



**PENGEMBANGAN MODEL *FILTERISASI* TEKS DENGAN
ALGORITMA *BOYER-MOORE* PADA PROTOTIPE PLATFORM
KOMUNIKASI PERSONAL**

SKRIPSI

PANDU UTOMO

2110511007

**PROGRAM STUDI S1 INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN” JAKARTA
JAKARTA
2025**



**PENGEMBANGAN MODEL *FILTERISASI* TEKS DENGAN
ALGORITMA *BOYER-MOORE* PADA PROTOTIPE PLATFORM
KOMUNIKASI PERSONAL**

SKRIPSI

**Diajukan Sebagai Salah Satu Syarat untuk Memperoleh Gelar Sarjana
Komputer**

PANDU UTOMO

2110511007

**PROGRAM STUDI S1 INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN” JAKARTA
JAKARTA
2025**

PERNYATAAN ORISINALITAS

PERNYATAAN ORISINALITAS

Tugas akhir ini adalah hasil karya sendiri dan semua sumber baik yang dikutip maupun yang dirujuk telah saya nyatakan dengan benar.

Nama : Pandu Utomo

NIM : 2110511007

Tanggal : 10 Juli 2025

Bilamana di kemudian hari ditemukan ketidaksesuaian dengan pernyataan saya ini, maka saya bersedia dituntut dan diproses sesuai dengan ketentuan yang berlaku

Jakarta, 10 Juli 2025

Yang Menyatakan



Pandu Utomo

PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS

PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai civitas akademika Universitas Pembangunan Nasional Veteran Jakarta, saya yang bertanda tangan di bawah ini:

Nama : Pandu Utomo
NIM : 2110511007
Fakultas : Ilmu Komputer
Program Studi : S-1 Informatika

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Pembangunan Nasional Veteran Jakarta Hak Bebas Royalti Non eksklusif (Non - exclusive Royalty Free Right) atas skripsi saya yang berjudul:

PENGEMBANGAN MODEL FILTERISASI TEKS DENGAN ALGORITMA BOYER-MOORE PADA PROTOTYPE PLATFORM KOMUNIKASI PERSONAL

Dengan Hak Bebas Royalti ini Universitas Pembangunan Nasional Veteran Jakarta berhak menyimpan, mengalih media/memformatkan, mengelola dalam bentuk pangkalan data (basis data), merawat dan mempublikasikan skripsi saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemilik Hak Cipta. Demikian pernyataan ini saya buat dengan sebenarnya

Dibuat di: Jakarta

Pada tanggal: 10 Juli 2025

Yang Menyatakan



Pandu Utomo

LEMBAR PENGESAHAN

LEMBAR PENGESAHAN

Judul : PENGEMBANGAN MODEL FILTERISASI TEKS DENGAN ALGORITMA BOYER-MOORE PADA PROTOTYPE PLATFORM KOMUNIKASI PERSONAL
Nama : Pandu Utomo
NIM : 2110511007
Program : SI Informatika
Studi

Disetujui oleh:

Penguji 1:
Bayu Hananto, S.Kom., M.Kom.

Penguji 2:
I Wayan Rangga Pinastawa, M.Kom.

Pembimbing 1:
Jayanta, S.Kom., M.Si.

Pembimbing 2:
Kharisma Wiati Gusti, M.T.

Diketahui oleh:

Koordinator Program Studi:
Dr. Widya Cholil, M.I.T.
NIP. 2211122080
Dekan Fakultas Ilmu Komputer:
Prof. Dr. Ir. Supriyanto, S.T., M.Sc., IPM
NIP. 197605082003121002

