor 2 adalah 1.
3. Pada pertanyaan nomor 3 dipilih jawaban sangat setuju yang memiliki skala nilai 5. Maka skor untuk pertanyaan nomor 3 adalah 4.
4. Pada pertanyaan nomor 4 dipilih sangat tidak setuju yang memiliki skala nilai 1. Maka skor untuk pertanyaan nomor 4 adalah 4.
5. Pada pertanyaan nomor 5 dipilih jawaban sangat setuju yang memiliki skala nilai 5. Maka skor untuk pertanyaan nomor 5 adalah 4.
6. Pada pertanyaan nomor 6 dipilih sangat tidak setuju yang memiliki skala nilai 1. Maka skor untuk pertanyaan nomor 6 adalah 4.
7. Pada pertanyaan nomor 7 dipilih jawaban netral yang memiliki skala nilai 3. Maka skor untuk pertanyaan nomor 7 adalah 2.



8. Pada pertanyaan nomor 8 dipilih sangat tidak setuju yang memiliki skala nilai 1. Maka skor untuk pertanyaan nomor 8 adalah 4.
9. Pada pertanyaan nomor 9 dipilih jawaban setuju yang memiliki skala nilai 4. Maka skor untuk pertanyaan nomor 9 adalah 3.
10. Pada pertanyaan nomor 10 dipilih tidak setuju yang memiliki skala nilai 2. Maka skor untuk pertanyaan nomor 10 adalah 3.

Untuk mendapatkan skor nilai dari responden A pada Tabel 7.15, maka seluruh nilai skor baru dari setiap pertanyaan di jumlahkan dan dikalikan 2.5.

$$\text{Skor Responden A : } (3+1+4+4+4+4+2+4+3+3) \times 2.5 = 70.$$

Perhitungan diatas diulangi sampai didapatkan skor SUS dari *test level satisfaction* setiap responden. Berikut ini skor SUS dari setiap *test level satisfaction* setiap responden :

Responden A : 70

Responden B : 70

Responden C : 75

Responden D : 65

Responden E : 70

Selanjutnya dilakukan perhitungan nilai rata-rata skor SUS dari keseluruhan responden dengan rumus sebagai berikut

Rata-rata nilai skor SUS = jumlah keseluruhan skor responden ÷ jumlah responden

$$\text{Rata-rata nilai skor SUS} = (70+70+75+65+70) \div 5 = 70.$$

Berdasarkan rujukan, bahwa perangkat lunak yang mempunyai kualitas yang baik berada pada skor  $\geq 70$ . Maka melihat hasil skor SUS dari perhitungan matrik *satisfaction* pengguna konsumen yang memiliki rata-rata nilai skor 70 menunjukkan bahwa Sistem Informasi *Review Smartphone* memiliki kualitas yang baik.



## BAB 8 PENUTUP

### 8.1 Kesimpulan

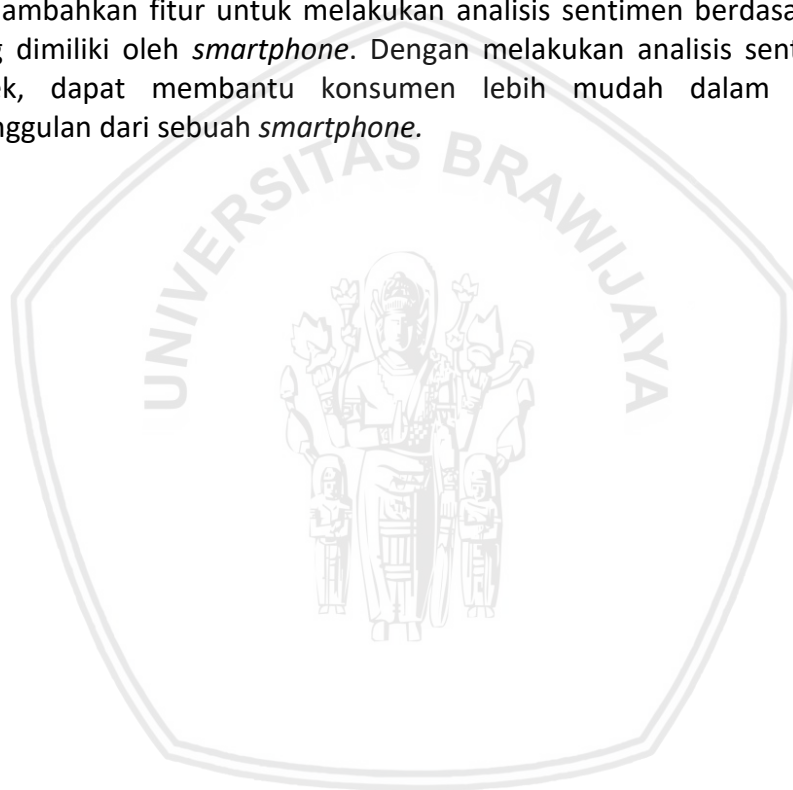
Berdasarkan penelitian yang telah dilakukan, kesimpulan yang dapat diambil dari pengembangan Sistem Informasi *Review Smartphone* adalah sebagai berikut:

