BAB III

ANALISIS DAN PERANCANGAN SISTEM

Analisis dan perancangan sistem adalah tahapan sistematis untuk mendapatkan aplikasi yang baik dan bersesuaian dengan kegunaan dan tujuannya. Tahap awal dari analisis adalah menganalisis kebutuhan-kebutuhan sistem mulai dari kebutuhan pengguna, kebutuhan non fungsional dan kebutuhan fungsional. Sedangkan untuk tahapan perancangan aplikasi yaitu perancangan struktur kodefikasi, perancangan sistem dan perancangan antarmuka.

3.1. Analisis Sistem

Analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat sesuai dengan kebutuhan. Dalam membangun perangkat lunak ini dilakukan beberapa analisis yaitu:

1. Analisis Masalah
2. Analisis Algoritma
3. Analisis Kebutuhan Non Fungsional
4. Analisis Kebutuhan Fungsional

3.1.1. Analisis Masalah

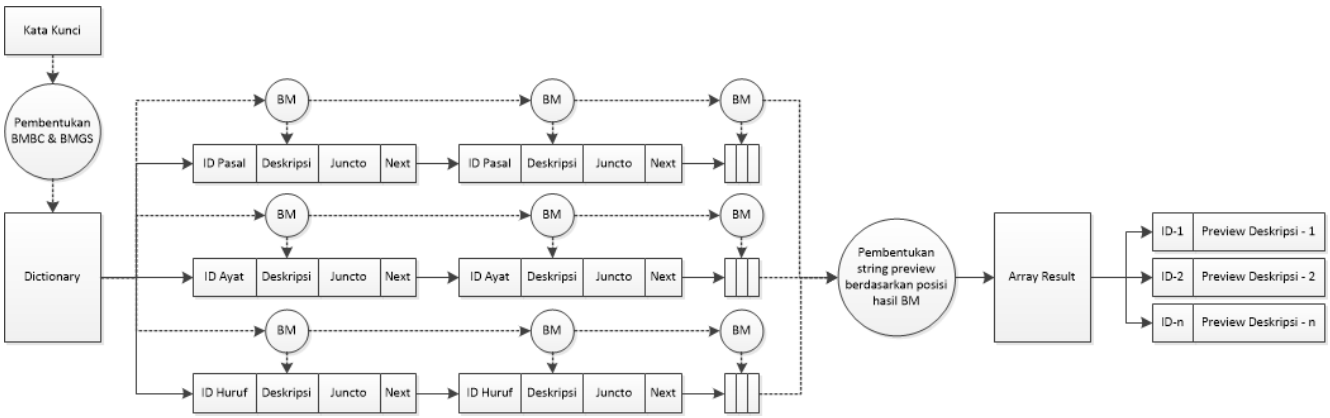
Dari hasil pengamatan diketahui bahwa kurangnya media informasi tentang aturan berlalu-lintas yang berakibat kurang kepekaan masyarakat akan aturan lalu-lintas, termasuk pada Undang-undang Lalu-lintas dan Angkutan Jalan (UU No 22 Tahun 2009), yang berdampak maraknya pelanggaran maupun korban kecelakaan lalu-lintas. Contoh kasus pelanggaran pengendara sepeda motor di jalan yang tidak memenuhi persyaratan teknis tidak mengenakan kaca spion, klakson lampu utama dan lain-lainnya seperti yang telah ditetapkan aturan

sebgaimana dalam Pasal 285 UU No 22 Tahun 2009 akan dikenakan pidana dengan kurungan paling lama satu bulan atau denda paling banyak Rp250.000 .

Oleh sebab itu aplikasi ini bisa membantu masyarakat untuk mengetahui pasal mana atau hukuman apa saja yang dikenakan apabila melanggar peraturan lalu-lintas yang dilakukan dengan efektif dan cepat tanpa harus mengakses informasi dari internet ataupun membuka buku, dengan menggunakan algoritma pencocokan string sebagai metode penyelesaian masalah pencarian.

3.1.2. Deskripsi Sistem

Aplikasi UU Lalu lintas adalah sebuah aplikasi mobile pada perangkat *smartphone* berbasis Android. Desripsi Sistem aplikasi sistem bisa dilihat pada gambar 3.1 berikut :



Gambar 3.1 Diagram blok gambaran umum sistem pencarian

1) Kata Kunci adalah *keyword* yang diketikan oleh user.
2) Pembentukan BMBC & BMGS adalah proses pembuatan nilai pencarian.
3) Dictionary adalah class yang merepresentasikan data dari ID Pasal, ID ayat dan ID Huruf.
4) ID Pasal adalah sebuah kodefikasi untuk menunjukan alamat atau posisi pada data teks Pasal.

5) ID Ayat adalah sebuah kodefikasi untuk menunjukan alamat atau posisi pada data teks Ayat.

6) ID Huruf adalah sebuah kodefikasi untuk menunjukan alamat atau posisi pada data teks Huruf.

7) Juncto adalah sebuah kodefikasi yang terdapat pada badan pasal atau pun ayat untuk menunjukan alamat atau posisi keterkaitan antara satu Pasal dengan dengan Pasal lainya, namun tidak semua Pasal ataupun ayat terdapat Juncto.

8) BM adalah proses Algoritma Boyer-Moore.

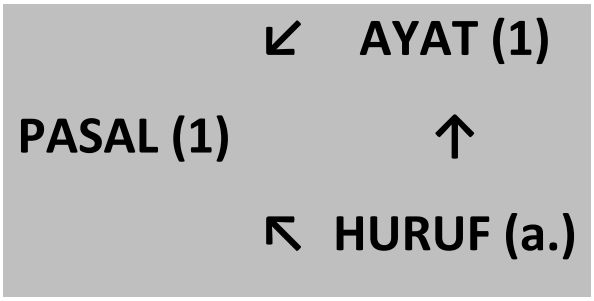
9) String Preview adalah hasil dari representasi hasil pencarian BM alamat atau posisi pada teks.

10) Array Result adalah larik yang dari hasil pencarian yang akan ditampilkan

11) ID & Preview Deskripsi adalah data yang akan ditampilkan dari hasil pencarian yang diambil dari Array Result.

3.1.3. Analisis Arsitektur Badan Pasal

Dalam perancangan Arsitektur badan pasal yang dilakukan pertama kali adalah membagi/meng kategorikan atau memecah pasal-pasal yang terdapat pada sumber dengan asumsi adalah sebuah pasal terdiri dari ayat, huruf atau ayat dan huruf. Sebuah ayat terdiri dari huruf maupun tidak dan sebuah huruf dapat berdiri sendiri. Maksud dan tujuan pembagian ini untuk membuat kodefikasi pada teks yang akan dijadikan sumber pencarian.



Gambar 3.2 Skema asumsi pembagian pasal

Seperti yang telah dijelaskan sebelumnya yaitu pemecahan pasal menjadi bagian-bagian di contohkan seperti dibawah ini:

1) Pasal terdiri dari hanya isi pasal atau deskripsi pasal tanpa ayat maupun huruf. Contoh nya :

   Pasal 281

   Setiap orang yang mengemudikan Kendaraan Bermotor diJalan yang tidak memiliki Surat Izin Mengemudi sebagaimana dimaksud dalam Pasal 77 ayat (1) dipidana dengan pidana kurungan paling lama 4 (empat) bulan atau denda paling banyak Rp1.000.000,00 (satu juta rupiah).

2) Pasal terdiri dari ayat saja tanpa deskripsi maupun huruf. Contoh nya :

   Pasal 131

   (1) Pejalan Kaki berhak atas ketersediaan fasilitas pendukung yang berupa trotoar, tempat penyeberangan, dan fasilitas lain.
   (2) Pejalan Kaki berhak mendapatkan prioritas pada saat menyeberang Jalan di tempat penyeberangan.
   (3) Dalam hal belum tersedia fasilitas sebagaimana dimaksud pada ayat (1), Pejalan Kaki berhak menyeberang di tempat yang dipilih dengan memperhatikan keselamatan dirinya.

3) Pasal terdiri dari huruf saja tanpa ayat. Contoh nya :

   Pasal 160

   Angkutan barang dengan Kendaraan Bermotor Umum terdiri atas:

   a. angkutan barang umum; dan
   b. angkutan barang khusus.

4) Pasal terdiri dari ayat dan huruf. Contoh nya :

   Pasal 173

   (1) Perusahaan Angkutan Umum yang menyelenggarakan angkutan orang dan/atau barang wajib memiliki:

a. izin penyelenggaraan angkutan orang dalam trayek;

b. izin penyelenggaraan angkutan orang tidak dalam trayek; dan/atau

c. izin penyelenggaraan angkutan barang khusus atau alat berat.

(2) Kewajiban memiliki izin sebagaimana dimaksud pada ayat (1) tidak berlaku untuk:

a. pengangkutan orang sakit dengan menggunakan ambulans; atau

b. pengangkutan jenazah.

Pemecahan atau pengelempokan bagian-bagian pasal ini bertujuan untuk membuat kodefikasi untuk data pencarian pasal yang lebih efektif dan efisien, karena kumpulan pasal dalam aplikasi ini berupa kumpulan teks yang dikodefikasi yang bertujuan untuk pemisah antara isi pasal,ayat dan huruf. Sehingga dalam aplikasi ini tidak memerlukan database sehingga kapasitas *memory* yang digunakan akan relatif kecil.

3.1.4. Analisis Kodefikasi

General ID merupakan kodefikasi yang dibuat khusus sebagai pemisah antara bagian-bagian baik itu pasal,ayat maupun huruf. General ID dibuat dengan tujuan sebgai pemisah dan juga sebgai pengalamatan posisi teks pada data teks keseluruhan yang nantinya terdapat pada file Pasal.txt, Ayat.txt dan Huruf.txt yang difungsikan sebagai kumpulan data UU No.22 Tahun 2009.

Hipotesis dalam pembuatan kodefikasi pada teks pasal ini adalah semakin panjang *pattern*, semakin bervariatif (tidak ada kesamaan karakter) pada *pattern* dan pattern General ID dirancang sedemikian rupa karena susunan karakternya tidak terdapat pada teks UU No.22 Tahun 2009 maka proses pencarian yang menggunakan algoritma Boyer-Moor akan semakin cepat, namun untuk mengefisiensikan penggunaan *memory* maka kodefikasi dibatasi panjang hanya 21

karakter, dengan bervariatif yaitu tidak ada kesamaan urutan dan jenis karakter pada kodefikasi maupun teks.

Tidak hanya dari segi panjang *pattern* dan bervariatf nya karakter pada pattern namun pada kodefikasi pasal, ayat, huruf ataupun juncto mempunyai aturan yaitu *General ID* yaitu kodefikasi yang berbeda-beda bertujuan untuk membedakan antar kodefikasi yang menunjuk untuk pasal, ayat maupun huruf contoh nya :

1. General ID untuk Pasal adalah : 1Q2W3E4R5T6Y7U8I9O
2. General ID untuk Ayat adalah : 1A2S3D4F5G6H7J8K9L
3. General ID untuk Huruf adalah : 1Z2X3C4V5B6N7M8P9#
4. General ID untuk Juncto adalah : [okmijnuhbygvtfcrdx]

Setelah menentukan general id maka akan di ikuti kode penunjuk pasal, ayat maupun huruf sesuai dengan jenis kodefikasi nya, seperti contoh :

1. Kodefikasi Pasal :
   General ID + No Pasal (3 Digit).
   1Q2W3E4R5T6Y7U8I9O + 001 (1Q2W3E4R5T6Y7U8I9O001) adalah contoh kodefikasi untuk Pasal 1. No Pasal dirancang 3 Digit karena pada UU No. 22 Tahun 2009 jumlah Pasal tidak lebih dari 3 Digit.
2. Kodefikasi Ayat :
   General ID + No Ayat (2 Digit) + No Pasal (3 Digit).
   1A2S3D4F5G6H7J8K9L + 01 + 001 (1A2S3D4F5G6H7J8K9L01001) adalah contoh untuk Pasal 1 ayat 1. No Ayat dirancang 2 Digit karena pada setiap Pasal yang memiliki Ayat jumlah Ayat nya tidak lebih dari 2 Digit, sedangkan No Pasal dirancang 3 Digit karena pada UU No. 22 Tahun 2009 jumlah Pasal tidak lebih dari 3 Digit.
3. Kodefikasi Huruf :
   General ID + No huruf (1 Digit) + No Ayat (2 Digit) + No Pasal (3 Digit).