Tanggal Ujian Tugas Akhir:
4 Juni 2025



PENGEMBANGAN MODEL *FILTERISASI* TEKS DENGAN ALGORITMA *BOYER-MOORE* PADA PROTOTIPE PLATFORM KOMUNIKASI PERSONAL

Pandu Utomo

ABSTRAK

Dalam era komunikasi digital yang semakin terbuka, penting untuk menjaga interaksi tetap sesuai dengan etika dan norma, terutama melalui moderasi konten guna mencegah penyebaran kata-kata tidak pantas. Penelitian ini membahas penerapan algoritma *Boyer-Moore* dalam sistem *filterisasi* teks pada prototipe platform komunikasi personal berbasis *web*. Algoritma ini digunakan untuk mencocokkan *input* teks dengan daftar kata tidak pantas (*blacklist*) menggunakan pendekatan pencocokan pola berbasis *heuristic*. Namun, hasil pengujian menunjukkan bahwa algoritma *Boyer-Moore* secara mandiri memiliki keterbatasan dalam akurasi (12,5%), meskipun waktu prosesnya sangat cepat ($\pm 0,023$ detik). Hal ini mengindikasikan ketidakefektifan *Boyer-Moore* dalam menangani variasi penulisan kata tidak pantas. Sebaliknya, algoritma *Fuzzy Regex* menunjukkan akurasi yang lebih baik (62,5%) dengan waktu proses yang masih dapat diterima ($\pm 0,048$ detik). Kombinasi antara *Boyer-Moore*, *Fuzzy Regex*, teknik normalisasi, dan penanganan *overlap* menghasilkan peningkatan signifikan dalam akurasi deteksi hingga 95,8% dengan waktu proses rata-rata 0,095 detik. Temuan ini menunjukkan bahwa peran algoritma *Boyer-Moore* lebih tepat sebagai komponen pelengkap dalam sistem *filterisasi* yang lebih kompleks dan adaptif.

Kata Kunci: Filterisasi Konten, Algoritma *Boyer-Moore*, Moderasi Otomatis, Komunikasi Anonim, Pencocokan String

PENGEMBANGAN MODEL *FILTERISASI* TEKS DENGAN ALGORITMA *BOYER-MOORE* PADA PROTOTYPE PLATFORM KOMUNIKASI PERSONAL

Pandu Utomo

ABSTRACT

In the increasingly open era of digital communication, it is essential to maintain interactions within ethical and normative boundaries, especially through content moderation to prevent the spread of inappropriate language. This study examines the implementation of the Boyer-Moore algorithm in a prototype web-based personal communication platform for detecting and blocking offensive terms. The algorithm matches user input against a predefined list of inappropriate words using a pattern-matching approach based on heuristic shifts. However, test results indicate that the Boyer-Moore algorithm alone has limited accuracy (12.5%), although it offers very fast processing times (approximately 0.023 seconds per check). This suggests that Boyer-Moore is less effective in handling variations of inappropriate words. In contrast, the Fuzzy Regex algorithm achieves higher accuracy (62.5%) with slightly slower processing (around 0.048 seconds). By combining Boyer-Moore, Fuzzy Regex, normalization techniques, and overlap handling, the system achieves significantly improved detection accuracy of up to 95.8% with an average processing time of 0.095 seconds. These findings suggest that Boyer-Moore functions better as a supporting component within a more comprehensive and adaptive filtering system, rather than as a standalone solution.

Keywords: *Content Filtering, Boyer-Moore Algorithm, Automated Moderation, Anonymous Communication, String Matching*

KATA PENGANTAR

Segala puji dan syukur kami panjatkan kepada Allah Subhanahu Wa Ta'ala atas nikmat iman, Islam, kesehatan, dan umur panjang, sehingga penelitian ini dapat diselesaikan dengan baik. Sehubungan dengan hal tersebut, penulis ingin menyampaikan terima kasih kepada:

1. Bapak Prof. Dr. Ir. Supriyanto, M.Sc., IPM. selaku Dekan Fakultas Ilmu Komputer Universitas Pembangunan Nasional Veteran Jakarta
2. Ibu Dr. Widya Cholil, M.I.T. selaku Koordinator Program Studi S1 Informatika
3. Pak Jayanta, S.Kom., M.Si sebagai pembimbing pertama proposal dan skripsi, atas bimbingan, motivasi, serta saran konstruktif yang diberikan.
4. Ibu Kharisma Wiati Gusti, M.T. sebagai pembimbing kedua proposal dan skripsi, atas bimbingan, dukungan, serta saran konstruktif yang telah diberikan, sehingga penulis dapat menyelesaikan skripsi ini.
5. Ibu dan Bapak dosen Informatika Universitas Pembangunan Nasional Veteran Jakarta yang telah memberikan ilmu selama saya menempuh pendidikan ini.
6. Kedua orang tua serta keluarga besar yang selalu memberikan dukungan tulus kepada penulis, dan
7. Teman-teman prodi Informatika yang telah saling membantu dan saling memberikan semangat dalam menyelesaikan skripsi.

Penulis berharap agar segala kebaikan dan bantuan yang diberikan oleh berbagai pihak mendapatkan berkah dari Tuhan Yang Maha Esa. Penulis menyadari bahwa proposal tugas akhir ini masih belum sempurna karena keterbatasan pengetahuan yang dimiliki. Oleh karena itu, dengan segala kerendahan hati, penulis sangat mengharapkan masukan dan kritik yang membangun dari berbagai pihak demi perbaikan laporan akhir ini.