1. Berdasarkan hasil dari analisis kebutuhan sistem untuk Sistem Informasi *Review Smartphone* didapatkan bahwa spesifikasi dan persyaratan untuk sistem diantaranya adalah kebutuhan fungsional sistem untuk Tambah data *smartphone*, *Scraping* spesifikasi dan *review*, Proses *review*, Rubah data *smartphone*, Hapus data *smartphone*, dan Lihat detail *smartphone*. Kemudian kebutuhan fungsional tersebut dimodelkan ke dalam diagram *use case*, selanjutnya masing-masing *Use Case* di jelaskan dengan menggunakan *Use Case Scenario*.
2. Hasil dari perancangan Sistem Informasi *Review Smartphone* adalah terbentuknya rancangan arsitektur sistem, rancangan *sequence* diagram sistem, rancangan *class* diagram sistem, rancangan skema basis data sistem, rancangan antarmuka pengguna serta perancangan algoritme sistem.
3. Hasil Implementasi dari rancangan sistem untuk Sistem Informasi *Review Smartphone* adalah Sistem Informasi *Review Smartphone* yang memiliki fitur diantaranya yaitu menambah data *smartphone*, *scraping* spesifikasi dan *review*, memproses *review*, merubah data *smartphone*, menghapus data *smartphone* serta melihat *detail smartphone*. Dalam tahap pengembangannya, fitur *scraping* spesifikasi dan *review* menggunakan *library* DOM sehingga sistem dapat mengambil spesifikasi dan *review* dari situs web yang sudah ditentukan. Selain itu, fitur memproses *review* juga telah berhasil dalam mengklasifikasikan *review* ke dalam klasifikasi positif dan negatif.
4. Berdasarkan hasil pengujian sistem pada Sistem Informasi *Review Smartphone* yang dilakukan menggunakan pengujian validasi didapatkan hasil 100% valid, dengan begitu dapat disimpulkan bahwa sistem yang dikembangkan dapat berjalan baik dan telah sesuai dengan kebutuhan pengguna. Selanjutnya pada pengujian *compatibility* didapatkan hasil bahwa sistem dapat berjalan baik pada beberapa *brower* yang berbeda. Kemudian untuk pengujian *usability* didapatkan hasil pada metrik *effectiveness* yaitu nilai rata-ratanya adalah 100% baik untuk admin maupun konsumen. Maka dapat disimpulkan bahwa *task* yang diberikan dapat diselesaikan dengan baik oleh pengguna. Selanjutnya pada pengujian matrik *efficiency* mendapatkan nilai TBE 0.132 *goals/sec*. Kemudian pada pengujian matrik *satisfaction* dengan menggunakan *system usability scale* (SUS) , menghasilkan skor 70. Maka dapat disimpulkan bahwa sistem memiliki kualitas yang bagus.