1Z2X3C4V5B6N7M8P9# + a + 01 + 001 (1Z2X3C4V5B6N7M8P9# a01001) adalah Pasal 1 ayat 1 huruf a. No huruf dirancang hanya 1 digit karena huruf hanya sebagai penjabaran suatu Pasal atau Ayat, No Ayat dirancang 2 Digit karena pada setiap Pasal yang memiliki Ayat jumlah Ayat nya tidak lebih dari 2 Digit, sedangkan No Pasal dirancang 3 Digit karena pada UU No. 22 Tahun 2009 jumlah Pasal tidak lebih dari 3 Digit.

4. Kodefikasi Juncto :

   General ID + No Huruf (1 Digit) + No Ayat (2 Digit) + No Pasal (3 Digit).

   [okmijnuhbygvtfcrdx] + a + 01 + 231 ([okmijnuhbygvtfcrdx]a01231) adalah Pasal 231 ayat 1 huruf a. No huruf dirancang hanya 1 digit karena huruf hanya sebagai penjabaran suatu Pasal atau Ayat, No Ayat dirancang 2 Digit karena pada setiap Pasal yang memiliki Ayat jumlah Ayat nya tidak lebih dari 2 Digit, sedangkan No Pasal dirancang 3 Digit karena pada UU No. 22 Tahun 2009 jumlah Pasal tidak lebih dari 3 Digit.

Untuk lebih jelasnya kodefikasi beserta Pasal di aplikasikan pada Tabel 3.1 Deskripsi Pasal :

Tabel 3.1 Deskripsi Pasal

Nama Field	Deskripsi
ID	Kodefikasi untuk menunjuk sebuah pasal tertentu.
Deskripsi	Pengertian pasal

Pada tabel pasal ini menjelaskan ID (General ID + No Pasal) adalah sebgai Kodefikasi untuk pasal dan pada deskripsi adalah untuk menjelaskan pengertian atau isi pasal tersebut. Sebgaimana dicontohkan pada Pasal 7, Pasal 8 dan Pasal 10 berikut :

Tabel 3.2 Implementasi Kodefikasi Pasal

ID	Deskripsi
1Q2W3E4R5T6Y7U8I9O007	<tidak ada deskripsi>
1Q2W3E4R5T6Y7U8I9O008	Penyelenggaraan di bidang Jalan meliputi kegiatan pengaturan, pembinaan, pembangunan, dan pengawasan prasarana Jalan sebagaimana dimaksud dalam Pasal 7 ayat (2) huruf a, yaitu:
1Q2W3E4R5T6Y7U8I9O010	Penyelenggaraan di bidang industri sebagaimana dimaksud dalam Pasal 7 ayat (2) huruf c meliputi:

Lalu implementasi pada teks nya adalah dengan kodefikasi untuk pasal adalah sebagai berikut:

1Q2W3E4R5T6Y7U8I9O0071Q2W3E4R5T6Y7U8I9O008Penyelenggaraan di bidang Jalan meliputi kegiatan pengaturan, pembinaan, pembangunan, dan pengawasan prasarana Jalan sebagaimana dimaksud dalam Pasal 7 ayat (2) huruf a, yaitu: 1Q2W3E4R5T6Y7U8I9O010Penyelenggaraan di bidang industri sebagaimana dimaksud dalam Pasal 7 ayat (2) huruf c meliputi:

Gambar 3.3 Implementasi Kodefikasi Pasal pada Teks

Dilanjutkan dengan kodefikasi ayat yang akan dijelaskan seperti berikut :

Tabel 3.3 Deskripsi Ayat

Nama Field	Deskripsi
ID	Kodefikasi untuk menunjuk sebuah ayat pada pasal tertentu.
Deskripsi	Pengertian ayat

Pada tabel ayat ini menjelaskan ID (General ID + No Ayat (2 Digit) + No Pasal (3 Digit)) adalah sebgai Kodefikasi untuk ayat dan sebgai penunjuk sebuah ayat pada pasal tertentu, deskripsi adalah penjelasan isi dari ayat tersebut. Sebgaimana dicontohkan pada Pasal 21 berikut :

Pasal 21

(1) Setiap Jalan memiliki batas kecepatan paling tinggi yang ditetapkan secara nasional.

(2) Batas kecepatan paling tinggi sebagaimana dimaksud pada ayat (1) ditentukan berdasarkan kawasan permukiman, kawasanperkotaan, jalan antarkota, dan jalan bebas hambatan.

(3) Atas pertimbangan keselamatan atau pertimbangan khusus lainnya, Pemerintah Daerah dapat menetapkan batas kecepatan paling tinggi setempat yang harus dinyatakan dengan Rambu Lalu Lintas.

(4) Batas kecepatan paling rendah pada jalan bebas hambatan ditetapkan dengan batas absolut 60 (enam puluh) kilometer per jam dalam kondisi arus bebas.

(5) Ketentuan lebih lanjut mengenai batas kecepatan sebagaimana dimaksud pada ayat (1) dan ayat (2) diatur dengan peraturan pemerintah.

Tabel 3.4 Implementasi Kodefikasi Ayat

ID	Description
1A2S3D4F5G6H7J8K9L01021	Setiap Jalan memiliki batas kecepatan paling tinggi yang ditetapkan secara nasional.
1A2S3D4F5G6H7J8K9L02021	Batas kecepatan paling tinggi sebagaimana dimaksud pada ayat (1) ditentukan berdasarkan kawasan permukiman, kawasan perkotaan, jalan antarkota, dan jalan bebas hambatan.
1A2S3D4F5G6H7J8K9L03021	Atas pertimbangan keselamatan atau pertimbangan khusus lainnya, Pemerintah Daerah dapat menetapkan batas kecepatan paling tinggi setempat yang harus dinyatakan dengan Rambu Lalu Lintas
1A2S3D4F5G6H7J8K9L04021	Batas kecepatan paling rendah pada jalan bebas hambatan ditetapkan dengan batas absolut 60 (enam puluh) kilometer per jam dalam kondisi arus bebas.
1A2S3D4F5G6H7J8K9L05021	Ketentuan lebih lanjut mengenai batas kecepatan sebagaimana dimaksud pada ayat (1) dan ayat (2) diatur dengan peraturan pemerintah.

Lalu implementasi pada teks nya adalah dengan kodefikasi untuk pasal adalah sebagai berikut:

1A2S3D4F5G6H7J8K9L01021Setiap Jalan memiliki batas kecepatan paling tinggi yang ditetapkan secara nasional.1A2S3D4F5G6H7J8K9L02021Batas kecepatan paling tinggi sebagaimana dimaksud pada ayat (1) ditentukan berdasarkan kawasan permukiman, kawasan perkotaan, jalan antarkota, dan jalan bebas hambatan.1A2S3D4F5G6H7J8K9L03021Atas pertimbangan keselamatan atau pertimbangan khusus lainnya, Pemerintah Daerah dapat menetapkan batas kecepatan paling tinggi setempat yang harus dinyatakan dengan Rambu Lalu Lintas.1A2S3D4F5G6H7J8K9L04021Batas kecepatan paling rendah pada jalan bebas hambatan ditetapkan dengan batas absolut 60 (enam puluh) kilometer per jam dalam kondisi arus bebas. 1A2S3D4F5G6H7J8K9L05021Ketentuan lebih lanjut mengenai batas kecepatan sebagaimana dimaksud pada ayat (1) dan ayat (2) diatur dengan peraturan pemerintah.

Gambar 3.4 Implementasi Kodefikasi Ayat pada Teks

Dilanjutkan dengan kodefikasi huruf yang akan dijelaskan seperti pada Tabel 3.7 berikut :

Tabel 3.5 Deskripsi Huruf

Nama Field	Deskripsi
ID	Kodefikasi untuk menunjuk sebuah HURUF pada pasal tertentu. Jika HURUF tidak Memiliki AYAT maka KODE AYAT = 00
Deskripsi	Pengertian Huruf

Pada tabel huruf ini menjelaskan ID (General ID + No huruf (1 Digit) + No Ayat (2 Digit) + No Pasal (3 Digit)), adalah kodefikasi untuk menunjuk kepada sebuah huruf pada pasal tertentu dan deskripsi adalah penjelasan isi dari huruf tersebut. Sebgaimana dicontohkan pada Pasal 74 berikut ini :

Pasal 74

(1) Kendaraan Bermotor yang telah diregistrasi sebagaimana dimaksud dalam Pasal 64 ayat (1) dapat dihapus dari daftar registrasi dan identifikasi Kendaraan Bermotor atas dasar:

a. permintaan pemilik Kendaraan Bermotor; atau
b. pertimbangan pejabat yang berwenang melaksanakan registrasi Kendaraan Bermotor.

(2) Penghapusan registrasi dan identifikasi Kendaraan Bermotor sebagaimana dimaksud pada ayat (1) huruf b dapat dilakukan jika:

a. Kendaraan Bermotor rusak berat sehingga tidak dapat dioperasikan; atau

b. pemilik Kendaraan Bermotor tidak melakukan registrasi ulang sekurang-kurangnya 2 (dua) tahun setelah habis masa berlaku Surat Tanda Nomor Kendaraan Bermotor.

(3) Kendaraan Bermotor yang telah dihapus sebagaimana dimaksud pada ayat (1) tidak dapat diregistrasi kembali.

Tabel 3.6 Implementasi Kodefikasi Huruf

ID	**Description**
1Z2X3C4V5B6N7M8P9#a01074	Permintaan pemilik Kendaraan Bermotor; atau
1Z2X3C4V5B6N7M8P9#b01074	Pertimbangan pejabat yang berwenang melaksanakan registrasi Kendaraan Bermotor.
1Z2X3C4V5B6N7M8P9#a02074	Kendaraan Bermotor rusak berat sehingga tidak dapat dioperasikan; atau
1Z2X3C4V5B6N7M8P9#b02074	Pemilik Kendaraan Bermotor tidak melakukan registrasi ulang sekurang-kurangnya 2 (dua) tahun setelah habis masa berlaku Surat Tanda Nomor Kendaraan Bermotor.

Lalu pengaplikasian pada teks nya adalah dengan kodefikasi untuk huruf seperti pada gambar 3.5 berikut:

1Z2X3C4V5B6N7M8P9#a01074Permintaan pemilik Kendaraan Bermotor; atau1Z2X3C4V5B6N7M8P9#b01074Pertimbangan pejabat yang berwenang melaksanakan registrasi Kendaraan Bermotor.1Z2X3C4V5B6N7M8P9#a02074Kendaraan Bermotor rusak berat sehingga tidak dapat dioperasikan; atau1Z2X3C4V5B6N7M8P9#b02074Pemilik Kendaraan Bermotor tidak melakukan registrasi ulang sekurang-kurangnya 2 (dua) tahun setelah habis masa berlaku Surat Tanda Nomor Kendaraan Bermotor.

Gambar 3.5 Implementasi Kodefikasi Huruf pada Teks

Dilanjutkan dengan kodefikasi Juncto yang akan dijelaskan seperti pada Tabel 3.7 berikut :

Tabel 3.7 Deskripsi Juncto

Nama Field	Deskripsi
ID	Kodefikasi untuk menunjuk keterkaitan dengan Pasal lain, namun tidak semua Pasal atau Ayat memiliki Juncto.
Deskripsi	Keterkaitan dengan Pasal lain.