Jakarta, 10 Mei 2025



Pandu Utomo

DAFTAR ISI

LEMBAR JUDUL	i
PERNYATAAN ORISINALITAS	ii
PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS	iii
LEMBAR PENGESAHAN	iv
ABSTRAK	v
ABSTRACT	vi
KATA PENGANTAR	vii
DAFTAR ISI	viii
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
DAFTAR SIMBOL	xvi
DAFTAR RUMUS	xxv
BAB I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
1.6 Sistematika Penulisan	3
BAB II. TINJAUAN PUSTAKA	5
2.1 Landasan Teori	5
2.1.1. <i>String Matching</i>	5
2.1.2. Algoritma <i>Boyer-Moore</i>	5
2.1.3. <i>Overlap</i> dalam Pencocokan Teks	6
2.1.4. <i>Leet Speak (L33t)</i>	6
2.1.5. <i>Fuzzy Regex</i>	7
2.1.6. <i>T5 (Text-to-Text Transfer Transformer)</i>	7
2.1.7. <i>JavaScript</i>	7
2.1.8. <i>TypeScript</i>	7
2.1.9. <i>Git</i>	7
2.1.10. <i>Postman</i>	8

2.1.11. <i>Prisma ORM</i>	8
2.1.12. <i>Yarn Package Manager</i>	8
2.1.13. <i>Next.js</i>	8
2.1.14. <i>RESTful API</i>	8
2.1.15. <i>ExpressJS</i>	9
2.1.16. <i>PostgreSQL</i>	9
2.1.17. <i>Rapid Application Development (RAD)</i>	9
2.1.18. <i>Pengujian Blackbox</i>	10
2.1.19. <i>User Acceptance Testing (UAT)</i>	11
2.1.20. <i>Flowchart</i>	11
2.1.21. <i>Unified Modelling Language (UML)</i>	13
2.1.22. <i>Use Case Diagram</i>	13
2.1.23. <i>Entity Relationship Diagram (ERD)</i>	14
2.1.24. <i>Activity Diagram</i>	16
2.1.25. <i>Sequence Diagram</i>	18
2.1.26. <i>Class Diagram</i>	20
2.2 <i>Penelitian Terdahulu</i>	21
BAB III. METODE PENELITIAN	24
3.1 <i>Metode Penelitian</i>	24
3.1.1 <i>Studi Literatur</i>	24
3.1.2 <i>Perancangan Kebutuhan</i>	25
3.1.3 <i>Prototipe</i>	25
3.1.4 <i>Pengembangan (Construction)</i>	65
3.1.5 <i>Implementasi (Cutover)</i>	66
3.1.6 <i>Dokumentasi</i>	67
3.2 <i>Rancangan Solusi/Metode yang Diusulkan</i>	67
3.3 <i>Metode Analisis</i>	67
3.3.1 <i>Algoritma Boyer-Moore</i>	68
3.4 <i>Alat Bantu Penelitian</i>	69
3.4.1 <i>Perangkat Keras</i>	69
3.4.2 <i>Perangkat Lunak dan Lingkungan Pengembangan</i>	69
3.5 <i>Jadwal Penelitian</i>	70
BAB IV. HASIL DAN PEMBAHASAN	71
4.1 <i>Model Filter Teks</i>	71
4.1.1 <i>Identifikasi Kata Tidak Sopan dan Sopan</i>	71

4.1.2 Pembuatan Daftar Kata.....	71
4.1.3 Pemetaan Kata Tidak Sopan ke Sopan	71
4.1.4 Implementasi <i>Boyer-Moore</i>	72
4.1.5 Implementasi <i>Text to Text Transfer Transformer Models</i>	80
4.2 Implementasi Desain Antarmuka	82
4.2.1 Halaman <i>Login</i>	82
4.2.2 Halaman <i>Rules</i>	82
4.2.3 Halaman <i>Register</i>	82
4.2.4 Halaman <i>Homepage</i>	83
4.2.5 Halaman <i>Browse</i>	83
4.2.6 Halaman <i>Profile</i>	84
4.2.7 Halaman <i>Detail Post</i>	84
4.2.8 Halaman Dashboard Admin	85
4.2.9 Halaman Tabel <i>Users</i>	85
4.2.10 Halaman Tabel <i>Posts</i>	85
4.2.11 Halaman Tabel <i>Replies</i>	86
4.2.12 Halaman Tabel <i>Tags</i>	86
4.2.13 Halaman Tabel <i>Reports</i>	87
4.2.14 Halaman Tabel <i>Good & Bad Words</i>	87
4.2.15 Halaman Tabel <i>Common Words</i>	88
4.3 <i>Deployment</i>	88
4.4 Pengujian	88
4.4.1 <i>Black Box Testing</i>	88
4.4.2 <i>User Acceptance Testing (UAT)</i>	90
4.4.3 Hasil Algoritma <i>Boyer-Moore</i>	91
4.4.4 Hasil <i>Text to Text Transfer Transformer Models</i>	92
BAB V. PENUTUP	93
5.1 Kesimpulan	93
5.2 Saran	93
DAFTAR PUSTAKA	94
DAFTAR RIWAYAT HIDUP	97
DAFTAR LAMPIRAN	98
Lampiran 1. Surat Hasil Validasi	98
Lampiran 2. Wawancara Dengan Pak Dr. Ryan Hidayat, M.Pd.	99
Lampiran 3. Hasil Kuesioner	100

