

BAB I

PENDAHULUAN

1.1. Latar Belakang

Dalam era transportasi yang semakin maju, mobil menjadi salah satu pilihan yang banyak diminati masyarakat. Semakin banyaknya jumlah penduduk pada suatu negara, ikut mempengaruhi banyaknya permintaan akan pembelian mobil. Salah satu masalah yang dapat dirasakan seiring dengan makin maraknya mobil adalah keterbatasan lahan parkir. Oleh karena itu pihak-pihak yang membutuhkan lahan parkir, mempunyai solusi dengan memakai sistem parkir di tiap lantai dalam gedung bertingkat. Tentunya permasalahan tidak berhenti sampai disana, banyak pengguna parkir yang mengeluh karena kesulitan didalam pencarian tempat parkir kosong. Beberapa metode telah ditawarkan dalam mengatasi permasalahan ini, salah satunya dengan memakai teknologi penginderaan jauh

Teknologi penginderaan jauh berbasis kamera saat ini merupakan salah satu masukan penting dalam pengolahan citra digital. Oleh karena itu analisis di dalam pengolahan citra digital menjadi kebutuhan yang amat penting. Analisis digital tidak hanya menawarkan waktu pengolahan data yang cepat, namun juga menawarkan analisis yang akurat seiring dengan berkembangnya teknologi dan ilmu pengetahuan.

Salah satu analisis dalam pengolahan citra digital adalah segmentasi. Segmentasi dapat menjadi sebuah analisis awal pada citra sebelum dilakukan analisis lanjut. Segmentasi memiliki keunggulan dari pada analisis yang berbasis piksel tunggal. Segmentasi merupakan analisis yang dapat bekerja pada sekelompok piksel dengan berbasiskan kepada nilai dan ruang spesialnya.

Secara umum segmentasi dapat dibagi menjadi tiga kelompok yaitu segmentasi berdasar klasifikasi (classification based segmentation), Segmentasi berdasar tepi (edge based segmentation), dan segmentasi berdasar daerah (region based segmentation). Segmentasi berdasar klasifikasi adalah proses segmentasi yang

dilakukan dengan cara mencari kesamaan dari ukuran tertentu pada nilai piksel, Segmentasi berdasar tepi adalah proses segmentasi dengan tujuan untuk mendapatkan batas tepi antar obyek. Sementara itu segmentasi berdasar daerah adalah segmentasi yang dilakukan guna mendapatkan daerah yang diduga sebagai obyek.

Pengolahan citra (*image processing*) dimana metode ini telah diaplikasikan di semua bidang. Dari bidang kedokteran sampai bidang militer. Pengolahan citra digunakan untuk memproses citra atau gambar dengan jalan memanipulasinya menjadi data gambar yang diinginkan untuk mendapatkan informasi tertentu..

Dalam pemrosesan citra, obyek yang dapat digunakan sangat luas dan bebas sesuai kebutuhannya. Dalam aplikasinya ,untuk mengolah Obyek untuk suatu keperluan tertentu pada umumnya menggunakan suatu kamera untuk menangkap Image obyek yang diolah . Obyek yang diambil dalam proyek akhir ini adalah mobil yang di deteksi dari jarak jauh sehingga dapat di diketahui dari luar ada tidaknya tempat kosong untuk parkir, dengan perwakilan board penunjuk.

1.2. Rumusan Masalah

Masalah yang ditangani pada tugas akhir ini, memakai pengolahan citra sebagai metode untuk sistem penuntun parkir. Dengan rumusan sebagai berikut:

- a. Mengambil gambar mobil yang dideteksi dengan menggunakan web kamera.
- b. Membuat program pendeteksi parkir dengan menggunakan metode pengolahan citra digital.
- c. Menginterpretasi ada nya mobil atau tidak pada tempat yang di-jadikan acuan.

1.3. Batasan Masalah

Dalam perencanaan dan penyusunan skripsi ini diperlukan batasan masalah sebagai berikut:

- a. Memakai 1 kamera dalam menangkap citra
- b. Benda yang di ukur tidak terlalu jauh sehingga, gambarnya dapat dengan jelas ditangkap oleh kamera web
- c. Kamera statis (tidak bergerak).
- d. Pengambilan contoh citra, dilakukan di dalam ruang tertutup (indor)
- e. Tidak memperhitungkan ketinggian kamera.
- f. Intensitas cahaya tidak terlalu rendah.

1.4. Tujuan

Tujuan dari tugas akhir ini adalah mengembangkan aplikasi pengolahan citra digital untuk untuk alternatif baru di dalam sistem penuntun parkir.

1.5. Manfaat

Manfaatnya dari penulisan tugas akhir ini adalah:

- a. Bagi penulis
Mengembangkan ilmu pengetahuan yang diperoleh dan mempraktikkan untuk menyelesaikan permasalahan yang timbul dalam keadaan sebenarnya.
- b. Bagi masyarakat
Memberikan kemudahan bagi masyarakat agar dapat mengetahui keadaan di dalam tempat parkir dalam suatu tempat dari luar..

1.6. Sistematika Penulisan

Sistematika penulisan dalam penulisan skripsi ini adalah sebagai berikut:

BAB I Pendahuluan

Memuat latar belakang, rumusan masalah, batasan masalah, tujuan, metodologi pembahasan, dan sistematika penulisan.

BAB II Dasar teori

Membahas teori-teori yang mendukung dalam perancangan dan pembuatan aplikasi.

BAB III Metodologi

Berisi tentang metode penelitian yang digunakan dalam perancangan dan pengujian sistem.

BAB IV Analisis kebutuhan dan Perancangan

Membahas tentang analisa kebutuhan dari sistem dan kemudian merancang hal-hal yang berhubungan dengan analisa tersebut.

BAB V Implementasi

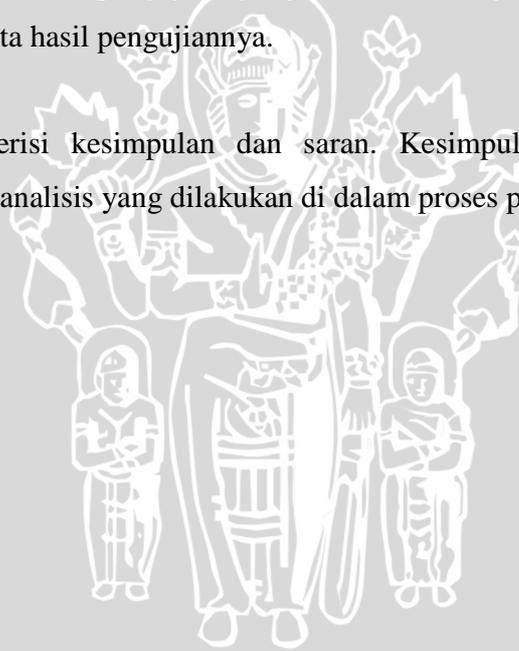
Bagian ini berisi penjelasan tentang implemetasi yang telah dilakukan (Sistem operasi, perangkat keras dan bahasa pemrograman yang digunakan) dan batasan-batasan implementasi.

BAB VI Pengujian dan Analisis

Bagian ini berisi penjelasan tentang strategi pengujian (unit, integrasi dan validasi) dan teknik pengujian yang dilakukan. Dijelaskan juga seluruh kasus uji beserta hasil pengujiannya.

BAB VII Penutup

Bagian ini berisi kesimpulan dan saran. Kesimpulan didasarkan atas pengujian dan analisis yang dilakukan di dalam proses penelitian.



BAB II

DASAR TEORI

2.1. Pengolahan Citra

Informasi adalah salah satu dari kebutuhan dasar manusia. Dalam perkembangannya, informasi bisa di kelompokkan menjadi bermacam-macam jenis. Salah satu dari jenis informasi adalah gambar. Gambar memiliki keuntungan dibanding bentuk informasi teks. Sebuah gambar dapat menjelaskan lebih banyak daripada berlembar-lembar tulisan. Dalam perkembangannya, gambar bukan lagi sekedar penyampai informasi, tetapi dapat juga menjadi penarik perhatian. Karena penggunaan gambar membawa keuntungan dalam penyajian informasi, media-media yang dahulu hanya menggunakan teks mulai menambahkan gambar.

Seperti layaknya bentuk informasi yang lain, gambar terkadang perlu di olah terlebih dahulu sebelum disajikan. Saat ini media-media informasi, dalam bentuk penyajian apapun, banyak menggunakan teknologi digital dalam pengolahan informasinya, termasuk didalamnya pengolahan gambar.