Pada tabel Juncto ini menjelaskan ID (General ID + No huruf (1 Digit) + No Ayat (2 Digit) + No Pasal (3 Digit)), adalah kodefikasi untuk menunjuk kepada sebuah huruf pada pasal tertentu dan deskripsi adalah penjelasan isi dari huruf tersebut. Sebgaimana dicontohkan pada Pasal 309 yang memiliki Juncto ke Pasal 189 berikut :

Pasal 309

Setiap orang yang tidak mengasuransikan tanggung jawabnya untuk penggantian kerugian yang diderita oleh Penumpang, pengirim barang, atau pihak ketiga sebagaimana dimaksud dalam **Pasal 189** dipidana dengan pidana kurungan paling lama 6 (enam) bulan atau denda paling banyak Rp1.500.000,00 (satu juta lima ratus ribu rupiah).

1Q2W3E4R5T6Y7U8I9O309Setiap orang yang tidak mengasuransikan tanggung jawabnya untuk penggantian kerugian yang diderita oleh Penumpang, pengirim barang, atau pihak ketiga sebagaimana dimaksud dalam Pasal 189 dipidana dengan pidana kurungan paling lama 6 (enam) bulan atau denda paling banyak Rp1.500.000,00 (satu juta lima ratus ribu rupiah).**[okmijnuhbygvtfcrdx]**189

Gambar 3.6 Implementasi Kodefikasi Juncto pada teks

3.1.5. Analisis Algoritma

Pembuatan aplikasi pada perangkat mobile tidak terlepas dari algoritma. Dibutuhkan proses analisis algoritma agar membantu dalam meningkatkan efisiensi aplikasi. Kecanggihan suatu aplikasi bukan dilihat dari tampilan program, tetapi berdasarkan efisiensi algoritma yang terdapat didalam aplikasi tersebut. Analisis algoritma adalah bahasan utama dalam ilmu komputer. Dalam proses analisis algoritma ini juga dilakukan pengkodefikasian, kumpulan pasal-

pasal yang telah di pecah dalam bentuk teks karena aplikasi ini tidak menggunakan *database* seperti aplikasi kebanyakan, yang bertujuan untuk mengefisiensikan pemakaian *memory*. Maka diperlukan juga analisis data atau managmen data teks.

3.1.6. Analisis Algoritma Boyer-Moore

Algoritma Boyer-Moore merupakan algoritma pencocokan string (*String Macthing*) yang ditemukan oleh Bob Boyer dan J.Stroher Moore pada tahun 1977. Algoritma Boyer-Moore menggunakan metode string dari kanan ke kiri yaitu memindai karakter *pattern* dari kanan ke kiri mulai dari karakter paling kanan ke kiri. Karakter paling kanan pada pola merupakan karakter pertama yang akan dicocokan dengan teks. Algoritma ini mempunyai dua fase, yaitu *preprocessing* dan fase pencarian.

Pada fase *preprocessing* terdapat dua buah fungsi untuk menggeser pola kearah kanan. Kedua fungsi ini disebut disebut *good-suffix-shift* dan *bad-character-shift* adalah fungsi yang memiliki aturan dan cara perhitungan yang akan menghasil kan nilai untuk dijadikan acuan pergeseran *pattern* pada teks. Untuk hasil perhitungan *bad-character-shift* akan menghasilkan nilai *Occurrence Heuristic* (OH) pada tabel BmBc. Sedangkan untuk hasil perhitungan *good-suffix-shift* akan menghasilkan nilai *Match Heuristic* (MH) pada tabel BmGs. Untuk lebih jelasnya akan dijelaskan pada tabel selanjutnya.

Algoritma Boyer-Moore menggunakan dua buah tabel (BmGs & BmBc) untuk mengolah informasi saat terjadi kegagalan pencocokan string. Tabel pertama akan menghitung banyaknya pergeseran yang harus dilakukan berdasarkan identitas karakter yang menyebabkan kegagalan pencocokan string. Jumlah pergeseran yang terdapat pada tabel pertama ini disebut bad character shitf. Tabel kedua juga akan menghitung banyaknya pergeseran yang harus dilakukan, tetapi berdasarkan jumlah karakter yang berhasil dicocokan sebelum pencocokan string tersebut gagal.

Tahap pertama menentukan *pattern* atau kata yang akan dicari, dalam contoh penerapan pada aplikasi digunakan contoh kata “SEPEDA” yang akan dicari pada array *linkedlist* dalam aplikasi.

Pattern : SEPEDA

Teks : 1Z2X3C4V5B6N7M8P9#g01025fasilitas untuk sepeda, Pejalan Kaki, dan penyandang cacat;

Tabel 3.8. BmBc

Char	D	E	P	S	A
OH value	1	2	3	5	6

Tabel 3.9. BmGs

Index	0	1	2	3	4	5
Char	S	E	P	E	D	A
MH value	6	6	6	6	6	1

Tahapan selanjutnya adalah proses pencocokan pattern pada teks yang akan dicari, dimana sampel teks yang akan dicari adalah :

Teks: 1Z2X3C4V5B6N7M8P9#g01025fasilitas untuk sepeda, Pejalan Kaki, dan penyandang cacat;

Berikut adalah keterangan pada proses pencocokan teks menggunakan algoritma Boyer-Moore:

Index pergeseran : *o* Panjang Pattern : *p*

Index Pattern : *m* Hasil Pencarian : *r*

Cocok : Tidak Cocok : Belum diproses :

Proses #1, Penggeseran. Menempatkan pattern di posisi *o*. Dengan *o* = 0. Seperti pada Tabel 3.10 berikut :

Tabel 3.10 Proses ke 1 Pergeseran Pencocokan

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	n
1	Z	2	X	3	C	4	V	5	B	6	N	7	M	8	P	9	#	G	0	1	0	2	5	F	A	…
S	E	P	E	D	A																					

Proses #2, Pencocokan. Mencocokan setiap karakter. pattern[*m*..0] : teks[(*m*+*o*)..*o*]. Seperti pada Table 3.11 berikut :

Tabel 3.11 Proses ke 2 Pergeseran Pencocokan**.**

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	**n**
1	Z	2	X	3	C	4	V	5	B	6	N	7	M	8	P	9	#	G	0	1	0	2	5	F	A	…
S	E	P	E	D	A																					

Dari Tabel 3.11 ditemukan ketidakcocokan pada pattern[p-1]:teks[o+(p-1)]. Maka akan dilakukan perhitungan o berdasarkan OH dan MH. Sehingga pattern akan di geser ke kanan sebesar 6 karakter. Proses selanjutnya:

Proses #3, Mengulangi proses #1 dan #2 dengan *o* = *o* + 18.

Tabel 3.12 Proses ke 3 Pergeseran Pencocokan

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	**n**
1	Z	2	X	3	C	4	V	5	B	6	N	7	M	8	P	9	#	G	0	1	0	2	5	F	A	…
						S	E	P	E	D	A															

Dari Tabel 3.12 ditemukan ketidakcocokan pada pattern[p-1]:teks[o+(p-1)]. Maka akan dilakukan perhitungan o berdasarkan OH dan MH. Sehingga pattern akan di geser ke kanan sebesar 18 karakter. Proses selanjutnya :

Proses #4, Mengulangi proses #1 dan #2 dengan o = o + 6.

Tabel 3.13 Proses ke 4 Pergeseran Pencocokan

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	**n**
1	Z	2	X	3	C	4	V	5	B	6	N	7	M	8	P	9	#	G	0	1	0	2	5	F	A	…
												S	E	P	E	D	A									

Proses #5, Mengulangi proses #1 dan #2 dengan *o* = *o* + 6.

Tabel 3.14 Proses ke 5 Pergeseran Pencocokan

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	**n**
1	Z	2	X	3	C	4	V	5	B	6	N	7	M	8	P	9	#	G	0	1	0	2	5	F	A	…
																		S	E	P	E	D	A			

Proses #6, Mengulangi proses #1 dan #2 dengan *o* = *o* + 6.

Tabel 3.15 Proses ke 6 Pergeseran Pencocokan

24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	**n**
F	A	S	I	L	I	T	A	S		U	N	T	U	K		S	E	P	E	D	A	,	P	E	J	…
S	E	P	E	D	A																					

Proses #7, Mengulangi proses #1 dan #2 dengan *o* = *o* + 6.

Tabel 3.16 Proses ke 7 Pergeseran Pencocokan

24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	**n**
F	A	S	I	L	I	T	A	S		U	N	T	U	K		S	E	P	E	D	A	,	P	E	J	…
						S	E	P	E	D	A															

Proses #8, Mengulangi proses #1 dan #2 dengan *o* = *o* + 6

Tabel 3.17 Proses ke 8 Pergeseran Pencocokan

24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	**n**
F	A	S	I	L	I	T	A	S		U	N	T	U	K		S	E	P	E	D	A	,	P	E	J	…
												S	E	P	E	D	A									

Proses #9, Mengulangi proses #1 dan #2 dengan *o* = *o* + 2.

Tabel 3.18 Proses ke 9 Pergeseran Pencocokan

24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	**n**
F	A	S	I	L	I	T	A	S		U	N	T	U	K		S	E	P	E	D	A	,	P	E	J	…
														S	E	P	E	D	A							

Proses #10, Mengulangi proses #1 dan #2 dengan *o* = *o* + 2.

Tabel 3.19 Proses ke 10 Pergeseran Pencocokan

24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	**n**
F	A	S	I	L	I	T	A	S		U	N	T	U	K		S	E	P	E	D	A	,	P	E	J	…
																S	E	P	E	D	A					

r = r,o atau r = 0,40

Proses #11, Mengulangi proses #1 dan #2 dengan $o = o + 18$.

Tabel 3.20 Proses ke 11 Pergeseran Pencocokan

46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	**n**
,		P	E	J	A	L	A	N		K	A	K	I	,	D	A	N		P	E	N	Y	A	N	D	…
S	E	P	E	D	A																					

Proses #12, Mengulangi proses #1 dan #2 dengan $o = o + 18$.

Tabel 3.21 Proses ke 12 Pergeseran Pencocokan

46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	**n**
,		P	E	J	A	L	A	N		K	A	K	I	,	D	A	N		P	E	N	Y	A	N	D	…
						S	E	P	E	D	A															

Proses #13, Mengulangi proses #1 dan #2 dengan $o = o + 6$.

Tabel 3.22 Proses ke 13 Pergeseran Pencocokan

46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	**n**
,		P	E	J	A	L	A	N		K	A	K	I	,	D	A	N		P	E	N	Y	A	N	D	…
												S	E	P	E	D	A									

Proses #14, Mengulangi proses #1 dan #2 dengan $o = o + 6$.

Tabel 3.23 Proses ke 14 Pergeseran Pencocokan

46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	**n**
,		P	E	J	A	L	A	N		K	A	K	I	,	D	A	N		P	E	N	Y	A	N	D	…
																		S	E	P	E	D	A			

Proses #15, Mengulangi proses #1 dan #2 dengan $o = o + 6$.

Tabel 3.24 Proses ke 15 Pergeseran Pencocokan

70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	**n**
N	D	A	N	G		C	A	C	A	T	;															…
S	E	P	E	D	A																					

Proses #16, Mengulangi proses #1 dan #2 dengan $o = o + 6$.

Tabel 3.25 Proses ke 16 Pergeseran Pencocokan

70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	n
N	D	A	N	G		C	A	C	A	T	;															...
						S	E	P	E	D	A															

Proses pencocokan dihentikan karena $o+p$ > panjang teks. Sehingga didapatlah posisi pattern didalam teks adalah Teks{ [40..(40+p-1)] }= teks { [40..46]}. Dan didapatkan Array posisi hasil proses BM : 66. Dengan pemaksimalan nilai OH dan MH, disimpulkan bahwa jumlah pergeseran BM 10:95 lebih cepat dibandingkan dengan pencocokan biasa pada kasus ini.