Lampiran 4. Hasil UAT	103
Lampiran 5. kode Algoritma.....	107
Lampiran 6. kode AI <i>Pretrained</i>	115
Lampiran 7. Daftar Kata.....	116

DAFTAR TABEL

Tabel 2.1 Flowchart	11
Tabel 2.2 <i>Simbol Use Case Diagram</i>	14
Tabel 2.3 Simbol ERD.....	15
Tabel 2.4 Simbol <i>Activity Diagram</i>	17
Tabel 2.5 <i>Sequence Diagram</i>	18
Tabel 2.6 Simbol <i>Class Diagram</i>	20
Tabel 2.7 Penelitian Terdahulu	21
Tabel 3.1 Tabel kata kurang sopan dan sopan	67
Tabel 3.2 Tabel <i>OH</i> dan <i>MH</i>	68
Tabel 3.3 Tabel pergeseran langkah satu	68
Tabel 3.4 Tabel pergeseran langkah dua.....	69
Tabel 3.5 Jadwal Penelitian	70

DAFTAR GAMBAR

Gambar 3.1 Tahapan Penelitian.....	24
Gambar 3.2 <i>Login Screen Wireframe</i>	26
Gambar 3.3 <i>Register Screen Wireframe</i>	26
Gambar 3.4 <i>Timeline Screen Wireframe</i>	26
Gambar 3.5 <i>Post Screen Wireframe</i>	27
Gambar 3.6 <i>Dashboard Admin Screen Wireframe</i>	27
Gambar 3.7 <i>Data User Post Screen</i>	27
Gambar 3.8 <i>Data Words Screen Wireframe</i>	28
Gambar 3.9 <i>ERD</i>	29
Gambar 3.10 <i>Use Case Diagram</i>	31
Gambar 3.11 <i>Activity Diagram Register</i>	32
Gambar 3.12 <i>Activity Diagram Login</i>	33
Gambar 3.13 <i>Activity Diagram Create Post</i>	33
Gambar 3.14 <i>Activity Diagram Delete Own Post</i>	34
Gambar 3.15 <i>Activity Diagram Add Likes</i>	34
Gambar 3.16 <i>Activity Diagram Add Reply</i>	35
Gambar 3.17 <i>Activity Diagram Report Post</i>	35
Gambar 3.18 <i>Activity Diagram Update Profile</i>	36
Gambar 3.19 <i>Activity Diagram Search Users/Posts</i>	36
Gambar 3.20 <i>Activity Diagram Create Users</i>	37
Gambar 3.21 <i>Activity Diagram Read Data</i>	37
Gambar 3.22 <i>Activity Diagram Update Users</i>	38
Gambar 3.23 <i>Activity Diagram Delete Data</i>	38
Gambar 3.24 <i>Activity Diagram Create Tags</i>	39
Gambar 3.25 <i>Activity Diagram Update Tags</i>	40
Gambar 3.26 <i>Activity Diagram Create Good Word</i>	40
Gambar 3.27 <i>Activity Diagram Update Good Words</i>	41
Gambar 3.28 <i>Activity Diagram Create Bad Words</i>	42
Gambar 3.29 <i>Activity Diagram Update Good Words</i>	43
Gambar 3.30 <i>Activity Diagram Delete Bad Words</i>	44
Gambar 3.31 <i>Activity Diagram Create Common Words</i>	44
Gambar 3.32 <i>Activity Diagram Update Common Words</i>	45
Gambar 3.33 <i>Activity Diagram Notification</i>	45
Gambar 3.34 <i>Sequence Diagram Register</i>	46
Gambar 3.35 <i>Sequence Diagram Login</i>	47
Gambar 3.36 <i>Sequence Diagram Create Post</i>	48
Gambar 3.37 <i>Sequence Diagram Delete Own Post</i>	49
Gambar 3.38 <i>Sequence Diagram Add Likes</i>	49
Gambar 3.39 <i>Sequence Diagram Add Reply</i>	50
Gambar 3.40 <i>Sequence Diagram Report Post</i>	51
Gambar 3.41 <i>Sequence Diagram Update Profile</i>	51
Gambar 3.42 <i>Sequence Diagram Search Users</i>	52