Bidang-bidang yang menunjang pengolahan citra adalah komputer grafik dan sistem visual. Komputer grafik melalui pemrograman grafik menghasilkan citra dari bentuk geometri yang primitif seperti titik, garis lurus dan garis lengkung, lingkaran dan bentuk-bentuk dasar geometri lainnya. Komputer grafik memainkan peranan penting dalam visualisasi dan *virtual-reality*. Sedangkan sistem visual bekerja sebaliknya, menduga bentuk geometri primitif dan ciri lainnya yang merupakan penyederhanaan dari citra asal yang sifatnya lebih kompleks. Jadi komputer grafik memadukan unsur-unsur pembentuk citra untuk membentuk atau mensintesa citra sedangkan sistem visual mengerjakan hal kebalikannya yaitu menganalisa citra dan terkadang menguraikannya menjadi bentuk-bentuk yang lebih sederhana, agar dapat dinilai secara kuantitatif

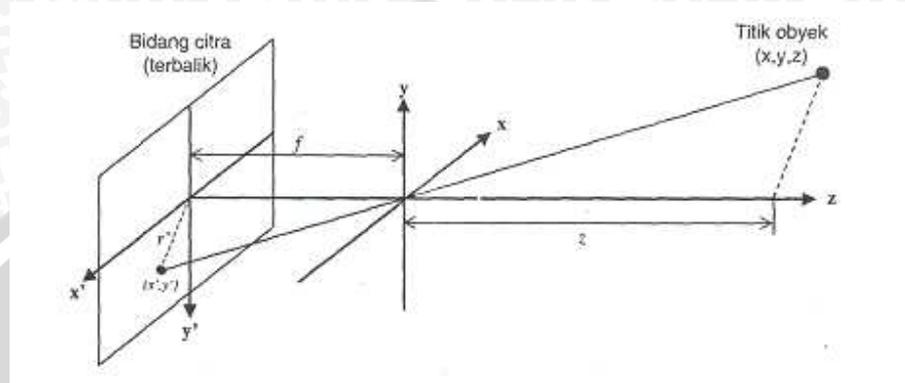
2.2. Geometri Citra

Untuk dapat mengerti dan melakukan operasi pengolahan citra, pertama kali harus dipahami dengan baik apa dan bagaimana sifat-sifat citra itu sendiri. Ada dua hal penting dan sangat mendasar pada proses pembentukan citra yang harus dipahami dan selanjutnya sangat perlu untuk terus diingat, yaitu:

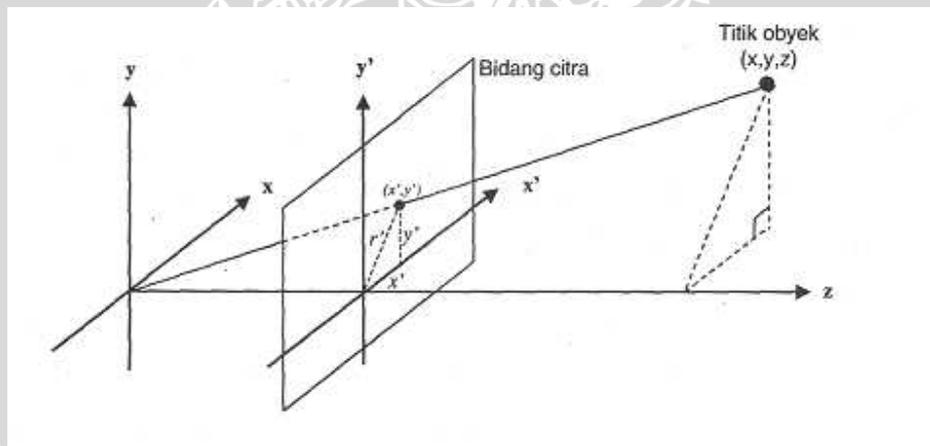
- Geometri formasi citra yang menentukan lokasi suatu titik dalam pemandangan yang diproyeksikan pada bidang citra, dan
- Fisik cahaya, yang menentukan kecerahan suatu titik pada bidang citra sebagai fungsi pencahayaan pemandangan dan sifat-sifat permukaan.

Untuk lebih memahami syarat pembentukan suatu citra di atas, bayangkanlah bahwa anda mempunyai sejumlah batu kerikil yang berwarna-warni. Bila anda sebarakan begitu saja sejumlah batu kerikil tadi dalam sebidang tanah, tidak akan terbentuk suatu citra yang bermakna. Tetapi bila anda menyusunnya sedemikian rupa, sehingga masing-masing kerikil mempunyai posisi tertentu, mungkin akan terbentuk suatu citra yang bermakna. Demikian pula bila susunannya diubah mengikuti pola yang lain, mungkin akan terbentuk citra yang lain pula. Itulah pentingnya formasi atau susunan titik-titik pembentuk citra dengan lokasinya masing-masing.

Model dasar untuk proyeksi suatu titik pada pemandangan ke dalam bidang citra diperlihatkan pada Gambar 2.1. Pada model ini pusat proyeksi sistem pembentukan citra berpotongan dengan titik awal dari koordinat sistem tiga-dimensi, yaitu x , y dan z . Posisi horizontal diwakili oleh x , posisi vertikal oleh y dan jarak dari kamera ke suatu titik obyek oleh z . Garis pandangan dari suatu titik dalam pemandangan adalah sebuah garis yang menyentuh titik tersebut dan titik pusat proyeksi, sedangkan jarak dari titik ke kamera dinyatakan dengan z , yang sejajar dengan sumbu z . Dalam sistem optik pada kamera yang sesungguhnya, citra hasil pembentukan berada di belakang pusat proyeksi dengan jarak. Untuk memudahkan, dalam ilustrasi ini diasumsikan bahwa bidang citra berada di depan pusat proyeksi seperti diperlihatkan pada Gambar 2.2.



Gambar 2.1 Diagram proyeksi pembentukan citra di belakang pusat proyeksi



Gambar 2.2 Diagram proyeksi pembentukan citra di depan pusat proyeksi

Jarak suatu titik (x,y,z) dalam pandangan dari sumbu z dinyatakan dengan $r=\sqrt{x^2+y^2}$, sedangkan jarak titik hasil proyeksi pada citra (x',y') dinyatakan dengan $r'=\sqrt{x'^2+y'^2}$. Selanjutnya terdapat hubungan:

$$\frac{f}{z} = \frac{r'}{r} \quad (2.2.1)$$

$$\frac{x'}{x} = \frac{y'}{y} = \frac{r'}{r} \quad (2.2.2)$$

Penggabungan persamaan (2.2.1) dan persamaan (2.2.2) menghasilkan:

$$\frac{x'}{x} = \frac{f}{z} \text{ dan } \frac{y'}{y} = \frac{f}{z} \quad (2.2.3)$$

Dengan demikian posisi suatu titik di dalam citra diberikan dengan persamaan berikut:

$$x' = \frac{f}{z} x \quad (2.2.4)$$

$$y' = \frac{f}{z} y \quad (2.2.5)$$

Uraian di atas mengasumsikan bahwa pusat proyeksi mempunyai titik pertemuan pada titik awal dari sumbu tiga-dimensi. Dalam operasi pengolahan citra pada sistem visual titik awal bidang citra terletak pada posisi kiri atas. Dengan demikian koordinat pada bidang citra merupakan koordinat absolut. Ini perlu diingat dengan baik karena berbeda dengan koordinat yang dipakai pada ilmu matematika yang sudah dipahami dengan baik, yaitu titik awal suatu grafik berada pada sudut kiri bawah.

2.3. Euclidean Distance

Jarak Euclidean dapat dianggap sebagai jarak yang paling pendek antara dua titik dan pada dasarnya sama halnya dengan persamaan Pythagoras ketika digunakan didalam dua dimensi. Secara matematis dapat dituliskan didalam persamaan (1) berikut

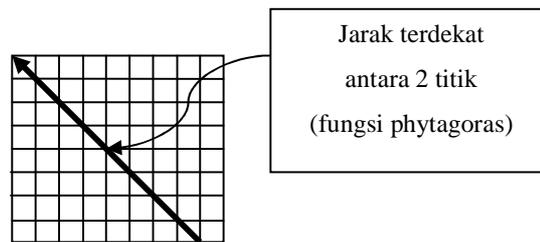
$$d(\mathbf{i}, \mathbf{j}) = \sqrt{|\mathbf{x}_{i1} - \mathbf{x}_{j1}|^2 + |\mathbf{x}_{i2} - \mathbf{x}_{j2}|^2 + \dots + |\mathbf{x}_{in} - \mathbf{x}_{jn}|^2}$$

Dimana $d(i, j)$ = jarak antara 2 titik

X_{in} = koordinat titik pertama

X_{jn} = koordinat titik kedua

Gambar 2.3 merupakan representasi dari jarak terdekat dari dua titik :



Gambar 2.3 Fungsi Euclidian

2.4. Delphi

Borland Delphi atau disebut Delphi merupakan perangkat lunak pengembangan aplikasi di lingkungan Windows. Perangkat lunak ini dapat digunakan untuk membuat aplikasi dari permainan basis data hingga pembuatan citra digital.