## 8.2 Saran

Saran yang dapat diberikan kepada peneliti selanjutnya adalah untuk melakukan pengembangan lanjut pada Sistem Informasi *Review Smartphone* diantaranya adalah sebagai berikut

1. Menambahkan sumber data *review* dari sumber data lain seperti media sosial atau platform lain yang sejenisnya. Penambahan sumber data dilakukan untuk mendapatkan *review* lebih banyak.
2. Perlu dilakukannya pengembangan lebih lanjut pada fitur memproses *review* (analisis sentimen) agar dapat mengklasifikasikan *review* ke dalam klasifikasi positif, negatif, atau netral.
3. Menambahkan fitur untuk melakukan analisis sentimen berdasarkan aspek yang dimiliki oleh *smartphone*. Dengan melakukan analisis sentimen pada aspek, dapat membantu konsumen lebih mudah dalam memahami keunggulan dari sebuah *smartphone*.



## DAFTAR REFERENSI

- Agarwal, B. B., Tayal, S. & Gupta, M., 2010. *Software Engineering & Testing*. Ontario, Jones and Bartlett.
- Alshamrani, A. & Bahattab, A., 2015. *A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model*. *IJCSI International Journal of Computer Science Issues*, XII(1), pp. 106-111.
- Anhar, 2010. *PHP & MySql Secara Otodidak*. Jakarta: Mediakita.
- Bertram, D., 2016. *Likert Scale*. s.l.:s.n.
- Cao, Q., Duan, W. & Gan, Q., 2011. *Exploring determinants of voting for the "helpfulness" of online user reviews: A text mining approach*. *Decision Support Systems*, Issue 50, pp. 511-521.
- Choizes, 2017. *Pengertian Skala Likert dan Contoh Cara Hitung Kuesionernya*. [Online]  
Tersedia di: <https://www.diedit.com/skala-likert/>  
[Diakses 12 Maret 2018].
- Desai, M. & Mehta, M. A., 2016. *Techniques for Sentiment Analysis of Twitter Data : A Comprehensive Survey*. s.l., International Conference on Computing, Communication And Automation (ICCCA).
- endorsement, O. r. a., 2015. *Online reviews and endorsement Report on the CMA's call for information*, s.l.: Competition and Markets Authority report.
- Everett, G. D. & Jr, R. M., 2007. *Software Testing : Testing Across the Entire Software Development Life Cycle*. New Jersey: John Wiley & Sons, Inc.
- Fachrurrozi, M. & Yusliani, N., 2015. *ANALISIS SENTIMEN PENGGUNA JEJARING SOSIAL MENGGUNAKAN METODE SUPPORT VECTOR MACHINE*. Manado, Researchgate.
- Feldman, R. & Sanger, J., 2007. *THE TEXT MINING HANDBOOK*. New York: Cambridge University Press.
- Group, O. M., n.d. *UML*. [Online]  
Tersedia di: <http://www.omg.org/technology/readingroom/UML.htm>  
[Diakses 16 Februari 2018].
- Havey, M., 2005. *Essential Business Process*. s.l.:Oreilly.
- Hughes, S. & Cohen, D., 2011. *Can Online Consumers Contribute to Drug Knowledge? A MixedMethods Comparison of Consumer-Generated and Professionally Controlled Psychotropic Medication Information on the Internet*. *Journal of Medical Internet Research*, XIII(3).
- Hutahean, J., 2014. *Konsep Sistem Informasi*. 1st ed. Yogyakarta: Deepublish.