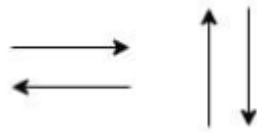
Gambar 3.43 <i>Sequence Diagram Read Data</i>	52
Gambar 3.44 <i>Sequence Diagram Create Users</i>	53
Gambar 3.45 <i>Sequence Diagram Update Users</i>	53
Gambar 3.46 <i>Sequence Diagram Delete Data</i>	54
Gambar 3.47 <i>Sequence Diagram Create Tags</i>	54
Gambar 3.48 <i>Sequence Diagram Update Tags</i>	55
Gambar 3.49 <i>Sequence Diagram Create Good Words</i>	55
Gambar 3.50 <i>Sequence Diagram Update Good Words</i>	56
Gambar 3.51 <i>Sequence Diagram Create Bad Words</i>	56
Gambar 3.52 <i>Sequence Diagram Update Bad Words</i>	57
Gambar 3.53 <i>Sequence Diagram Delete Bad Words</i>	58
Gambar 3.54 <i>Sequence Diagram Create Common Words</i>	58
Gambar 3.55 <i>Sequence Diagram Update Common Words</i>	59
Gambar 3.56 <i>Sequence Diagram Notification</i>	59
Gambar 3.57 <i>Class Diagram</i>	62
Gambar 3.58 Model Filter Teks	63
Gambar 3.59 <i>The Transformer-model Architecture</i>	64
Gambar 4.1 Contoh data tabel <i>goodWords</i>	72
Gambar 4.2 Contoh data tabel <i>badWords</i>	72
Gambar 4.3 Alur Kerja Algoritma	72
Gambar 4.4 Fungsi <i>normalizeWord()</i>	73
Gambar 4.5 Fungsi <i>getNormalizationMapping()</i>	73
Gambar 4.6 Fungsi <i>getBadWordsFromDB()</i>	74
Gambar 4.7 Inisialisasi Set Kata Umum	75
Gambar 4.8 Fungsi <i>buildBadCharacterTable()</i>	75
Gambar 4.9 Fungsi <i>BoyerMooreSearch()</i>	76
Gambar 4.10 Fungsi <i>buildFuzzyRegex()</i>	76
Gambar 4.11 Penanganan <i>Overlap</i> dan menyatukan kembali teks	77
Gambar 4.12 Implementasi <i>Boyer-Moore</i> di Pembuatan <i>Post</i>	78
Gambar 4.13 Implementasi <i>Boyer-Moore</i> di Pembuatan <i>Reply</i>	78
Gambar 4.14 Implementasi <i>Boyer-Moore</i> Saat Pengetikan <i>Post</i> dan <i>Reply</i>	79
Gambar 4.15 Implementasi <i>CheckWord</i> Di <i>Frontend</i>	80
Gambar 4.16 Kode <i>Text to Text Transfer Transformer Models</i>	81
Gambar 4.17 Halaman <i>Login</i>	82
Gambar 4.18 Halaman <i>Rules</i>	82
Gambar 4.19 Halaman <i>Register</i>	83
Gambar 4.20 Halaman <i>Homepage</i>	83
Gambar 4.21 Halaman <i>Browse</i>	84
Gambar 4.22 Halaman <i>Profile</i>	84
Gambar 4.23 Halaman <i>Detail Post</i>	84
Gambar 4.24 Halaman <i>Dashboard Admin</i>	85
Gambar 4.25 Halaman Tabel <i>Users</i>	85
Gambar 4.26 Halaman Tabel <i>Posts</i>	86
Gambar 4.27 Halaman Tabel <i>Replies</i>	86

Gambar 4.28 Halaman Tabel <i>Tags</i>	87
Gambar 4.29 Halaman Tabel <i>Reports</i>	87
Gambar 4.30 Halaman Tabel <i>Good & Bad Words</i>	87
Gambar 4.31 Halaman Tabel <i>Common Words</i>	88
Gambar 4.32 Hasil <i>Deployment</i>	88
Gambar 4.33 Hasil <i>Text to Text Transfer Transformer</i>	92

DAFTAR SIMBOL

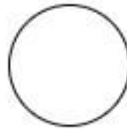
1. *Flowchart*

a. *Flow*



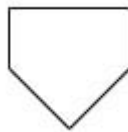
Flow adalah penghubung antar simbol dalam diagram (disebut juga *Connecting Line*).