Ide munculnya dhelpi sebenarnya berasal dari bahasa pemrograman yang cukup terkenal, yaitu Pascal. Bahasa Pascal sendiri telah diciptakan pada tahun 1971 oleh ilmuwan dari swiss, yaitu niklaus wirth. Nama pascal diambil dari ahli matematika dan filsafat dari Prancis, yaitu Blaise Pascal (1623-1662).

Sejak itu muncul beberapa versi Pascal diantaranya Turbo Pascal yang dirilis oleh Borland Internasional tahun 1983. Turbo Pascal yang muncul pertama kali hanya bisa dijalankan di sistem operasi DOS. Namun pada perkembangan selanjutnya Turbo Pascal juga bisa berjalan di Windows 3.x yaitu Turbo Pascal For Windows.

Karena pemrograman Windows dengan Turbo Pascal dirasa cukup sulit, akhirnya Borland internasional mengembangkan bahasa Pascal yang bersifat visual. Hasil dari pengembangan ini adalah di rilisnya dhelpi 1. Perkembangan delphi tidak

berhenti sampai disitu dalam tahun-tahun berikutnya borland merilis beberapa versi pengembangan dhelpi. Sampai Delphi 7 dan Delphi 8, dalam pembuatan program citra digital disini digunakan Delphi 7 sebagai software pendukung.

2.5 Mobil

Benda yang dideteksi dalam aplikasi penuntun parkir dengan menggunakan metode pengolahan citra digital ini adalah mobil. Banyak definisi yang berhubungan dengan mobil, oleh sebab itu perlu diketahui bagaimana definisi kendaraan bermotor secara umum dan mobil secara khusus.

Kendaraan bermotor adalah kendaraan yang digerakkan oleh peralatan teknik yang berada pada kendaraan itu. Dalam hal ini, terlalu luas pengertian antara kendaraan bermotor dan mobil yang dipakai sebagai studi kasus dalam pengerjaan tugas akhir ini. Mobil penumpang adalah setiap kendaraan bermotor yang dilengkapi sebanyak-banyaknya 8 (delapan) tempat duduk tidak termasuk tempat duduk pengemudi, baik dengan maupun tanpa perlengkapan pengangkutan bagasi, sehingga bisa dimungkinkan untuk memakai fasilitas parkir di dalam gedung. Selain itu, ada klasifikasi lain dari mobil yaitu mobil barang dimana mobil barang adalah setiap kendaraan bermotor selain dari yang termasuk dalam klasifikasi sepeda motor, mobil penumpang maupun mobil bus. Kendaraan khusus adalah kendaraan bermotor selain daripada kendaraan bermotor untuk penumpang dan kendaraan bermotor untuk barang, yang penggunaannya untuk keperluan khusus atau mengangkut barang-barang khusus.

Penjelasan masalah spesifikasi minimum mobil untuk tempat parkir akan dijelaskan pada sub-bab berikutnya.

2.6 Tempat Parkir

Parkir adalah keadaan tidak bergerak suatu kendaraan yang tidak bersifat sementara. Berhenti adalah keadaan tidak bergerak suatu kendaraan untuk sementara dengan pengemudi tidak meninggalkan kendaraan. Fasilitas parkir adalah lokasi yang ditentukan sebagai tempat pemberhentian kendaraan yang tidak bersifat sementara untuk melakukan kegiatan pada suatu kurun waktu. Tempat parkir di badan jalan, (on

street parking) adalah fasilitas parkir yang menggunakan tepi jalan. Fasilitas parkir di luar badan jalan (off street parking) adalah fasilitas parkir kendaraan di luar tepi jalan umum yang dibuat khusus atau penunjang kegiatan yang dapat berupa tempat parkir dan/atau gedung parkir. Jalan adalah tempat jalan yang diperuntukan bagi lalu lintas umum.

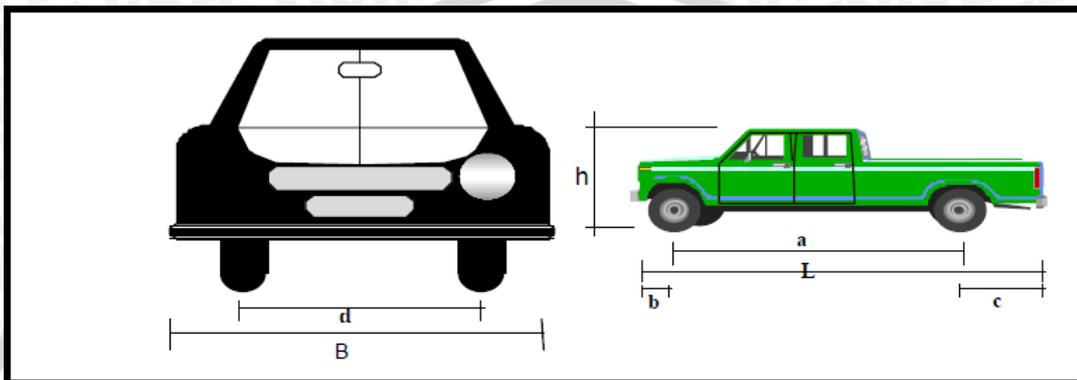
Satuan ruang parkir (SRP) adalah ukuran luas efektif untuk meletakkan kendaraan (mobil penumpang, bus/truk, atau sepeda motor), termasuk ruang bebas dan lebar buka pintu. Untuk hal-hal tertentu bila tanpa penjelasan, SRP adalah SRP untuk mobil penumpang. Jalur sirkulasi adalah tempat, yang digunakan untuk pergerakan kendaraan yang masuk dan keluar dari fasilitas parkir. Jalur gang merupakan jalur antara dua deretan ruang parkir yang berdekatan. Kawasan parkir adalah kawasan atau areal yang memanfaatkan badan jalan sebagai fasilitas parkir dan terdapat pengendalian parkir melalui pintu masuk.

Berdasar pedoman teknis penyelenggaraan fasilitas parkir yang telah ditetapkan oleh departemen perhubungan Direktur Jendral perhubungan darat, terdapat beberapa ukuran kebutuhan ruang parkir, yang ada pada tabel berikut.

Tabel 1.1 Ukuran Kebutuhan Tempat Parkir

Peruntukan	Satuan (SRP untuk mobil penumpang)	Kebutuhan Ruang Parkir
Pusat Perdagangan		
· Pertokoan	SRP / 100 m ² luas lantai efektif	3,5-7,5
· Pasar Swalayan	SRP / 100 m ² luas lantai efektif	3,5-7,5
· Pasar	SRP / 100 m ² luas lantai efektif	3,5-7,5
Pusat Perkantoran		
· Pelayanan bukan umum	SRP / 100 m ² luas lantai	1,5-3,5
· Pelayanan umum	SRP / 100 m ² luas lantai	1,5-3,5
Sekolah	SRP / mahasiswa	0,7 - 1,0
Tempat Penginapan	SRP / kamar	0,2 - 1,0
Rumah Sakit	SRP / tempat tidur	0,2 - 1,3
Bioskop	SRP / tempat duduk	0,1 - 0,4

Penentuan satuan ruang parkir atau SRP, yang telah dijelaskan pada penjelasan sebelumnya, mempunyai beberapa ketentuan. Ketentuan SRP untuk mobil dijelaskan pada gambar berikut :



Gambar 2.4 Dimensi Kendaraan Standar untuk Mobil Penumpang

a = jarak gandar

h = tinggi total

b = depan tergantung

B = lebar total

c = belakang tergantung

L = panjang total

d = lebar

Ruang bebas kendaraan parkir diberikan pada arah lateral dan longitudinal kendaraan. Ruang bebas arah lateral ditetapkan pada saat posisi pintu kendaraan dibuka, yang diukur dari ujung terluar pintu ke badan kendaraan parkir yang ada di sampingnya. Ruang bebas ini diberikan agar tidak terjadi benturan antara pintu kendaraan dan kendaraan yang parkir di sampingnya pada saat penumpang turun dari kendaraan.

Ruang bebas arah memanjang diberikan di depan kendaraan untuk menghindari benturan dengan dinding atau kendaraan yang lewat jalur gang (aisle).

Jarak bebas arah lateral diambil sebesar 5 cm dan jarak bebas arah longitudinal sebesar 30 cm.

Ukuran lebar bukaan pintu merupakan fungsi karakteristik pemakai kendaraan yang memanfaatkan fasilitas parkir. Sebagai contoh, lebar bukaan pintu kendaraan karyawan kantor akan berbeda dengan lebar bukaan pintu kendaraan pengunjung pusat kegiatan perbelanjaan. Dalam hal ini, karakteristik pengguna kendaraan yang memanfaatkan fasilitas parkir dipilih menjadi tiga seperti pada tabel berikut.