- Kogoya, D., 2015. *DAMPAK PENGGUNAAN HANDPHONE PADA MASYARAKAT : Studi Pada Masyarakat Desa Piungun Kecamatan Gamelia Kabupaten Lanny Jaya Papua. Acta Diurna*, IV(4).
- Liu, B., 2010. *Sentiment Analysis : A Multi-Faceted Problem. IEEE Intelligent Systems*.
- Lukmanul, H. & Musalini, U., 2004. *Cara Cerdas Menguasai Layout, Desain dan Aplikasi Web*. Jakarta: PT Elex Media Komputindo.
- Manulu, B., 2010. *ANALISIS SENTIMEN PADA TWITTER MENGGUNAKAN TEXT MINING*, s.l.: s.n.
- Medhat, W., Hassan, A. & Korashy, H., 2014. *Sentiment analysis algorithm and applications : A survey. Ain Shams Engineering Journal*, V(4), pp. 1093-1113.
- Mifsud, J., 2015. *Usability Metrics – A Guide To Quantify The Usability Of Any System*. [Online]  
Tersedia di: <https://usabilitygeek.com/usability-metrics-a-guide-to-quantify-system-usability/>  
[Diakses 10 Desember 2018].
- Mifsud, J., 2015. *Usabilitygeek*. [Online]  
Tersedia di: <https://usabilitygeek.com/usability-metrics-a-guide-to-quantify-system-usability/>  
[Diakses 9 Mei 2018].
- Mishra, J., 2011. *Software Engineering*. India: Pearson Education.
- Nisa, U. Z., 2013. *Model Prediksi Finansial Distress Pada Perusahaan Manufaktur Go Public di Indonesia*, s.l.: Thesis Program Magister Bidang Optimasi Sistem Informasi Jurusan Teknik Industri Institut Teknologi Sepuluh Nopember.
- Object Management Group, 2011. *Business Process Model and Notation (BPMN)*. [Online]  
Tersedia di: <http://www.omg.org/spec/BPMN/2.0/PDF/>  
[Diakses 1 April 2018].
- Pendidikan, G., 2015. *PENGERTIAN PEMODELAN PROSES BISNIS MENURUT PARA AHLI*. [Online]  
Available at: <http://www.gurupendidikan.com/pengertian-pemodelan-prosesbisnis-menurut-para-ahli/>  
[Accessed 01 April 2018].
- Polidoro, F. et al., 2015. *Web scraping techniques to collect data on consumer electronics and airfares for Italian HICP compilation. Statistical Journal of the IAOS*, Issue 31, pp. 165-176.

- Pramana, Y. & Mungguna, W., 2015. *Implementasi Web Scrapping pada Website Ednesia.Com untuk Pengukur Kesenjangan Jumlah Mahasiswa Perguruan Tinggi di Indonesia*. *ULTIMA InfoSys*, VI(2), pp. 130-135.
- Pressman, R. S., 2010. *Software Engineering : a practioner's approach*. 7th ed. New York: McGraw-Hill.
- Programmer, P., 2013. *Pengenalan BPMN*. [Online] Tersedia di: <http://pojokprogrammer.net/content/pengenalan-bpmn> [Diakses 1 April 2018].
- Royce, W. W., 1970. *MANAGING THE DEVELOPMENT OF LARGE SOFTWARE SYSTEM*. *IEEE WESCON*, pp. 1-9.
- Rubin, J. & Chisnell, D., 2008. *Handbook of Usability Testing, Second Edition: How to Plan, Design, and Conduct*. 2nd ed. Indianapolis: Wiley Publishing, Inc.
- Rumbaugh, J., Jacobson, I. & Booch, G., 2005. *The Unified Modeling Language Reference Manual*. 2nd ed. Boston: Addison-Wesley.
- Sangeetha, T., Balaganesh, M. & Muneeswaran, D., 2017. *Aspect based Opinion Mining from Online Reviews for Product Recommendation*. s.l., s.n.
- Santoso, B., 2007. *Data Mining : Teknik Pemanfaatan Data untuk Keperluan Bisnis*. Yogyakarta: Graha Ilmu.
- Sauro, J., 2012. *Measuringu*. [Online] Tersedia di: <https://measuringu.com/seq10/> [Diakses 9 Mei 2018].
- Scott, K., 2004. *Fast Track UML 2.0*. USA: Appress.
- Sommerville, I., 2011. *Software Engineering*. 9th ed. Boston: Addison-Wesley.
- Sukanto, R. & Shalahudin, M., 2013. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika.
- Turland, M., 2010. *php/architect's Guide to Web Scrapping with PHP*. 1st ed. Toronto: Marco Tabini.
- Wikipedia, 2018. *Ponsel Cerdas*. [Online] Tersedia di: [https://id.wikipedia.org/wiki/Ponsel\\_cerdas](https://id.wikipedia.org/wiki/Ponsel_cerdas) [Diakses 14 Maret 2018].
- Wirartri, L., 2015. *Orang Indonesia Lebih Suka Beli Ponsel Langsung Ketimbang Online*. [Online] Tersedia di: <https://techno.okezone.com/read/2015/07/08/207/1178314/orang-indonesia-lebih-suka-beli-ponsel-langsung-ketimbang-online> [Diakses 26 Maret 2018].