b. *On-Page Reference*



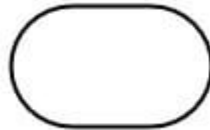
On-Page Reference adalah penanda sambungan proses dalam lembar kerja yang sama.

c. *Off-Page Reference*



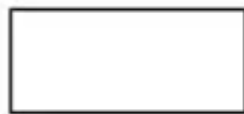
Off-Page Reference adalah Penanda sambungan proses dalam lembar kerja yang berbeda.

d. *Terminator*



Terminator menunjukkan awal atau akhir dari suatu program atau proses.

e. *Process*



Process adalah langkah atau proses yang dilakukan oleh komputer.

f. *Decision*



Decision adalah Menunjukkan percabangan keputusan dengan dua kemungkinan, yaitu ya atau tidak.

g. *Input/Output*



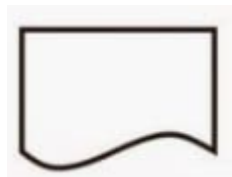
Input/Output adalah menunjukkan aktivitas *input* atau *output* tanpa memperhatikan perangkat yang digunakan.

h. *Manual Operation*



Manual Operation menunjukkan proses yang dilakukan secara manual, bukan oleh komputer.

i. *Document*



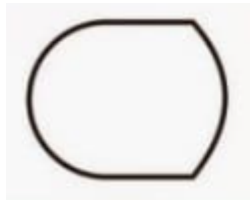
Document menyatakan bahwa input berasal dari atau *output* menuju dokumen fisik.

j. *Predefined Process*



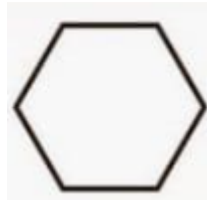
Predefined Process adalah proses terdefinisi seperti subprogram atau prosedur.

k. *Display*



Display menunjukkan penggunaan alat *output*, seperti layar monitor.

l. *Preparation*



Preparation menunjukkan penyiapan atau inisialisasi nilai awal sebelum proses berjalan.

2. *Use Case Diagram*

a. *Actors*



Actors merupakan entitas yang memiliki hubungan atau melakukan interaksi dengan suatu sistem.

b. *Use Case*



Use case merupakan aktivitas atau fungsi yang dapat dilakukan oleh *aktor* saat berinteraksi dengan sistem.

c. *Accosiation*



Accosiation merupakan hubungan atau keterkaitan antara aktor dan *use case* di dalam suatu sistem.

d. *Sistem*



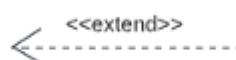
Sistem adalah batasan atau ruang lingkup yang mengatur area pengembangan dari sistem yang sedang dibangun.

e. *Include*



Include menunjukkan bahwa suatu *use case* sepenuhnya merupakan bagian dari fungsionalitas yang dimiliki oleh *use case* lainnya, dan selalu dijalankan sebagai bagian dari proses utama.

f. *Extend*



Extend menunjukkan bahwa suatu *use case* merupakan

tambahan fungsionalitas dari *use case* lain, yang akan dijalankan apabila kondisi tertentu terpenuhi.

3. *Entity Relationship Diagram* (ERD)

a. Entitas ERD



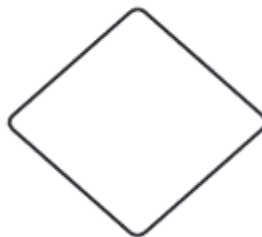
Entitas adalah objek yang memiliki wujud dan dapat dibedakan dari objek lainnya. Entitas ini bisa bersifat konkret maupun abstrak. Contoh entitas konkret meliputi orang, buku, pegawai, dan perusahaan, sedangkan entitas abstrak mencakup mata kuliah, kejadian, dan pekerjaan.

b. Atribut ERD



Atribut menggambarkan ciri atau karakteristik dari sebuah entitas. Sebagai contoh, entitas karyawan bisa memiliki atribut seperti nama, alamat, dan NIP. Dalam basis data, atribut direpresentasikan sebagai kolom atau *field* pada tabel.

c. Relasi ERD



Relasi menunjukkan keterkaitan antara dua atau lebih entitas. Sebagai contoh, relasi "menjual" menghubungkan entitas "penjual" dengan "produk." Dalam implementasi basis data, relasi biasanya direpresentasikan sebagai *foreign key* yang menghubungkan tabel.

d. Partisipasi Opsional ERD



Partisipasi opsional dalam relasi menunjukkan bahwa entitas di salah satu sisi

e. Partisipasi Wajib ERD



f. Relasi *Many* ERD



g. Relasi *Zero or Many* ERD



h. Relasi *One or Many* ERD



i. Relasi *One and Only* ERD



relasi tidak harus terlibat dalam relasi tersebut. Dengan kata lain, keterlibatan entitas dalam relasi bersifat tidak wajib atau opsional.