Tabel 1.2 Lebar Bukaan Pintu Kendaraan

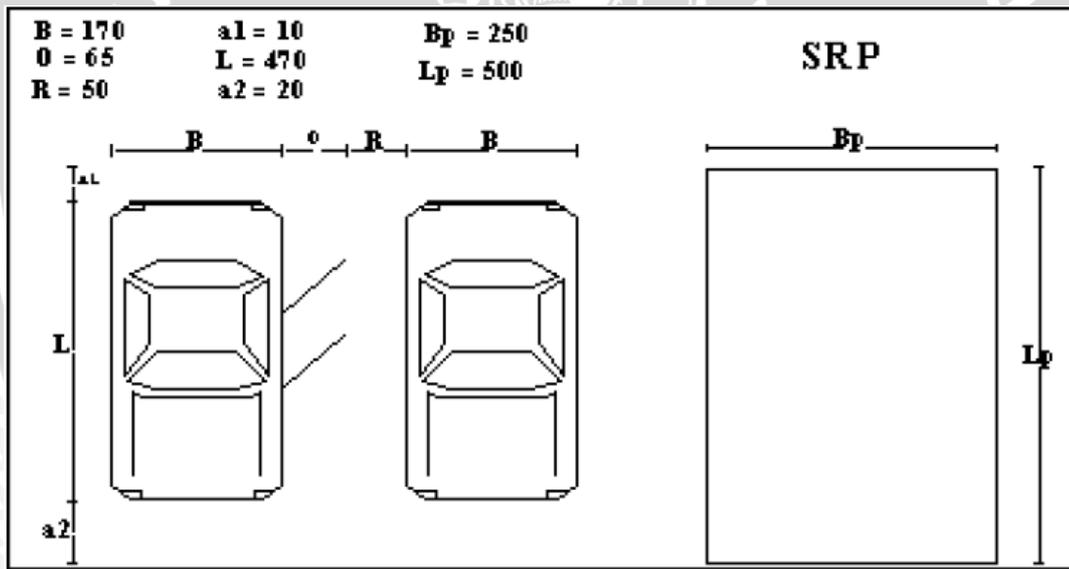
Jenis Bukaan Pintu	Pengguna dan/atau Peruntukan Fasilitas Parkir	Golongan
Pintu depan/belakang terbuka tahap awal 55 cm.	<ul style="list-style-type: none"> · Karyawan/pekerja kantor · Tamu/pengunjung pusat kegiatan perkantoran, perdagangan, pemerintahan 	I
Pintu depan/belakang terbuka penuh 75 cm	Pengunjung tempat olahraga, pusat hiburan/rekreasi, hotel, pusat perdagangan eceran/swalayan, rumah sakit, bioskop	II
Pintu depan terbuka penuh dan ditambah untuk pergerakan kursi roda	Orang cacat	III

Berdasarkan atas 2 tabel di atas penentuan SRP untuk mobil penumpang diklasifikasikan menjadi tiga golongan, seperti yang ditunjukkan pada tabel berikut.

Tabel 1.3 Penentuan Satuan Ruang Parkir

Jenis Kendaraan	Satuan Ruang Parkir (m ²)
1. a. Mobil penumpang untuk golongan I b. Mobil penumpang untuk golongan II c. Mobil penumpang untuk golongan III	2,30 x 5,00 2,50 x 5,00 3,00 x 5,00
2. Bus/truk	3,40 x 12,50
3. Sepeda motor	0,75 x 2,00

Besar satuan ruang parkir untuk mobil seperti ditunjukkan pada gambar sebagai berikut.



Gambar 2.5 Satuan Ruang Parkir (SRP) untuk Mobil Penumpang (dalam cm)

Keterangan :

- B = lebar total kendaraan
- O = lebar bukaan pintu
- R = jarak bebas arah lateral
- L = panjang total kendaraan
- a1, a2 = jarak bebas arah longitudinal

Gol I :

$$B = 170$$

$$a_1 = 10$$

$$B_p = 230 = B + O + R$$

$$O = 55$$

$$L = 470$$

$$L_p = 500 = L + a_1 + a_2$$

$$R = 5$$

$$a_2 = 20$$

Gol II :

$$B = 170$$

$$a_1 = 10$$

$$B_p = 250 = B + O + R$$

$$O = 75$$

$$L = 470$$

$$L_p = 500 = L + a_1 + a_2$$

$$R = 5$$

$$a_2 = 20$$

Gol III :

$$B = 170$$

$$a_1 = 10$$

$$B_p = 300 = B + O + R$$

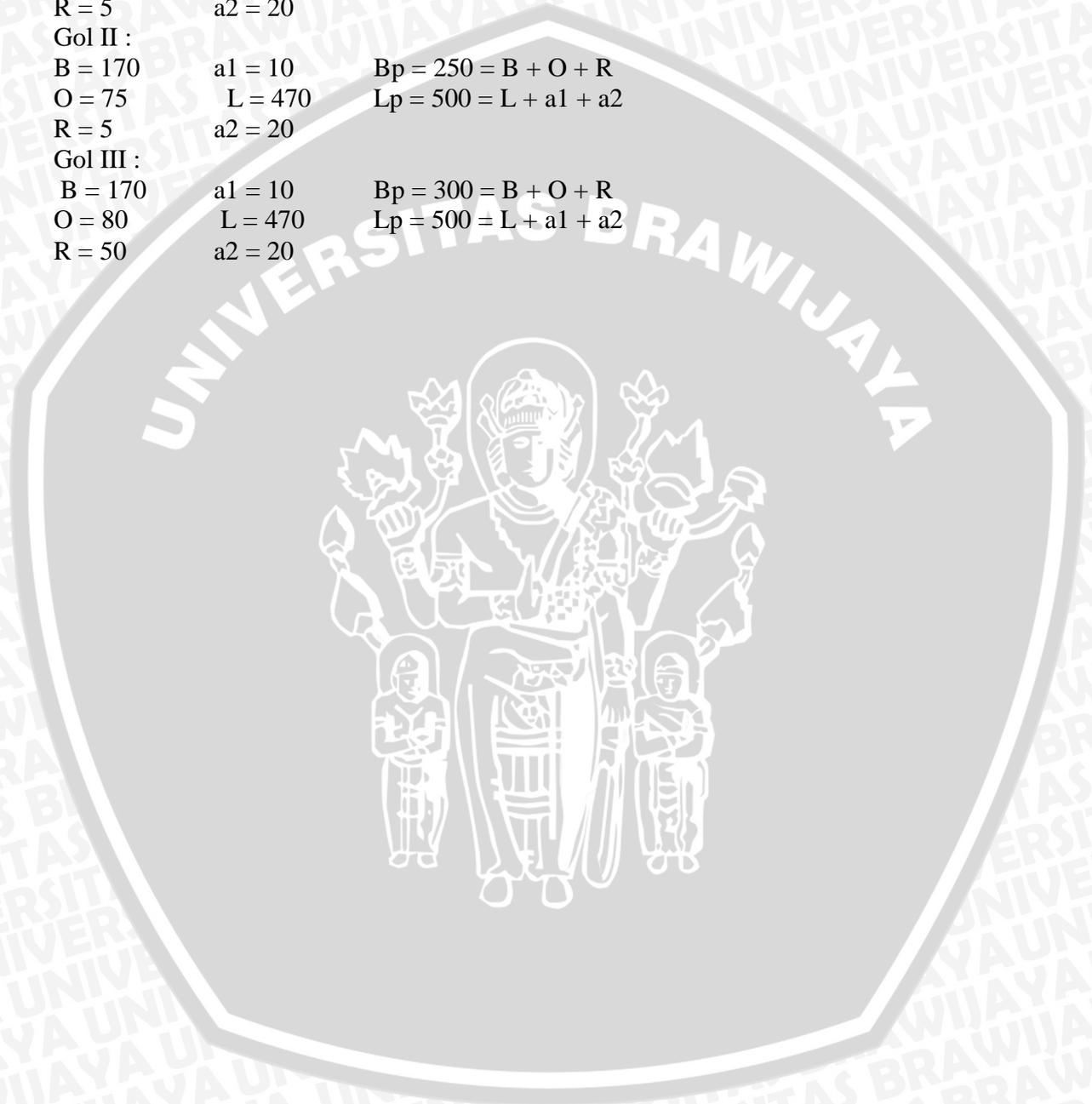
$$O = 80$$

$$L = 470$$

$$L_p = 500 = L + a_1 + a_2$$

$$R = 50$$

$$a_2 = 20$$



BAB III

METODE PENELITIAN

Metodologi penelitian merupakan tahap-tahap penelitian yang dilakukan untuk memperoleh pemecahan masalah. Metodologi yang digunakan dalam penyusunan tugas akhir antara lain: studi literatur, analisis dan perancangan, implementasi, pengujian, dan pengambilan kesimpulan..

3.1. Studi Literatur

Melakukan studi literatur yaitu dengan pengumpulan dan pendalaman bahan pustaka yang dibutuhkan dalam penyelesaian tugas akhir ini. Bahan pustaka tersebut membahas tentang:

1. Dasar image processing.
2. Citra digital dengan menggunakan delphi.
3. Sistem deteksi obyek.

3.2. Penentuan Analisis dan Perancangan

Pada analisis dan perancangan terdiri dari dua tahap yaitu analisis kebutuhan perangkat lunak yang mencakup spesifikasi kebutuhan sistem, serta tahap kedua adalah perancangan perangkat lunak sistem penuntun parkir dengan pengolahan citra digital dengan menggunakan Delphi.