Dengan memanfaatkan dictionary yang telah terbentuk, ruang pencarian pun semakin kecil karena pencarian tidak dilakukan di teks asset melainkan di deskripsi yang terdapat di masing-masing node. Sehingga, dengan kombinasi karakter apapun yang menjadi kata kunci pencarian, memungkinkan pencapaian proses BM yang optimal.

Peran BM pada proses ini adalah untuk menemukan posisi pemenggalan deskripsi agar dapat dibentuk menjadi string preview. String preview adalah potongan kalimat yang di dalamnya terdapat kata kunci yang dicari, seperti contoh:

Deskripsi : 1Z2X3C4V5B6N7M8P9#g01025fasilitas untuk sepeda, Pejalan Kaki, dan penyandang cacat;

Kata Kunci : sepeda

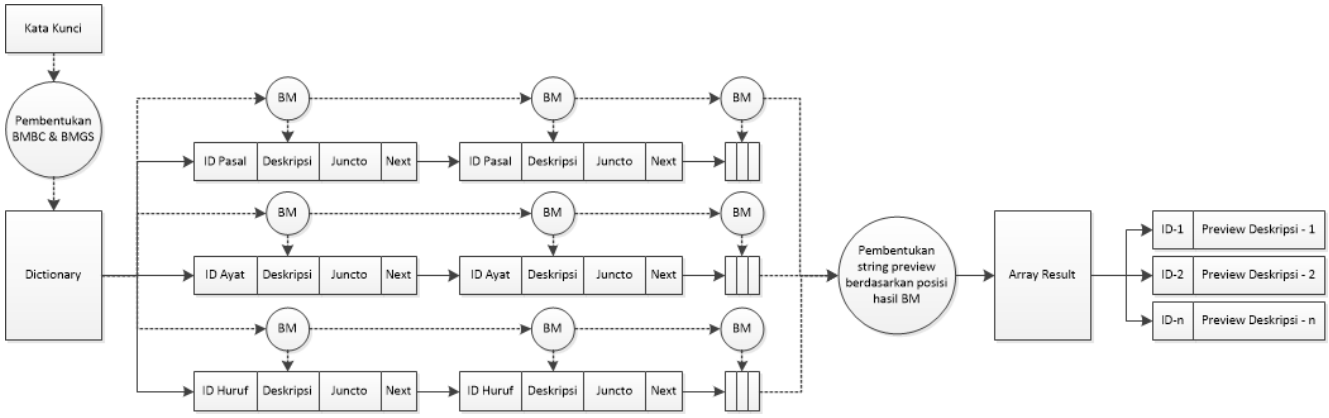
Dengan menerapkan pencocokan string dengan metode BM (seperti contoh sebelumnya), maka didapatkan lah posisi karakter sebagai tolok ukur pemenggalan. Sehingga didapatkan hasil seperti contoh :

String Preview : fasilitas untuk sepeda, Pejalan Kaki, dan penyandang cacat;

Pada proses pencarian Boyer-Moore yang kedua ini terdapat kondisi khusus yaitu:

1. Proses pembentukan nilai OH dan MH, hanya akan dilakukan 1 kali pada awal proses pencarian sebelum dilakukan penelusuran. Atau dengan kata lain, nilai OH dan MH tidak akan diproses ulang pada saat BM di proses di setiap node dictionary.
2. BM akan berhenti apabila telah ditemukan kecocokan string pada deskripsi dengan minimal satu kecocokan, atau BM akan berhenti apabila telah mencapai ujung deskripsi.
3. BM akan mencocokan karakter dengan membandingkan ASCII masing-masing karakter. Dengan kata lain, BM menjadi case sensitive pada proses ini.

Kondisi khusus diatas, dilakukan untuk mengoptimalisasikan proses pencarian dari sudut pembentukan string preview



Gambar 3.7 Representasi Pencarian UU Lalu Lintas

3.1.7. Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional adalah sebuah langkah dimana seorang pembangun perangkat lunak (*software develover)* menganalisis sumber daya manusia yang akan menggunakan perangkat lunak yang dibangun, perangkat keras dan perangkat dimiliki sesuai dengan kebutuhan, sehingga dapat ditentukan kompabilitas apliasi yang dibangun terhadap sumber daya yang ada.

Setelah melakukan analisis non fungsional, maka dilanjutkan ke langkah berikutnya yaitu menentukan kebutuhan non fungsional sistem yang akan dibangun untuk disesuaikan dengan fakta yang ada. Apabila kebutuhan tidak terpenuhi maka sistem yang dibangun tidak akan berjalan dengan baik sesuai yang diharapkan.

Analisis kebutuhan non fungsional yang dilakukan dibagi dalam tiga tahap, yaitu :

1. Analisis perangkat keras (*hardware*).
2. Analisis perangkat lunak (*software*).
3. Analisis pengguna (*user)*.

3.1.7.1. Analisis Perangkat Keras (*hardware)*

Berdasarkan analisis spesifikasi perangkat keras minimum yang dibutuhkan untuk membangun aplikasi ini adalah :

1) Processor minimal 2 Ghz
2) Hardisk minimal 40 GB
3) RAM minimal 512 MB
4) Grapic Card minimal 64 MB
5) Mouse & Keyboard

Sedangkan spesifikasi perangkat keras pada smartphone Android minimum yang dibutuhkan adalah :

1) Processor minimal 600Mhz
2) RAM minimal 128 MB
3) Memory / Storage minimal 128 MB

4) Layar Touchscreen minimal (320 x 480 pixels)

5) Speaker

6) Jaringan GSM / CDMA

Spesifikasi minimal perangkat keras yang tercantum pada umum nya perangkat smartphone yang beredar dipasaran sekarang ini telah memenuhi persyaratan minimum untuk menjalankan aplikasi sehingga tidak memerlukan penambahan.

3.1.7.2. Analisis perangkat Lunak (*software*)

Perangkat lunak diguanakan dalam sebuah sistem merupakan perintah-perintah yang diberikan kepada perangkat keras agar bisa saling berinteraksi di antara keduanya. Perangkat lunak yang dibutuhkan untuk membangun aplikasi pencocokan pasal ini adalah sebgai berikut :

1) Sistem Operasi Windows Vista atau 7

2) J2SE SDK (*Java* 2 *Standart Edition Software Development Kit*) versi 5.

3) Java Runtime Edition, sebagai *platform* untuk menjalankan sistem.

4) Eclipse versi 4.2.0 (Juno).

5) ADT (Android Development Tools).

6) AVD (Android Virtual Device).

Program ini dirancang menggunakan Java Eclipse dengan menggunakan standar untuk aplikasi GUI pada level tinggi dan level rendah. Jadi aplikasi ini dapat digunakan pada *smartphone* berbasis android dengan minimal versi 2.1 atau biasa dikenal dengan Froyo (*Frozen Yoghurt).* Oleh karena itu dalam pengujianya menggunakan *emulator* android.

3.1.7.3. Analisis Pengguna (*user).*

Spesifikasi pengguna adalah kriteria yang harus dimiliki oleh pengguna agar tidak kesulitan dalam menjalankan aplikasi pencarian pasal ini, spesifikasi penggunanya adalah sebgai berikut :

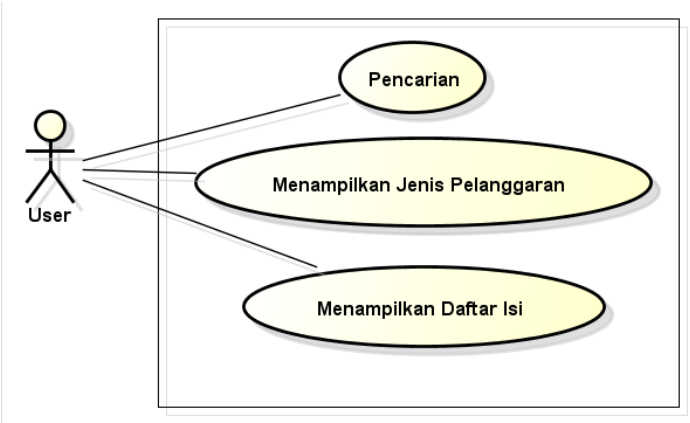
1) Mengerti operasional *Smartphone* Android.
2) Dapat mengoperasikan aplikasi pada *Smartphone* Android.
3) Menguasai bahasa Indonesia dengan baik dan benar.
4) Mengetahui istilah-istilah baik dalam aktivitas lalu-lintas maupun berkendara.
5) Memiliki *Smartphone* Android dengan minimal versi 2.1 (Froyo).

3.1.8. Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar sistem dapat berjalan dengan baik sesuai kebutuhan. Analisis kebutuhan fungsional dimodelkan dengan menggunakan UML (Unified Modeling Language).

1. Use Case Diagram

Pemodelan yang digunakan dalam menggambarkan kebutuhan fungsional pada aplikasi yang dibangun yaitu dengan pemodelan Use Case Diagram. Use Case Diagram pada aplikasi pencarian pasal lalu-lintas ini seperti pada gambar Gambar 3.8. berikut ini :



Gambar 3.8. Use Case Diagram

Gambar 3.9. menjelaskan bahwa aktor sebagai pengguna aplikasi (*frontend)* yang berinteraksi dengan perangkat mobile atau *smartphone* berbasis android.

1.1 Identifikasi Aktor

Aktor yang berperan disini adalah *user* yang memiliki peran dalam menggunakan aplikasi pencarian Pasal Lalu lintas untuk mendapatkan informasi Pasal.

1.2 Identifikasi *Use Case*

Identifikasi *Use Case* adalah *Use Case* yang bekerja pada sistem aplikasi adalah pada Tabel 3.33 berikut :

Tabel 3.33. Identifikasi Use Case

No	Nama use case	Deskripsi
UC-01	Pencarian	Fungsionlaitas utama untuk melakukan pencarian kata ataupun pasal yang ingin dicari secara detail beserta dengan juncto (pasal yang berkaitan dengan pasal yang dicari).
UC-02	Menampilkan Daftar Isi	Fungsionalitas sekunder untuk melakukan pencarian pasal BAB atau kategori.
UC-03	Menampilkan Jenis Pelanggaran	Fungsionalitas yang dikhususkan bertujuan memudahkan user dalam melakukan berdasarkan intensitas pelanggaran yang sering dilakukan oleh masyarakat umumnya.

1.3 Sekenario Use Case

Sekenario Use Case adalah alur cerita atau proses-proses yang terjadi pada sistem antara aktor dengan Use Case. Sekenario Use Case ditunjukan pada tabel Tabel 3.34. berikut :

Tabel 3.34. Sekenario Use Case Pencarian

Identifikasi	
Nama Use Case	Pencarian
Aktor	Pengguna
Tujuan	Mencari informasi tentang pasal.
Deskripsi	Pengguna mengetikan kata (*pattern)* / pasal yang dicari, kemudian system merespon dan menampilkan hasil pencarian.

Skenario Utama (UC01-A01)	
Kondisi awal	Tampilan awal setelah program
Aksi Aktor	**Reaksi Sistem**
	1.Ketika program pertama kali dijalankan maka sistem langsung melakukan pencarian, untuk pembuatan larik indeks pasal dan indeks suggest kata dari kumpulan teks pasal, ayat dan huruf.
2.Pengguna mengetikan kata (*pattern)* atau pasal yang akan dicari pada teks box kemudian menekan tombol cari yang tertampil pada sistem.	3.Ketika pengguna mengetikan huruf sistem langsung merespon dengan mencari suggest pada larik indeks suggest yang telah dibuat dan ditampilkan secara realtime apabila kata atau huruf tidak terdapat pada indeks suggest maka tidak akan tampil suggest.
4.Jika pengguna memilih suggest yang tampil.	5.Jika pengguna memilih suggest maka sistem akan menampilkan pasal yang terpilih beserta juncto (pasal yang berhubungan)
6.Jika pengguna tidak memilih suggest namun menekan tombol cari.	7.Jika pengguna menekan tombol cari maka sistem akan menampilkan pasal-pasal yang terkait dengan kata (*pattern*) beserta juncto (pasal yang berhubungan)
8.Memilih Pasal terkait	8.Menampilkan Pasal terkait sesuai dengan yang dipilih pengguna

Skenario Alternatif – Autentikasi Gagal	
Aksi Aktor	**Reaksi Sistem**
	1.Menampilkan pesan bahwa proses pencarian tidak dapat dilakukan (“Masukan kata yang akan dicari”)
	2.Menampilkan pesan bahwa proses pencarian tidak menemukan hasil (“cek penulisan kata dengan benar”)
Kondisi akhir	Menampilkan informasi pasal sesuai yang dicari, beserta juncto (pasal yang berhubungan) apabila pasal yang dicari memiliki keterkaitan dengan pasal yang lain.