Menunjukkan bahwa partisipasi dalam relasi bersifat wajib, artinya entitas di salah satu sisi relasi harus terlibat dalam relasi tersebut.

Menunjukkan bahwa setiap entitas di salah satu sisi relasi dapat terhubung dengan banyak entitas lainnya.

Menunjukkan bahwa entitas dapat tidak memiliki hubungan atau memiliki banyak hubungan dengan entitas lain.

Menunjukkan bahwa entitas wajib memiliki banyak hubungan dengan entitas lain, atau setidaknya harus berhubungan dengan satu entitas atau lebih.

Menunjukkan bahwa entitas hanya dapat berhubungan dengan satu entitas lain, tanpa dapat berhubungan lebih dari itu.

4. Activity Diagram

a. *Initial State*



b. Aktivitas



Menandakan awal dari aliran aktivitas

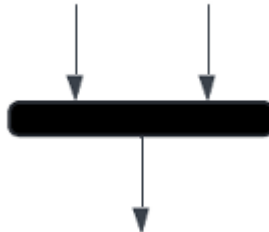
Aktivitas yang dilakukan oleh sistem biasanya dimulai dengan kata kerja.

c. Percabangan



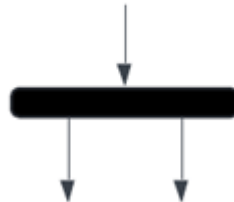
Menunjukkan pengambilan keputusan. Dari *node* ini, alur dapat bercabang menjadi beberapa jalur berdasarkan kondisi tertentu.

d. *Joint Node*



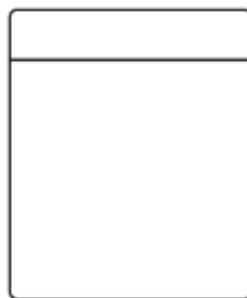
Menggabungkan beberapa alur bercabang kembali menjadi satu alur.

e. *Fork Node*



Membagi satu alur menjadi beberapa alur paralel yang dapat berjalan secara bersamaan.

f. *Swimlane*



Memisahkan aktivitas berdasarkan aktor atau departemen yang bertanggung jawab.

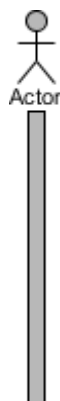
g. *Final State*



Menandakan akhir dari aliran aktivitas.

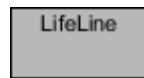
5. *Sequence Diagram*

a. *Actors*



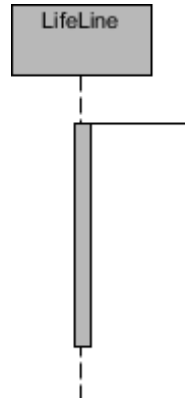
Actors merupakan representasi dari peran yang dimainkan oleh entitas di luar sistem yang sedang dimodelkan. Aktor berinteraksi dengan sistem melalui pertukaran sinyal dan data. Entitas ini bisa berupa pengguna manusia, perangkat keras eksternal, maupun sistem lain.

b. *Lifeline*



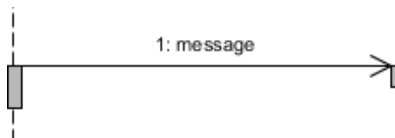
Lifeline (garis kehidupan) merepresentasikan satu partisipan dalam suatu interaksi. *Lifeline* menggambarkan keberadaan suatu entitas selama berlangsungnya interaksi dalam rentang waktu tertentu.

c. *Activations*



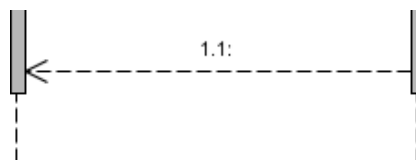
Sebuah persegi panjang tipis pada *lifeline* menunjukkan periode ketika suatu elemen sedang menjalankan sebuah operasi. Bagian atas dan bawah dari persegi panjang tersebut menunjukkan waktu dimulainya dan selesainya operasi secara berurutan.

d. *Call Message*



Pesan panggilan (*call message*) adalah jenis pesan yang merepresentasikan pemanggilan suatu operasi pada *lifeline* target.

e. *Return Message*



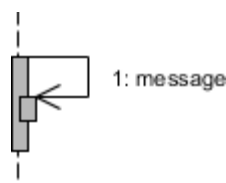
Pesan *return* (pengembalian) merepresentasikan pengiriman informasi kembali kepada pemanggil dari pesan sebelumnya yang sesuai.

f. *Self Message*



Pesan diri (*self message*) merupakan pemanggilan suatu operasi oleh sebuah objek terhadap dirinya sendiri dalam *lifeline* yang sama.