Untuk merancang software citra digital ini, perlu ditentukan spesifikasi perangkat keras dan perangkat lunak yang digunakan. Spesifikasinya adalah sebagai berikut:

Perangkat Keras:

- 1 unit PC dengan spesifikasi tertentu, yang digunakan sebagai pembuat perangkat lunak citra digital.
- Kamera (kamera digital, kamera ponsel atau kamera web)

- Scanner (dipakai untuk memasukkan data gambar kedalam pc dalam pengujian secara manual)
- Monitor (digunakan untuk menampilkan data hasil pengolahan citra)

Perangkat Lunak:

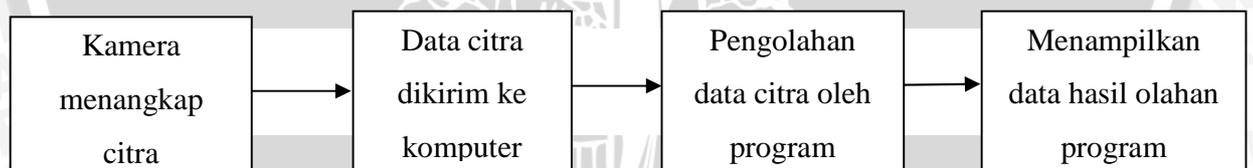
- Dhelphi 7

3.3. Gambaran Umum Sistem

Perancangan sistem ini diperlukan untuk mengetahui sistem yang digunakan dalam tugas akhir ini dengan pengolahan citra digital dan juga alat bantu yang digunakan.

- Kamera : Alat yang digunakan untuk mengambil citra/obyek.
- Program : Perangkat lunak yang digunakan untuk memanipulasi citra di dalam komputer
- Board : Alat atau monitor yang digunakan untuk menampilkan hasil dari manipulasi citra digital

3.4. Block Diagram Sistem



Gambar 3.1 Proses Kerja Sistem

3.5. Implementasi

Pada proses implementasi, dilakukan pembuatan lingkungan implementasi untuk melakukan spesifikasi sistem. Kegiatan yang dilakukan adalah menerjemahkan dari bentuk perancangan yang ada dalam bentuk program Delphi.

3.6. Pengujian

Pengujian dilakukan untuk menemukan kesalahan aplikasi yang dibuat, sehingga sebelum aplikasi yang dibuat digunakan oleh pengguna aplikasi ini diharapkan dapat sedikit mungkin atau bahkan tidak ada kesalahan pada saat digunakan nantinya. Pengujian yang dilakukan pada benda, kemudian di ambil gambarnya dan di uji dengan menggunakan software citra digital yang telah dibuat

3.7. Pengambilan Kesimpulan dan Saran

Berisi kesimpulan-kesimpulan yang diperoleh dari hasil perancangan, implementasi, dan pengujian untuk menjawab perumusan masalah yang telah disebutkan pada bab pendahuluan serta saran yang menurut penulis sangat diperlukan untuk pengembangan aplikasi ini pada masa yang akan datang.



BAB IV

PERANCANGAN

Perancangan perangkat lunak. dilakukan untuk membahas langkah-langkah yang dilakukan seperti analisis kebutuhan, perancangan kerja, prinsip kerja program untuk dilanjutkan pada bab-bab berikutnya. Pada bab ini juga membahas proses deteksi obyek yang menjadi pokok utama, metode untuk mendapatkan intensitas warna, dan penulisan informasi intensitas warna dalam penulisan tugas akhir ini.

4.1 Analisis Kebutuhan

Dalam tahap ini, yang perlu dilakukan adalah mengidentifikasi tiap kebutuhan informasi yang diberikan pada para pemakai parkir atau orang yang mengoperasikan program ini. Dengan adanya analisis kebutuhan ini diharapkan dapat menampilkan hasil dari tempat parkir yang terisi atau kosong, sehingga para pemakai parkir dapat memilih slot mana yang bisa digunakan untuk parkir.

Program ini dibentuk untuk memberikan informasi mengenai pemakaian ruang parkir yang kosong, sehingga bisa digunakan. Data yang digunakan adalah data berdasarkan hasil *capture* kamera dan pengolahan intensitas warna.

4.1.1 Perancangan Sistem

Perancangan sistem merupakan tahap awal dari perancangan perangkat lunak. Perancangan ini dilakukan untuk mengetahui aplikasi sistem yang dibuat secara umum. Perancangan sistem ini didahului dengan pendefinisian pelaku atau user yang menggunakan program penuntun parkir dengan teknik pengolahan citra dan juga alat bantu yang digunakan, yaitu:

1. User : Pelaku yang mengambil citra dan juga menjalankan program dengan tujuan untuk mengetahui dan mengecek tempat parkir yang kosong atau terisi oleh mobil

2. Kamera : Alat yang digunakan user untuk mengambil gambar dari tempat parkir.
3. Program : Perangkat lunak yang digunakan untuk mengetahui jumlah dan posisi masing-masing mobil.
4. Media pendukung lain yang selanjutnya dijelaskan pada sub-bab berikut.

Diagram konteks adalah sebuah diagram sederhana yang menggambarkan hubungan dengan entitas luar, masukan dan keluaran dari sistem. Diagram konteks aplikasi penuntun parkir dengan metode pengolahan citra digital seperti ditunjukkan pada gambar 4.1.

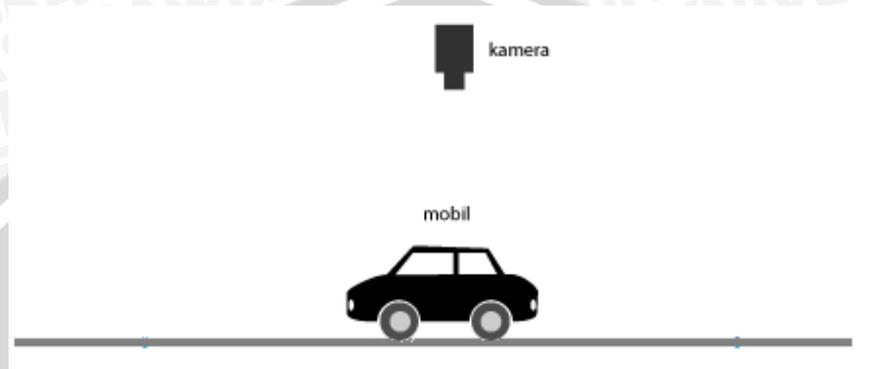


4.1.3 Cara Kerja Sistem

Cara kerja dari aplikasi penuntun parkir dengan pengolahan citra digital seperti ditunjukkan adalah sebagai berikut:

1. User atau dalam hal ini adalah operator yang mengoperasikan program, menekan tombol capture pada form yang telah disediakan.
2. Gambar yang telah diambil oleh kamera digunakan sebagai masukan untuk perangkat lunak yang telah disediakan seperti ditunjukkan pada gambar 4.2 terhadap slot parkir yang ditunjukkan pada gambar 4.3. Untuk slot parkir pada aplikasi ini, dibatasi 6 slot dimana terdapat 2 kolom dan 3 baris slot parkir. Keterangan lebih lanjut untuk slot parkir ini dibahas pada penjelasan lebih lanjut.

3. Keluaran program adalah informasi parkir, dimana terdapat 2 kemungkinan yaitu, slot parkir telah terisi atau slot parkir masih kosong. Tambahan pada keluaran adalah besarnya intensitas warna pada slot parkir sehingga bisa diketahui pula besar kecil nya obyek yang menempati satu slot parkir.

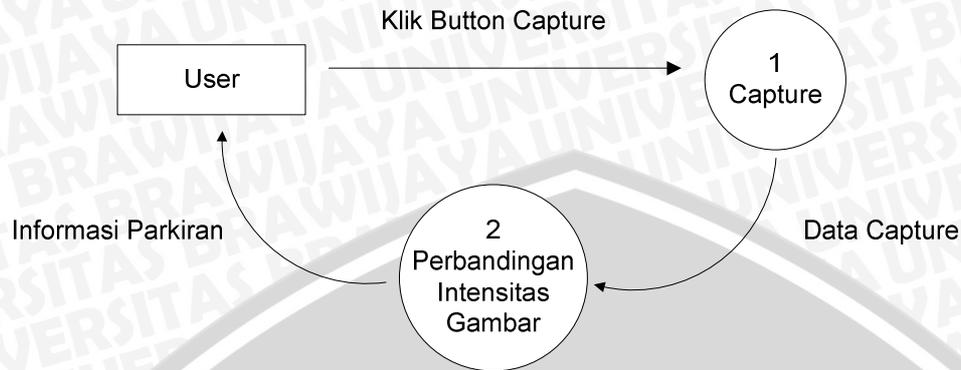


Proses Pengambilan Gambar dengan Menggunakan Kamera tampak samping

Slot parkir yang menjadi simulasi pada aplikasi penuntun parkir dengan metode pengolahan citra digital ini seperti ditunjukkan pada gambar berikut ini

4.2 Data Flow Diagram

Untuk menelusuri bagaimana masukan dan keluaran dari tiap proses yang menggunakan aplikasi penuntun parkir dengan pengolahan citra digital, menggunakan satu metode yaitu *Data Flow Diagram*(DFD). DFD untuk aplikasi penuntun parkir dengan pengolahan citra digital seperti ditunjukkan pada gambar 4.3.