Tabel 3.35. Sekenario Use Case Daftar Isi

Identifikasi	
Nama Use Case	Daftar Isi
Aktor	Pengguna
Tujuan	Mencari informasi tentang pasal melalui kategori atau BAB melalui menu daftar isi.
Deskripsi	Pengguna memilih menu Daftar Isi
Skenario Utama (UC02-A02)	
Kondisi awal	Tampilan awal setelah program
Aksi Aktor	**Reaksi Sistem**
	1.Ketika program pertama kali dijalankan maka sistem melakukan pembuatan list daftar isi.
2.Pengguna menekan tombol daftar isi	3.Munculkan urutan BAB beserta kategorinya
4.Pengguna memilih salah satu kategori atau BAB	5.Memunculkan daftar Pasal yang terkait dengan kategori atau BAB tersebut
6.Pengguna memilih salah satu pasal yang terdapat pada BAB atau kategori	7.Menampilkan detail dari pasal secara keseluruhan.
Skenario Alternatif – Autentikasi Gagal	
Aksi Aktor	**Reaksi Sistem**
Kondisi akhir	Menampilkan informasi salah pasal sesusuai kategori atau BAB yang dicari.

Tabel 3.36. Sekenario Use Case Jenis Pelanggaran

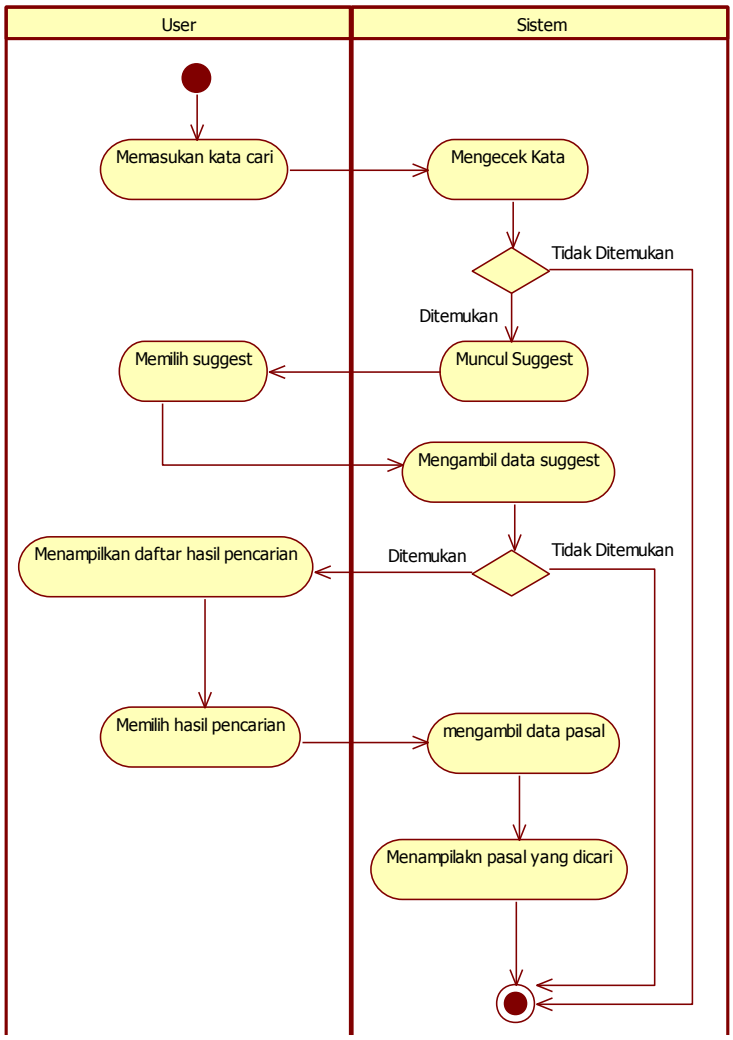
Identifikasi		
Nama Use Case	Jenis Pelanggaran	
Aktor	Pengguna	
Tujuan	Mencari informasi Pasal yang terkait dengan pelanggaran yang sering terjadi dilingkungan sekitar melalui menu contoh kasus pelanggaran	
Deskripsi	Pengguna memilih menu Jenis Pelanggaran	
Skenario Utama (UC03-A03)		
Kondisi awal	Tampilan awal setelah program	
Aksi Aktor		**Reaksi Sistem**
		1.Ketika program pertamakali dijalankan maka sistem akan menampilkan menu Contoh kasus pelanggaran.
2.Pengguna memlih Contoh kasus pelanggaran roda 2 / roda 4 ataupun roda 2 dan 4.		3.Sistem menampilakan Contoh Kasus sesuai kategori pelanggaran roda 2 / roda 4 atau roda 2 & 4.
4.Pengguna memilih salah satu Contoh kasus pelanggaran.		5.Sistem menampilkan deskripsi keterangan pelanggaran contoh kasus yang dipilih beserta keterangan Pasal yang di langgar dan pasal yang terkait.
6.Pengguna memilih pasal yang dilanggar atau pasal yang terkait(junto pasal yang dilanggar)		7.Sistem menampilakan Deskripsi Pasal yang dilanggar atau pasal yang terkait(junto pasal yang dilanggar)
Skenario Alternatif – Autentikasi Gagal		
Aksi Aktor		**Reaksi Sistem**
Kondisi akhir	Menampilkan informasi Pasal sesuai kategori Contoh pelanggaran yang dicari, beserta juncto (pasal yang berhubungan) apabila pasal yang dicari memiliki keterkaitan dengan pasal yang lain.	

2. Activity Diagram

Activity diagram adalah teknik untuk mendeskripsikan logika procedural dan alur kerja program. Activity diagram mempunyai peran seperti halnya *flowchart* akan tetapi perbedaannya dengan *flowchart* adalah activity diagram bisa mendukung perilaku pararel sedangkan *flowchart* tidak bisa.

2.1. Activity Diagram Pencarian

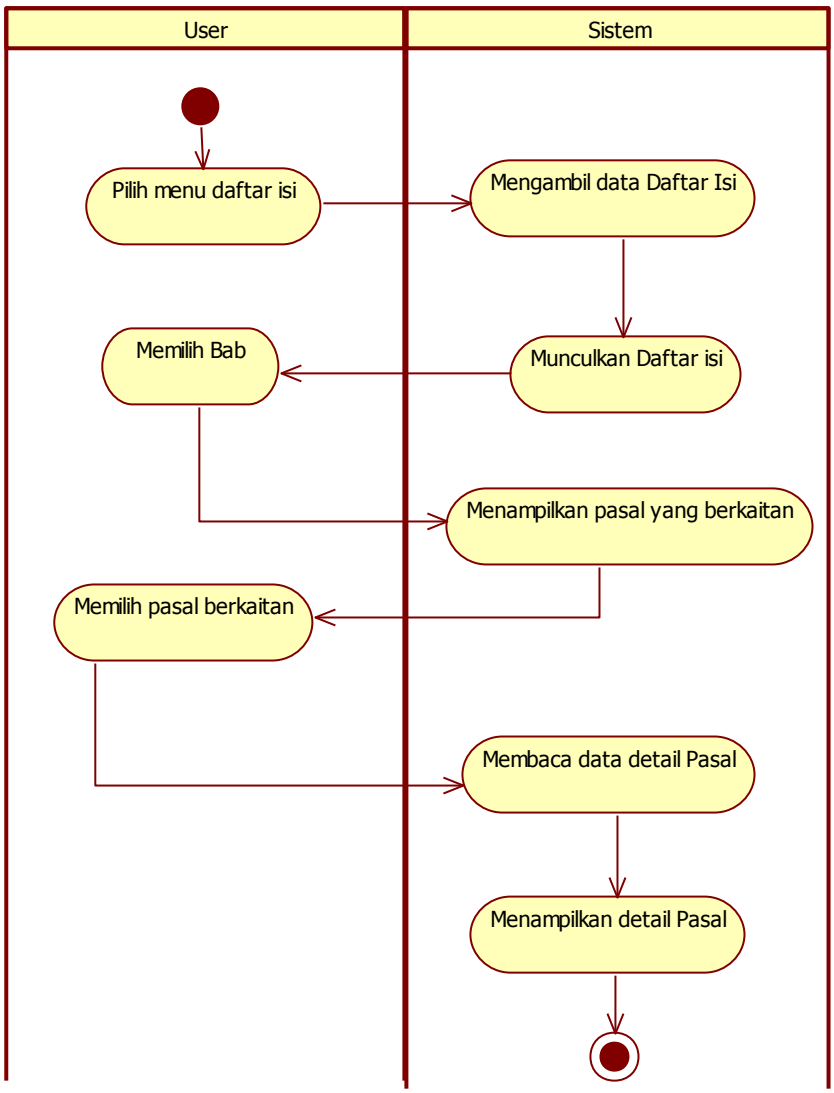
Berikut ini adalah prosedur pencarian pasal menggunakan *pattern* / kata cari dalam bentuk activity diagram seperti yang terlihat pada Gambar 3.9. dibawah ini :



Gambar 3.9. *Activity Diagram* Pencarian

2.2. Activity Diagram Daftar Isi

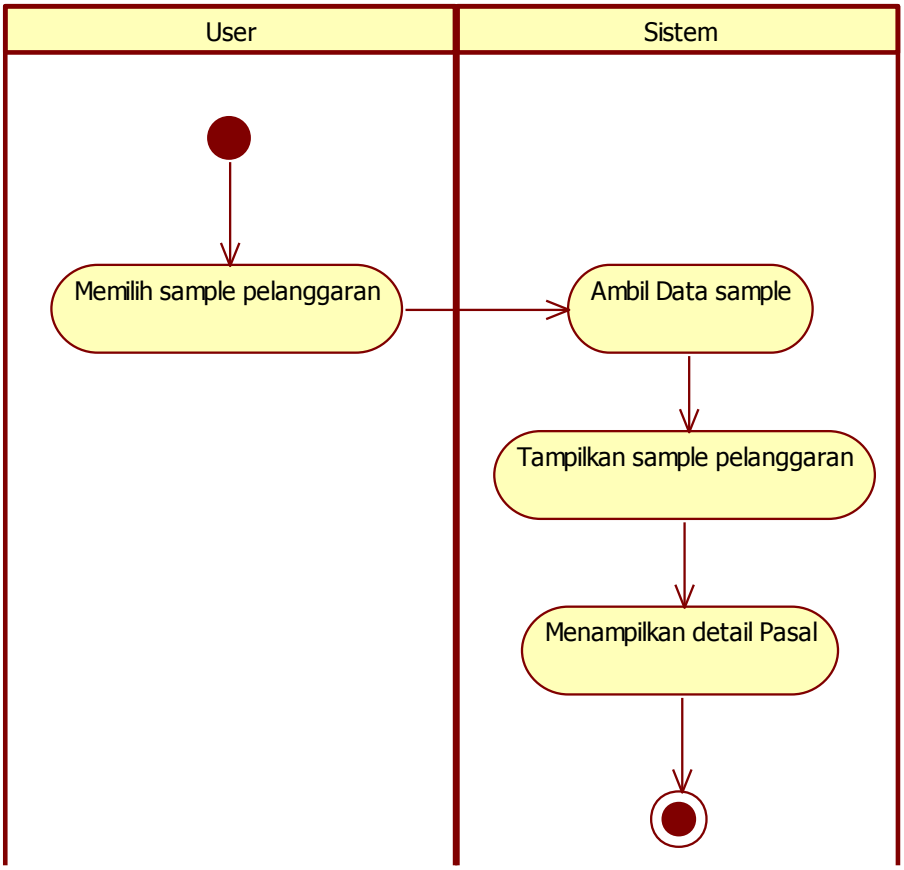
Berikut ini adalah prosedur pencarian Pasal melalui fitur daftar isi dalam bentuk activity diagram seperti yang terlihat pada Gambar 3.10. dibawah ini:



Gambar 3.10. Activity Diagram Daftar Isi

2.3. Activity Diagram Jenis Pelanggaran

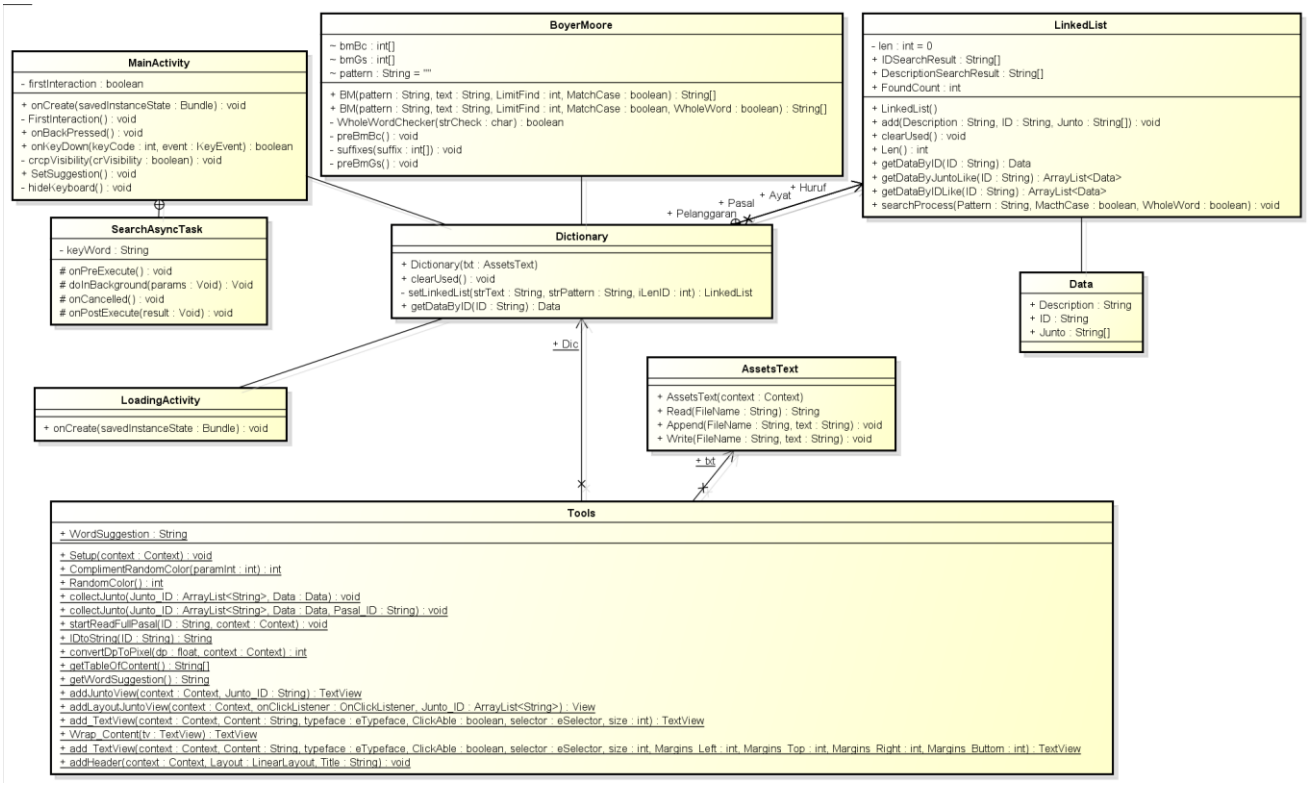
Berikut ini adalah prosedur pencarian Pasal melalui fitur sampel pelanggaran dalam bentuk activity diagram seperti yang terlihat pada Gambar 3.11. dibawah ini:



Gambar 3.11. Activity Diagram Sampel Pelanggaran

3. Class diagram

Class Diagram adalah spesifikasi objek, yang memiliki atribut/property dan layanan / fungsional (metode/fungsi).

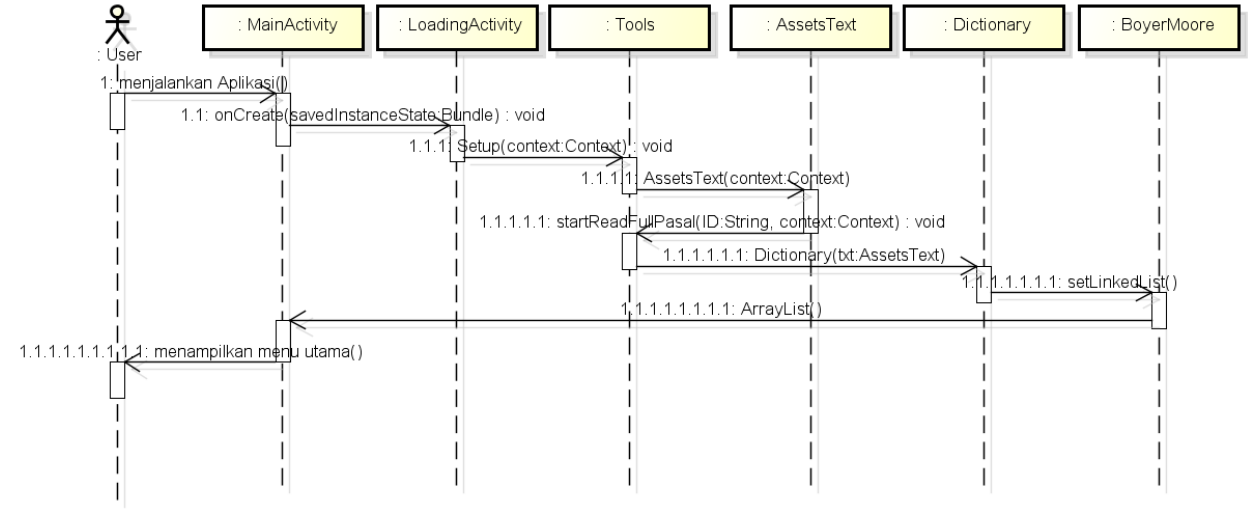


Gambar 3.12. Class Diagram

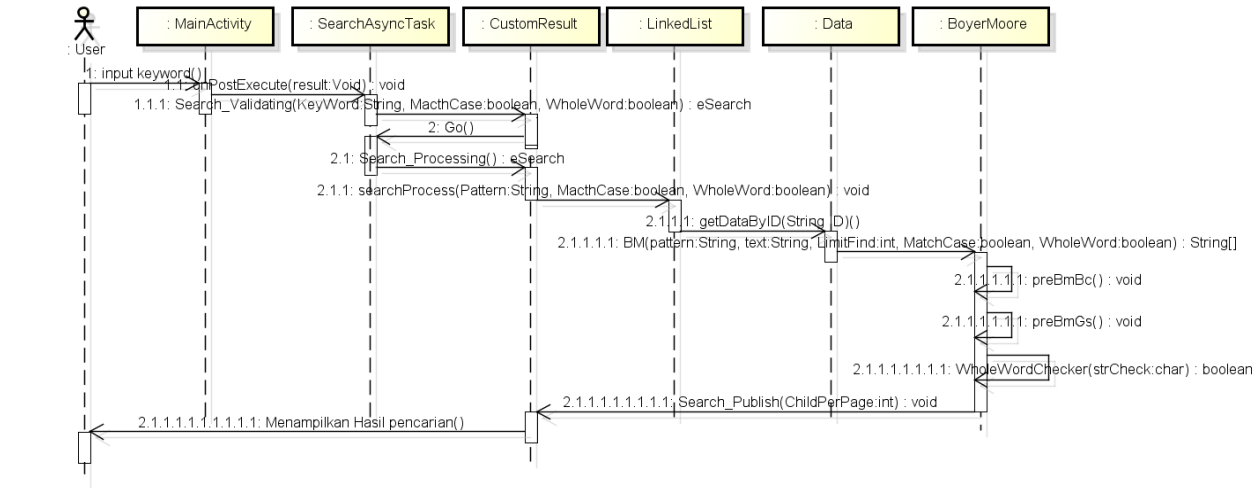
4. Squence Diagram

Diagram *sequence* atau *sequence* diagram merupakan gambaran kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Lebih jelas nya bisa dilihat pada gambar Gambar 3.13. , Gambar 3.14, Gambar 3.15 dan Gambar 3.16.