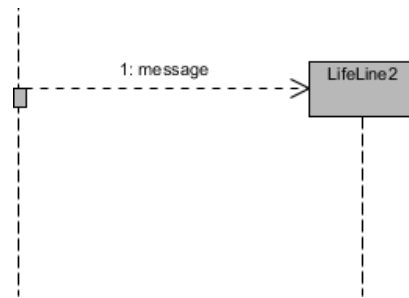
g. *Recursive Message*



Pesan rekursif (*recursive message*) adalah pemanggilan pesan pada *lifeline* yang sama, di mana

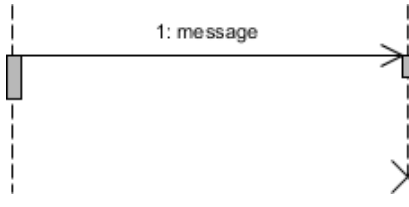
pesan tersebut mengarah pada aktivasi yang berada di atas aktivasi asal pemanggilan. Ini menunjukkan bahwa operasi memanggil dirinya sendiri secara berulang dalam satu alur eksekusi.

h. *Create Message*



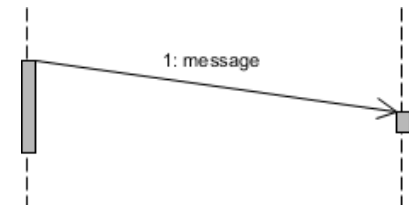
Pesan *create* merepresentasikan proses instansiasi atau penciptaan objek baru pada *lifeline* target.

i. *Destroy Message*



Pesan *destroy* merepresentasikan permintaan untuk mengakhiri atau menghentikan siklus hidup dari *lifeline* target.

j. *Duration Message*



Pesan *duration* menunjukkan jarak atau selang waktu antara dua titik waktu dalam suatu pemanggilan pesan.

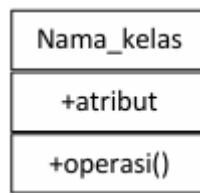
k. *Note*



Catatan atau komentar digunakan untuk menambahkan berbagai keterangan pada elemen-elemen dalam diagram. Meskipun tidak memiliki makna semantik terhadap model, catatan ini berisi informasi yang berguna bagi para pemodel.

6. Class Diagram

a. Kelas



Kelas tersebut menunjukkan notasi dasar *class diagram*, yang terdiri dari tiga bagian: nama kelas, atribut (dengan visibilitas publik), dan operasi atau metode yang juga bersifat publik. Diagram ini digunakan untuk merepresentasikan struktur dan fungsi dari sebuah kelas dalam sistem.

b. Association



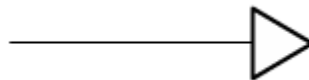
Relasi antar kelas menunjukkan hubungan antar objek. Asosiasi adalah relasi umum yang sering disertai *multiplicity* untuk menyatakan jumlah objek yang terlibat dalam hubungan tersebut.

c. Directed Association



Relasi antar kelas menunjukkan bahwa satu kelas menggunakan atau bergantung pada kelas lain. Relasi ini disebut asosiasi dan biasanya dilengkapi dengan *multiplicity* untuk menunjukkan jumlah objek yang terlibat.

d. Generalisasi



Relasi generalisasi-spesialisasi menggambarkan hubungan antara kelas umum (*superclass*) dan kelas khusus (*subclass*). *Subclass* mewarisi atribut dan operasi dari *superclass*, mencerminkan prinsip pewarisan dalam pemrograman.

e. Dependency



Relasi kebergantungan menunjukkan bahwa satu kelas bergantung pada kelas lain untuk menjalankan

f. *Aggregation*



fungsinya. Perubahan pada kelas yang digunakan dapat memengaruhi kelas yang bergantung padanya.

Relasi *whole-part* atau **agregasi** menunjukkan hubungan bahwa satu kelas merupakan keseluruhan (*whole*) yang terdiri dari bagian-bagian (*part*) berupa kelas lain. Bagian dapat tetap ada meskipun keseluruhannya dihapus.

DAFTAR RUMUS

Rumus 2.1 <i>String Matching</i>	5
Rumus 2.2 <i>String Matching</i>	5
Rumus 2.3 Nilai <i>Occurrence Heuristic</i>	6
Rumus 4.1 Evaluasi Manual.....	91