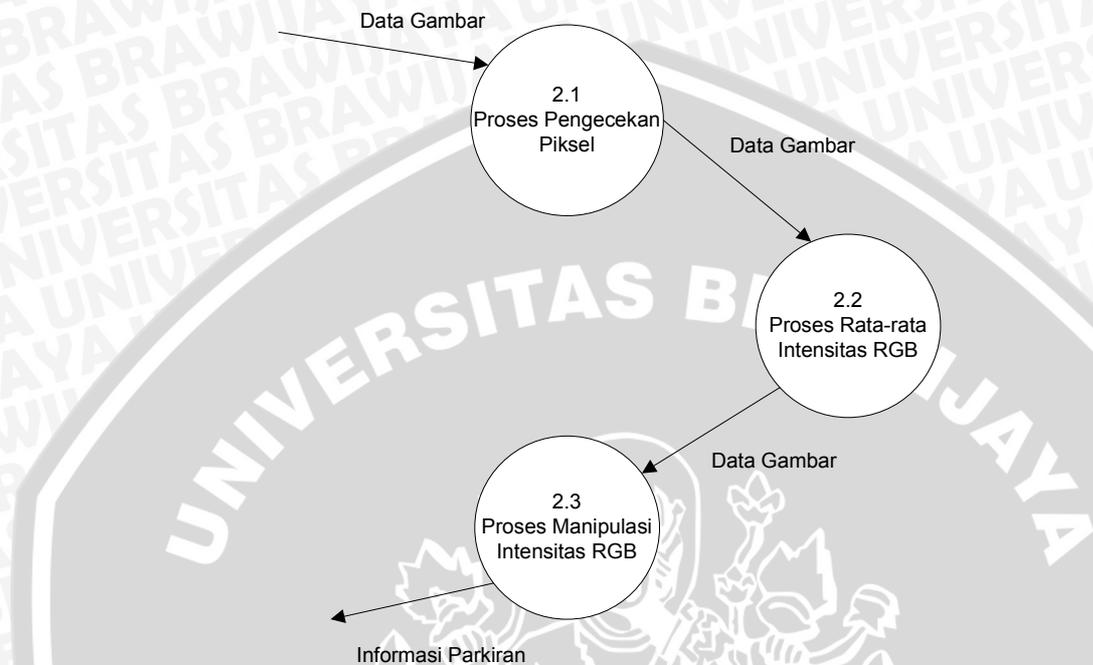


Data Flow Diagram level 1 Aplikasi Penuntun Parkir dengan pengolahan Citra Digital

Pada *Data Flow Diagram Level 1* aplikasi penuntun parkir dengan pengolahan citra digital ini terdiri dari dua proses yaitu:

1. Proses Capture yaitu proses pengambilan gambar oleh program dengan indikasi adanya klik tombol capture pada form yang telah disediakan.
2. Proses Perbandingan Intensitas Gambar yaitu proses pengolahan data capture yang pada akhirnya digunakan untuk memberikan informasi pada user tentang keberadaan slot parkir. Proses ini melibatkan pengecekan intensitas 3 warna dasar yaitu *Red*, *Green*, dan *Blue*.

4.2.1 Data Flow Diagram level 2 Proses Perbandingan Intensitas Gambar



Gambar 4.4 Data Flow Diagram level 2 Proses Perbandingan Intensitas Gambar

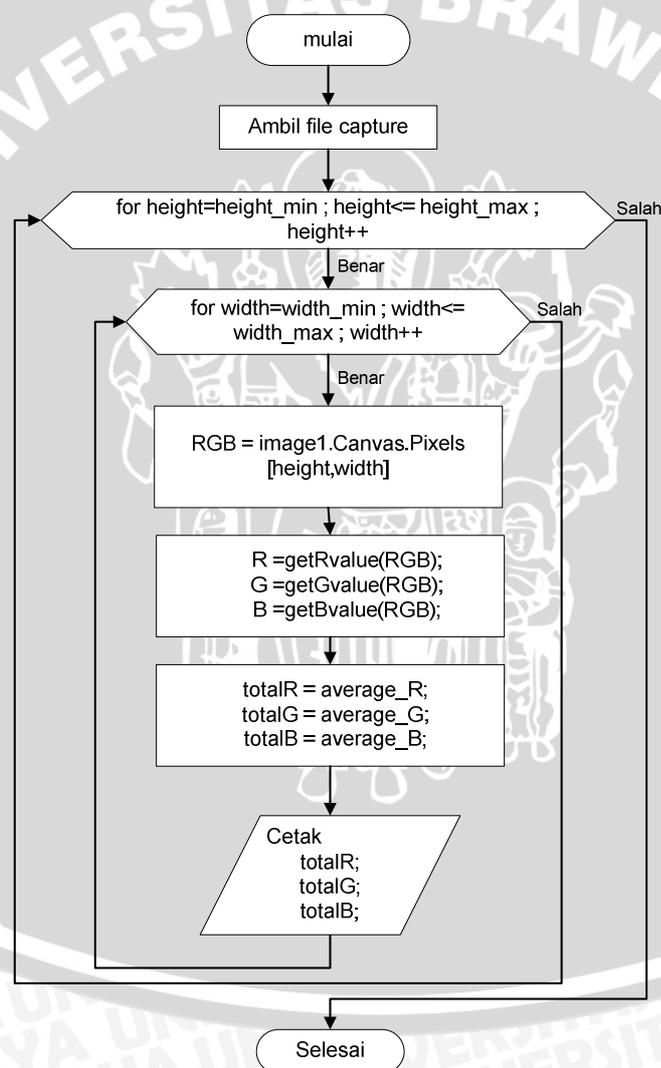
Pada *Data Flow Diagram level 2* Proses Perbandingan Intensitas Gambar ini terdiri dari 3 proses yaitu :

1. Proses Pengecekan Piksel yaitu proses untuk mengetahui intensitas per piksel pada gambar.
2. Proses Rata-rata Intensitas RGB yaitu proses untuk merata-ratakan nilai masing-masing masing intensitas *Red*, *Green* dan *Blue* pada tiap-tiap slot parkir.
3. Proses Manipulasi Intensitas RGB yaitu proses untuk memanipulasi perhitungan rata-rata intensitas yang telah didapatkan sebelumnya. Proses perhitungan ini yang pada akhirnya digunakan untuk menentukan keluaran keseluruhan dari seluruh proses.

4.3 Diagram Alir Sistem

Untuk mengetahui jalannya program secara detail, dalam suatu proses di dalam aplikasi penuntun parkir dengan pengolahan citra digital ini, menggunakan satu metode yang umum dipakai dalam bentuk pemrograman sederhana maupun yang rumit yaitu Diagram Alir sistem. Diagram Alir Sistem untuk aplikasi penuntun parkir dengan pengolahan citra digital ditunjukkan pada sub-bab sub-bab berikut.

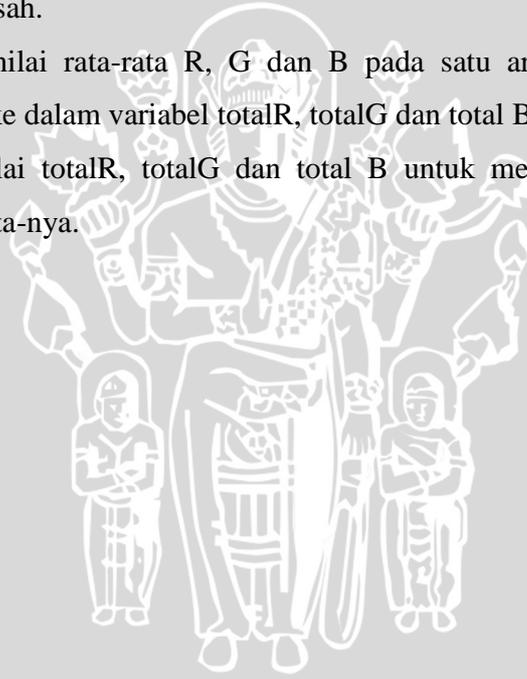
4.3.1 Diagram Alir untuk Proses Pengecekan Pixel