4.1. Squence Diagram Loading Activity & Pencarian

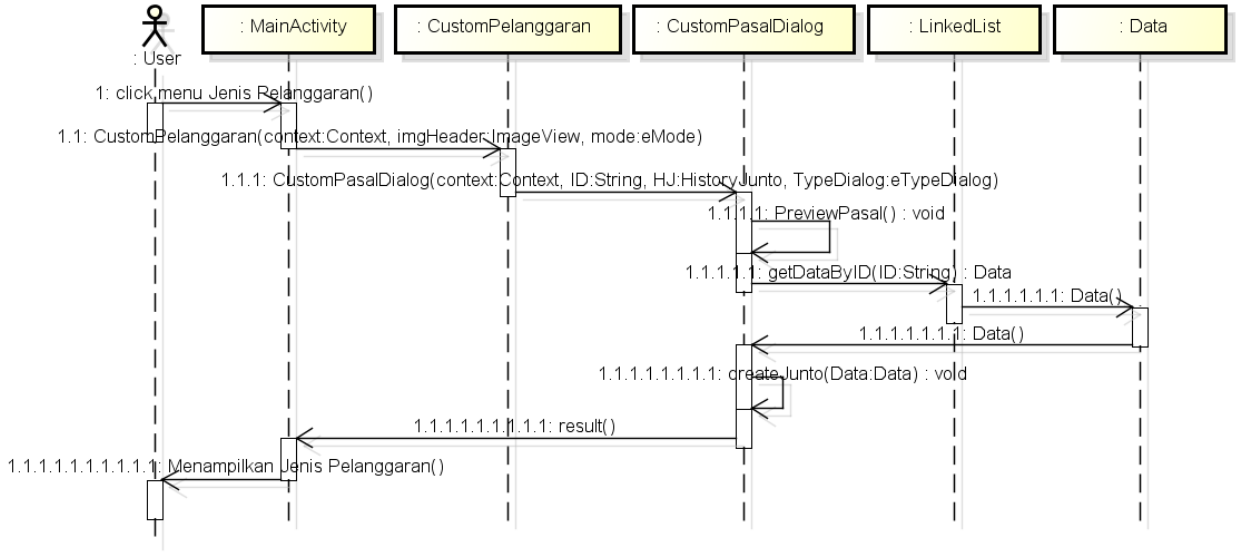


Gambar 3.13. Squence Diagram Loading Activity



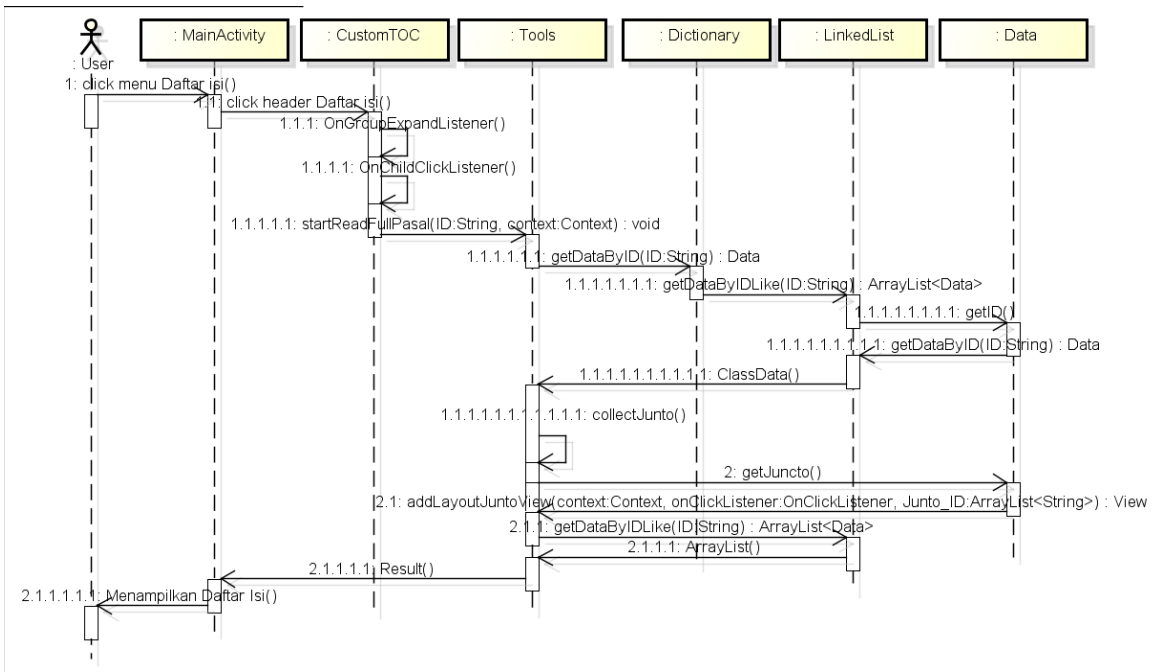
Gambar 3.14. Squence Diagram Pencarian

4.2. Squence Diagram Melihat Daftar Isi



Gambar 3.15. Squence Diagram Daftar Isi

4.3. Squence Diagram Melihat Jenis Pelanggaran



Gambar 3.16. Squence Diagram Jenis Pelanggaran

3.2. Perancangan Sistem

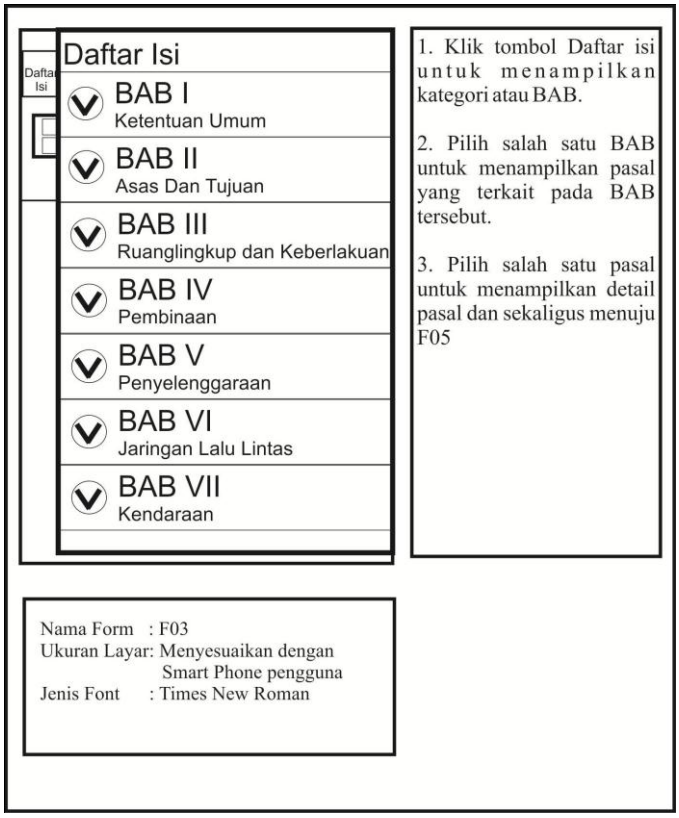
Perancangan merupakan penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam suatu kesatuan yang utuh. Tahapan ini meliputi konfigurasi perangkat lunak dan perangkat keras dari suatu sistem. Tahap perancangan desain bertujuan untuk mencari bentuk yang optimal dari aplikasi yang akan dibangun dengan pertimbangan fatktor-faktor permasalahan dan kebutuhan yang ada pada sistem seperti yang telah ditetapkan dengan cara mengkombinasikan penggunaan teknologi perangkat keras dan perangkat lunak yang tepat sehingga diperoleh yang optimal dan mudah diimplementasikan. Adapun perancangan sistem dari aplikasi ini pencarian pasal ini sebagai berikut.

3.2.1. Perancangan Antarmuka Perangkat Lunak

Perancangan antarmuka merupakan sebuah penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi. Adapun perancangan antarmuka aplikasi ini adalah sebagai berikut :

1. Desain Form Daftar isi

   Form daftar isi merupakan form yang digunakan sebagai tampilan user pada saat akan melakukan pencarian pasal melalui kategori ataupun BAB. Desain tampilan form dan objek dari aplikasi ini dapat dilihat pada Gambar 3.17. Desain tampilan form daftar isi.



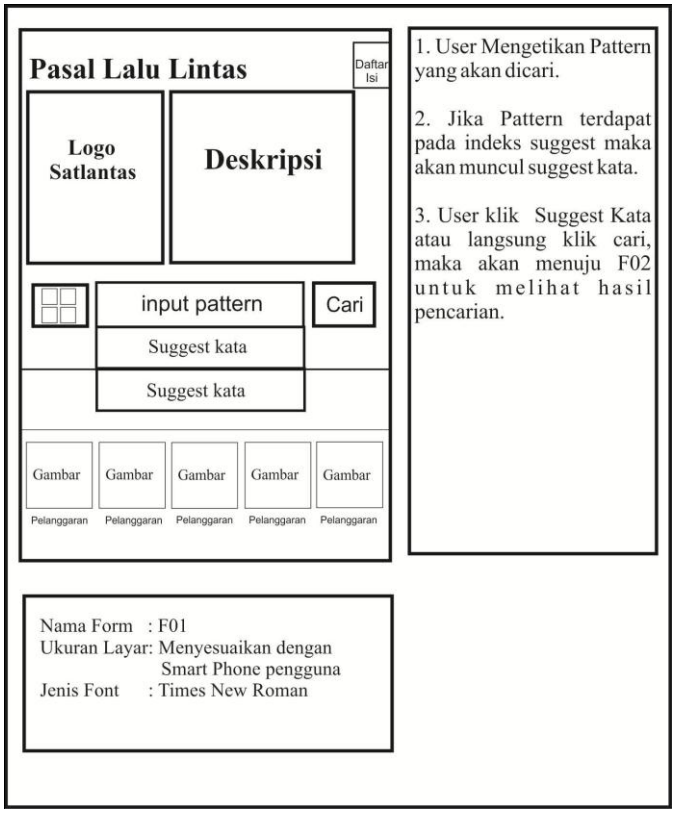
Gambar 3.17. Desain tampilan form Daftar isi

Tabel 3.36. Deskripsi objek form Daftar isi

Objek	Jenis	Keterangan
Daftar Isi	Button	Memunculkan menu daftar isi.
List Daftar isi	Spiner Dropdown	Memunculkan BAB / Kategori beserta Pasal yang mencakup kategori atau BAB tersebut

2. Desain Form Pencarian

Form pencarian merupakan form yang digunakan sebagai tampilan user pada saat akan melakukan pencarian pasal dengan cara memasukan kata yang akan dicari. Desain tampilan Form Pencarian ini dapat dilihat pada Gambar 3.18. Desain tampilan form Pencarian.



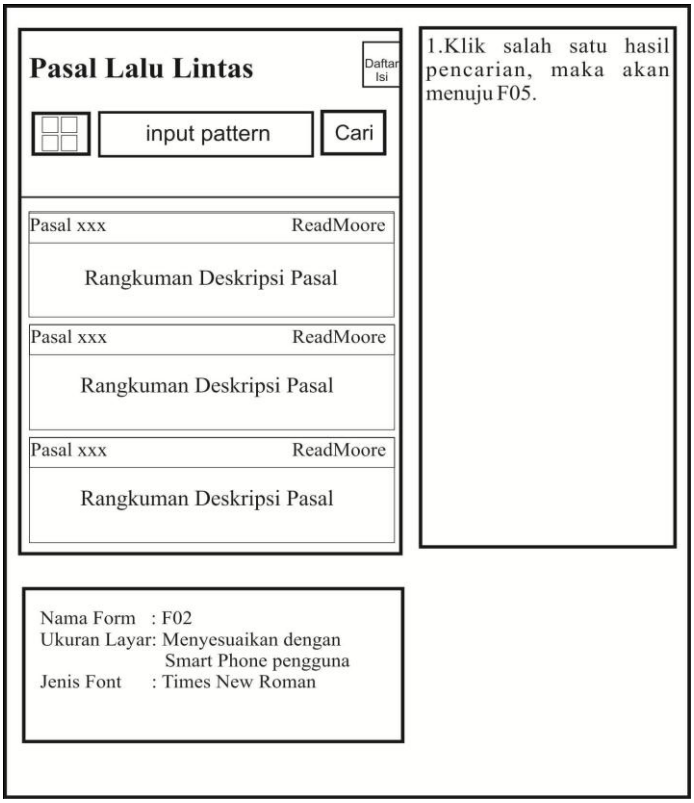
Gambar 3.18. Desain tampilan form Pencarian

Tabel 3.37. Deskripsi objek form Pencarian

Objek	Jenis	Keterangan
Input pattern	EditText	Menginputkan kata cari / *pattern*
Suggest kata	AutoCompleteTextView	Menampilkan suggest kata yang dicari
Cari	Button	Melakukan pencarian

3. Desain Form Hasil Pencarian kata

Form hasil pencarian kata merupakan form yang digunakan sebagai tampilan user pada saat sesudah melakukan pencarian kata pada seluruh Undang-undang. Desain tampilan Form hasil pencarian kata ini dapat dilihat pada Gambar 3.19. Desain tampilan form sampel pelanggaran.



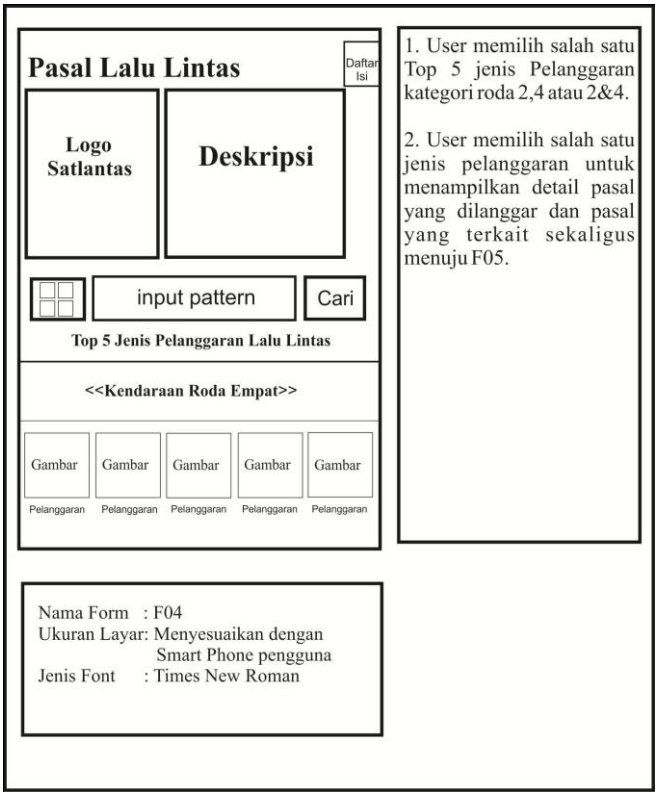
Gambar 3.19. Desain form hasil pencarian kata

Tabel 3.38. Deskripsi objek form Pencarian

Objek	Jenis	Keterangan
Read More	Button	Menampilkan Pasal yang terdapat kata yang dicari sekaligus menuju F05

4. Desain Form Sampel Pelanggaran

Form sampel pelanggaran merupakan form yang digunakan sebagai tampilan user pada saat akan melakukan pencarian contoh pelanggaran pasal yang sering terjadi. Desain tampilan Form sampel pelanggaran ini dapat dilihat pada Gambar 3.20. Desain tampilan form sampel pelanggaran.