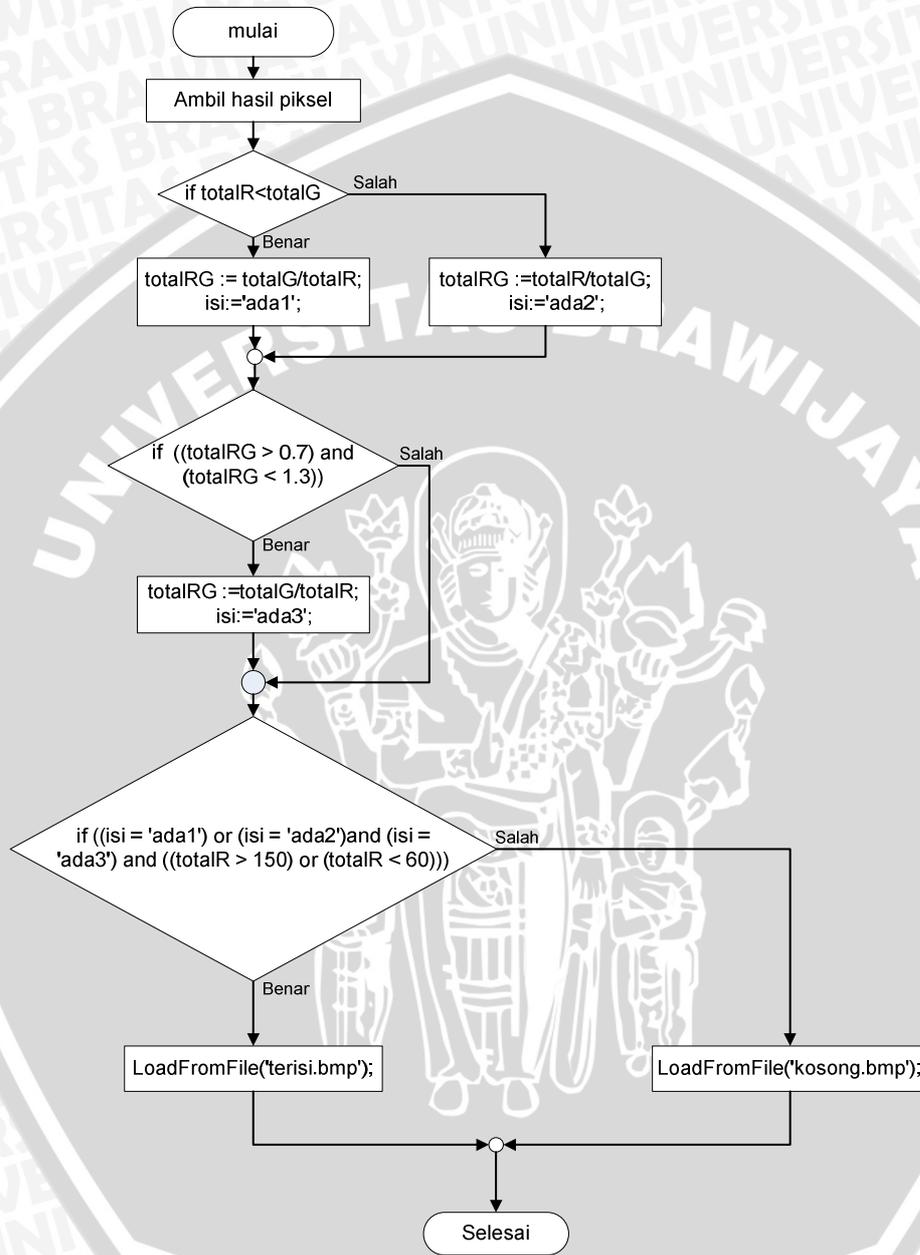
Gambar 4.5 Diagram Alir untuk Proses Pengecekan Pixel

Algoritma untuk proses pengecekan piksel pada gambar 4.4, dijelaskan pada point-point berikut:

1. Mengambil file hasil capture
2. Menentukan koordinat *height* untuk 1 *coverage area*, dengan batas *height_min* dan *height_max*.
3. Menentukan koordinat *width* dengan batas *width_min* dan *width_max*.
4. Nilai *height* dan *width* tersebut disimpan dalam satu variabel matriks bernama RGB yang dikemas dalam fungsi yang telah disediakan oleh Delphi yaitu *.canvas
5. Proses selanjutnya adalah memisahkan nilai R, G, dan B dalam suatu variabel terpisah.
6. Mengambil nilai rata-rata R, G dan B pada satu area. Masing-masing dimasukkan ke dalam variabel totalR, totalG dan total B.
7. Mencetak nilai totalR, totalG dan total B untuk mengetahui kesahihan kebenaran data-nya.



4.3.2 Diagram Alir untuk Proses Manipulasi Intensitas RGB



Gambar 4.6 Diagram Alir untuk Proses Manipulasi Intensitas RGB

Algoritma untuk proses Manipulasi Intensitas RGB pada gambar 4.5, dijelaskan pada point-point berikut:

1. Mengambil file hasil pengecekan piksel
2. Membandingkan antara nilai dari totalR dan total G.
3. Jika nilai totalR kurang dari totalG, maka variabel totalRG adalah totalR dibagi totalG. Indikasi dari keadaan ini dimasukkan ke dalam variabel isi dan mempunyai nilai 'ada1'.
4. Pada keadaan nilai totalR lebih besar dari totalG, maka variabel totalRG adalah totalG dibagi totalR. Indikasi dari keadaan ini dimasukkan ke dalam variabel isi dan mempunyai nilai 'ada2'.
5. Variabel totalRG mempunyai nilai ambang, yang ditentukan dengan mengambil nilai sedikit dibawah 1 dan sedikit diatas 1. Dalam hal ini ditentukan untuk nilai ambangnya adalah 0,7 sampai 1,3.
6. Jika keadaan ini bernilai benar, maka nilai totalRG langsung diisi dengan totalG dibagi totalR. Indikasi dari keadaan ini dimasukkan ke dalam variabel isi dan mempunyai nilai 'ada3'.
7. Jika kondisi memenuhi nilai isi salah satu dari 'ada1' atau 'ada2' dan nilai isi adalah 'ada3' dan nilai totalR adalah lebih besar dari 150 atau nilai totalR adalah kurang dari 60, maka ditampilkan keadaan terisi. Jika kondisi tersebut tidak terpenuhi, maka ditampilkan keadaan kosong.

BAB V

IMPLEMENTASI DAN PENGUJIAN

Bab ini membahas mengenai implementasi dan pengujian perangkat lunak untuk penuntun parkir dengan teknik pengolahan citra digital. Berdasarkan hasil yang telah didapatkan dari proses perancangan perangkat lunak yang telah dibuat. Implementasi merupakan proses transformasi hasil perancangan perangkat lunak ke dalam kode (*coding*) sesuai dengan sintaks dari bahasa pemrograman yang digunakan. Pembahasan terdiri dari penjelasan tentang lingkungan implementasi, pengambilan gambar, algoritma implementasi dan beberapa kendala dalam implementasi. Pengujian adalah suatu metode untuk mengetahui bagaimana sistem ini berjalan dan juga untuk mengetahui kelemahannya

5.1 Lingkungan Implementasi

Sistem dibuat dengan menggunakan aplikasi pemrograman Borland Delphi. Sistem diimplementasikan dengan menggunakan spesifikasi sebagai berikut:

1. Perangkat Keras :

1.1. Komputer

Spesifikasi :

- Processor : AMD Athlon™ 64 processor 3000+, MMX, 1,8 GHz
- Memory : 1024 MB RAM
- VGA : ATI RADEON HD 4650 480K piksel kamera

1.2. Kamera

Spesifikasi :

- Model : PC Web Camera Sturdy
- Resolusi : 4000 x 3000 piksel
- Data Format : RGB 24
- Port : 2.0

- Iluminasi Min : 2.5 LUX
- Lensa : F 2.2/f6.85
- Rasio S/N : > 48 db
- Sudut Pandang : $38,622^0$
- Jarak Fokus : 30 mm – tak terhingga

2. Perangkat Lunak :

2.1. Sistem operasi : Microsoft Windows XP SP3

2.2. Bahasa pemrograman: Borland Delphi 7

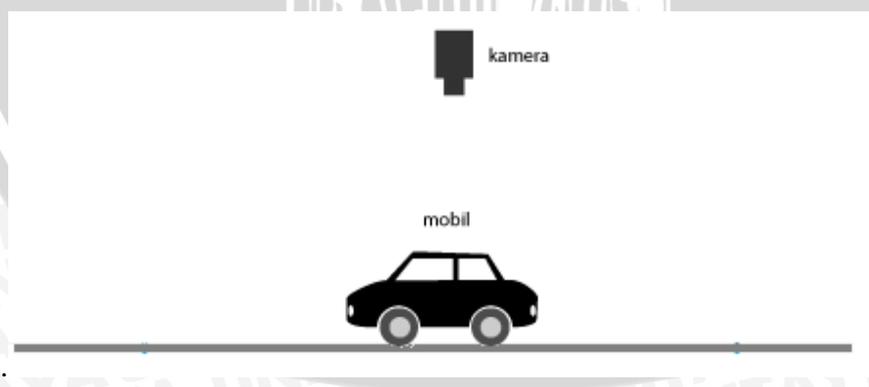
2.3. Perangkat lunak pendukung :

2.3.1 Microsoft Visio 2003

2.3.2 Microsoft Paint 5.1

5.2 Proses Pengambilan Gambar

Proses pengambilan gambar menggunakan kamera dilakukan pada media abu-abu yang telah diberi garis-garis pembatas dengan jarak tertentu dengan cara meletakkan posisi kamera tepat di atasnya. Kamera dihadapkan kepada titik tengah dari media, lalu kamera menyimpan keadaan bidang *capture*. Setelah selesai proses capture gambar, selanjutnya data disimpan di PC pada folder yang sama dengan program aplikasi ini PC memproses data sehingga didapatkan hasil untuk sistem penuntun parkir tersebut. Cara pengambilan gambar digambarkan seperti gambar 5.1.