Gambar 3.20. Desain tampilan form Pencarian

Tabel 3.39. Deskripsi objek form Pencarian

Objek	Jenis	Keterangan
Contoh Kasus Pelanggaran	Button	Membuka tampilan full page sampel pelanggaran F05
Gambar	ImageButton	Menampilkan isi contoh pelanggaran beserta Pasal yang dilanggar dan Pasal yang terkait

5. Desain Form Deskripsi Pasal

Form deskripsi Pasal merupakan form yang digunakan sebagai tampilan hasil pencarian setelah user melakukan pencarian Pasal melalui kata, daftar isi ataupun sampel pelanggaran. Desain tampilan Form sampel pelanggaran ini dapat dilihat pada Gambar 3.21. Desain tampilan form Deskripsi Pasal.



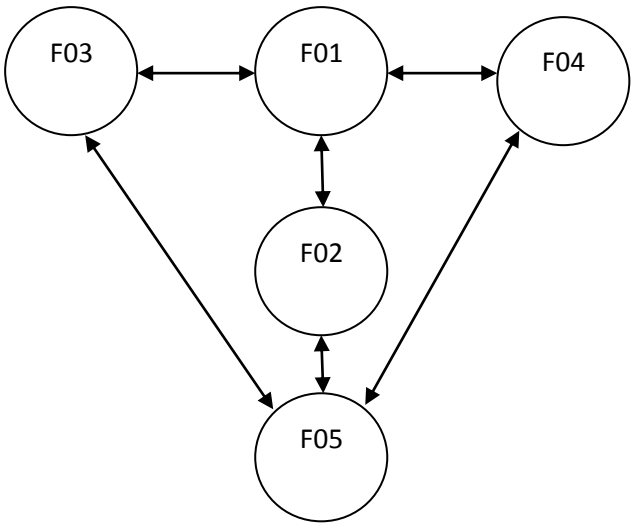
Gambar 3.21. Desain tampilan form Pencarian

Tabel 3.40. Deskripsi objek form Pencarian

Objek	Jenis	Keterangan
Pasal Terkait (juncto)	Button	Menampilkan Pasal terkait
Pasal Selengkapnya	Button	Menampilkan Pasal secara Detail beserta ayat dan huruf

3.2.2. Jaringan Simantik

Jaringan Semantik merupakan gambaran pengetahuan grafis yang menunjukkan hubungan antar objek (lingkaran) dan garis yang menggambarkan formasi antar objek tersebut. Adapun jaringan semantik untuk aplikasi ini dapat dilihat pada Gambar 3.11.



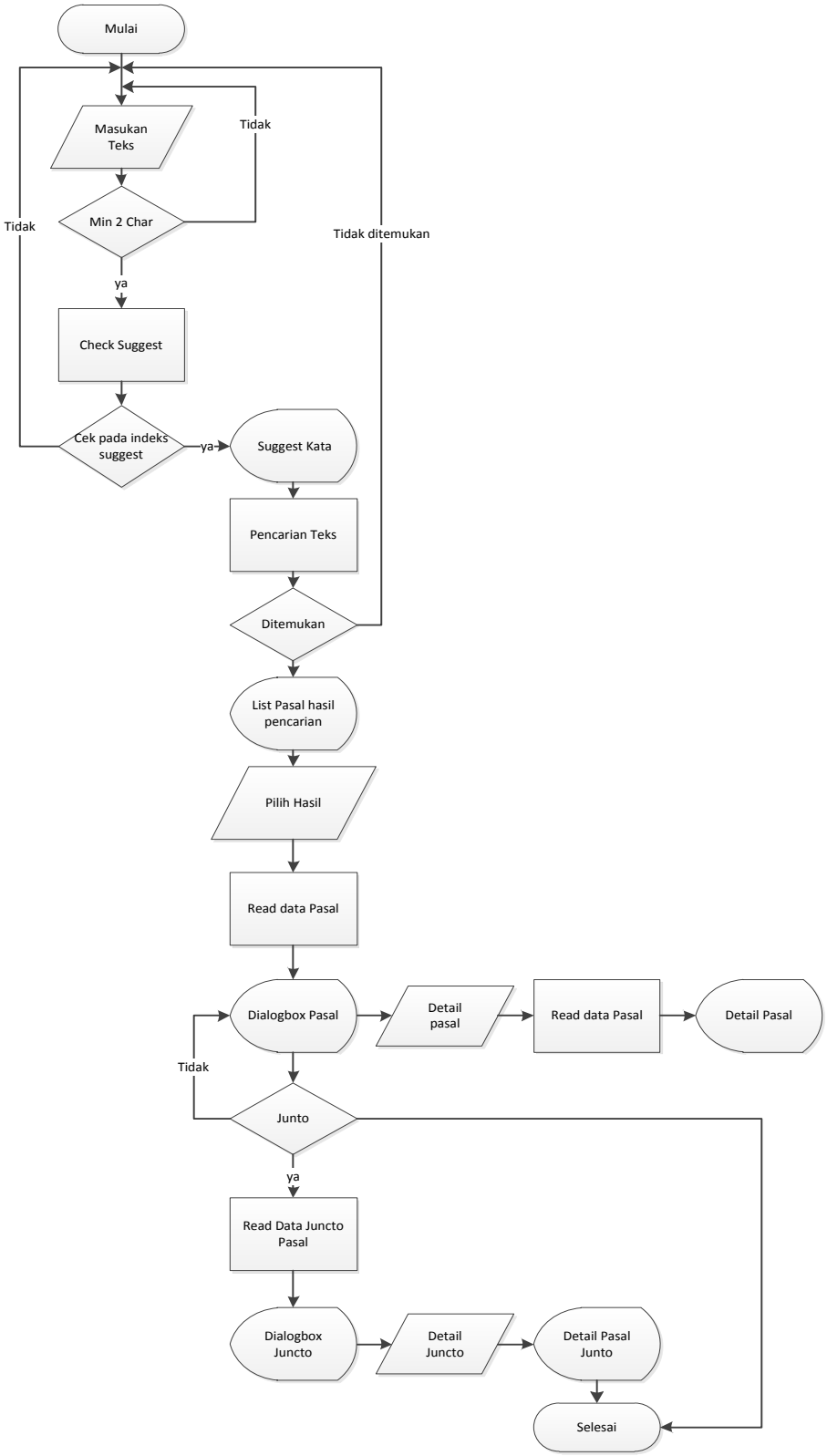
Gambar 3.22. Jaringan Semantik

3.2.3. Perancangan Prosedural

Perancangan prosedural adalah tata cara atau urutan langkah – langkah untuk melakukan suatu proses. Perancangan prosedural menggambarkan algoritma dari prosedur - prosedur yang terdapat pada perangkat lunak yang dibangun.

3.2.3.1. Perancangan Method Pencarian

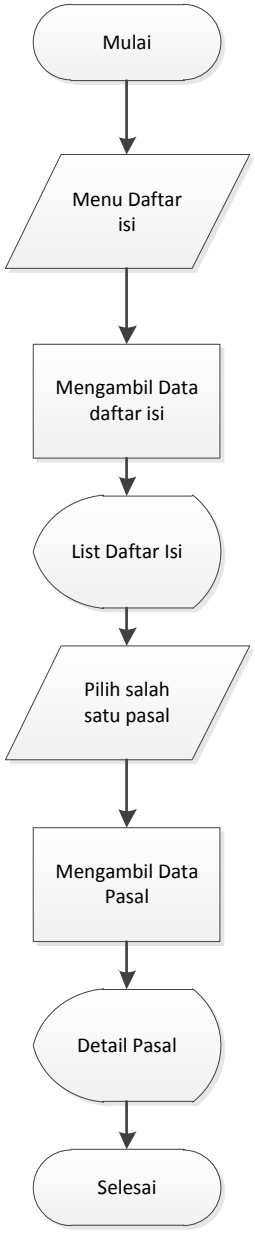
Berikut ini perancangan method Pencarian dapat dilihat pada Gambar 3.23. Rancangan Method Pencarian.



Gambar 3.23. Rancangan Method Pencarian

3.2.3.2. Perancangan Method Daftar Isi

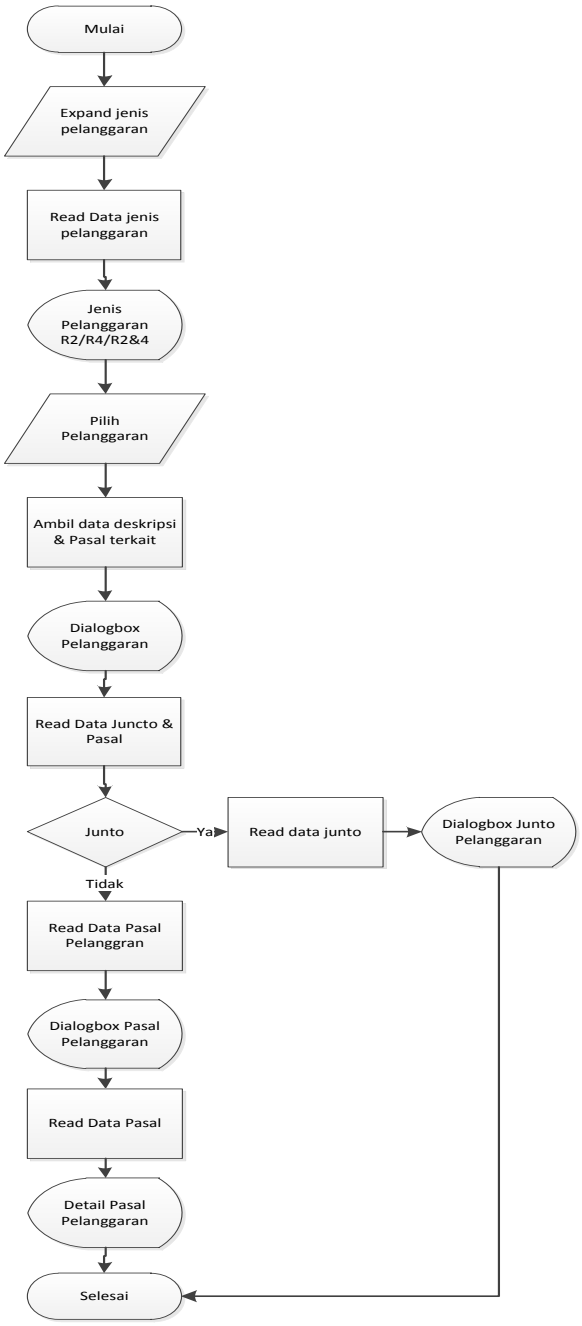
Berikut ini perancangan method Daftar Isi dapat dilihat pada Gambar 3.24. Rancangan Method Daftar Isi



Gambar 3.24. Rancangan Method Daftar Isi

3.2.3.3. Perancangan Method Jenis Pelanggaran

Berikut ini perancangan method Jenis Pelanggran dapat dilihat pada Gambar 3.25. Rancangan Method Jenis Pelanggaran



Gambar 3.25. Rancangan Method Jenis Pelanggaran