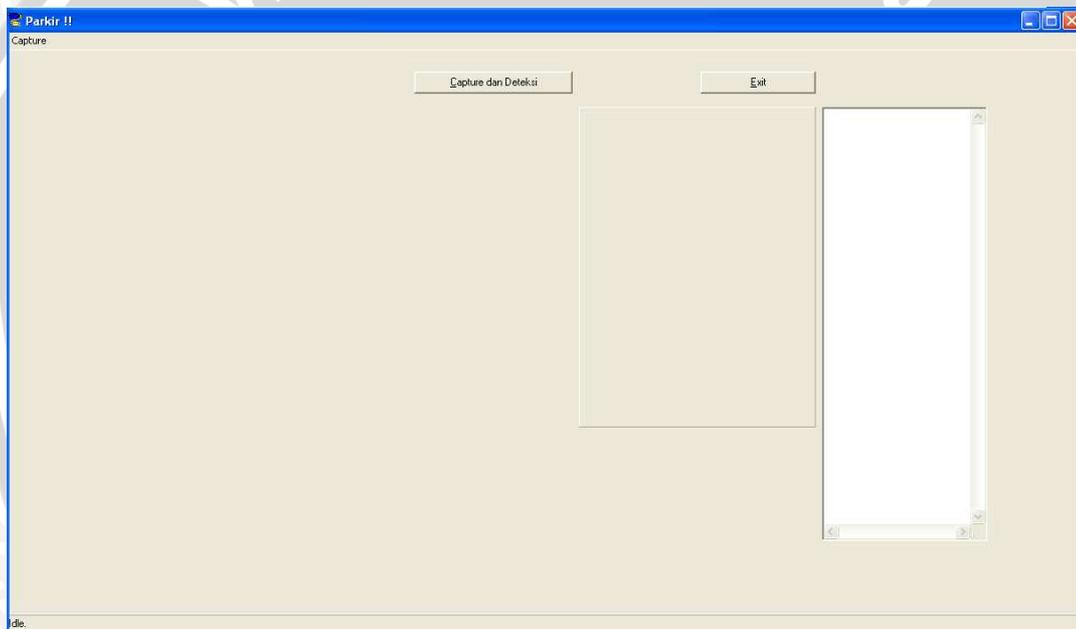
Gambar 5.1 Pengambilan Gambar dengan Menggunakan Kamera tampak samping

5.3 Tampilan

Agar para pengguna program dapat mengerti interface dari perangkat lunak untuk penuntun parkir dengan teknik pengolahan citra digital maka disertakan pula beberapa tampilan dasar dan tampilan pada saat berjalannya program. Tampilan ini juga digunakan untuk mendukung

5.3.1 Tampilan awal

Aplikasi ini mempunyai beberapa tampilan panel dan tampilan awal memo yang kosong. Tombol yang ada adalah tombol untuk menjalankan program dan tombol untuk keluar dari program.

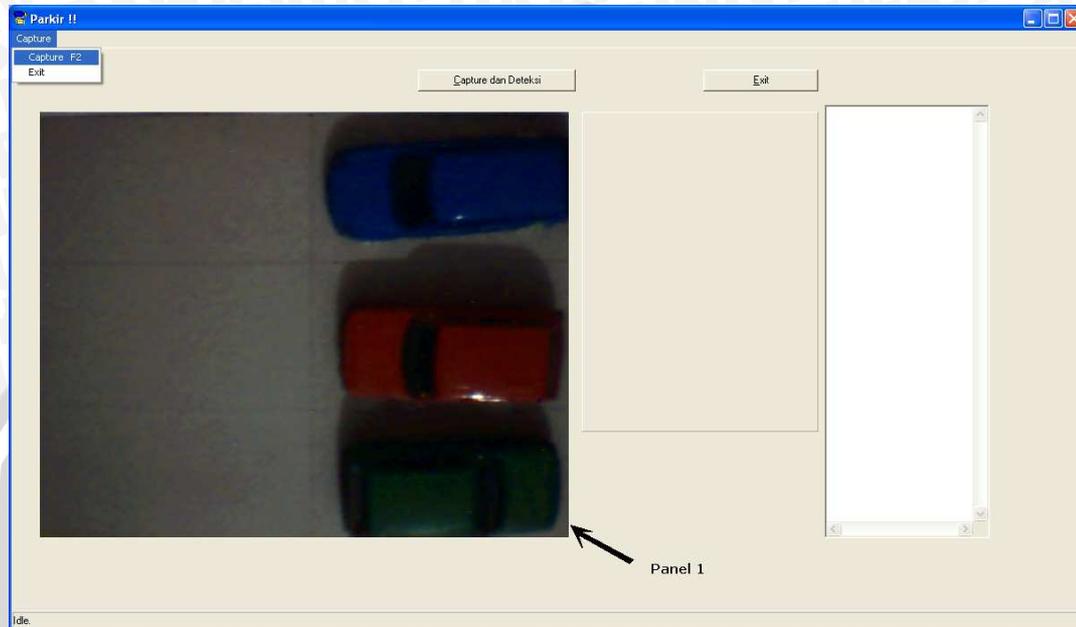


Gambar 5.2 Tampilan awal aplikasi

Sumber : Implementasi dan Pengujian

5.3.2 Tampilan Panel 1

Pada tampilan ini, dapat mengetahui obyek atau region yang dicapture. Pada saat program berjalan, panel ini menampilkan apa yang ditangkap oleh *web-camera*

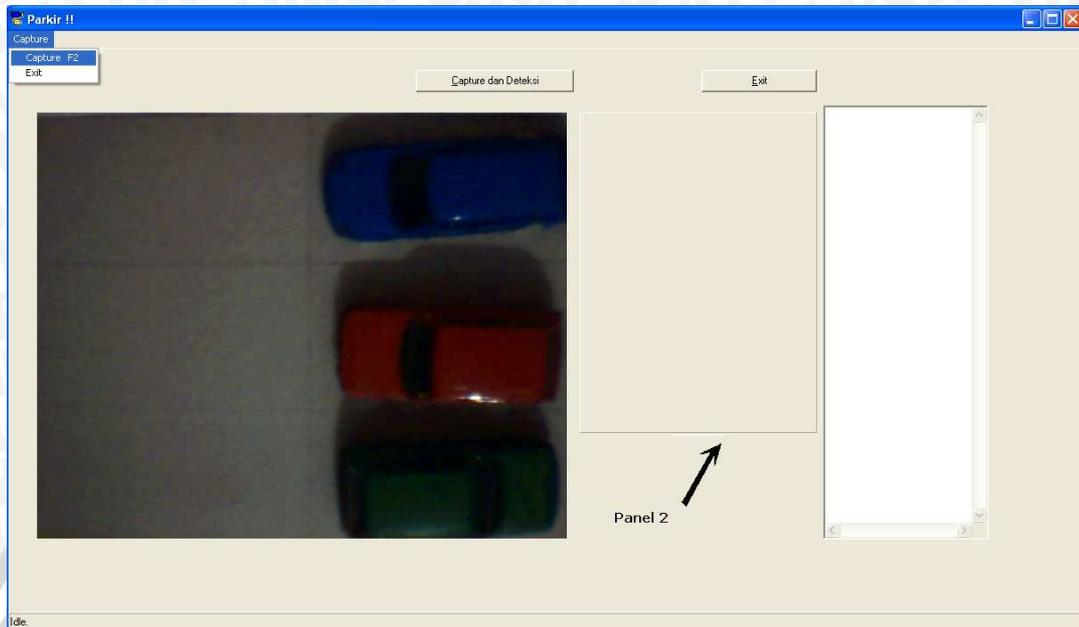


Gambar 5.3 Tampilan Panel 1

Sumber : Implementasi dan Pengujian

5.3.3 Tampilan Panel 2

Tampilan ini menampilkan hasil setelah program berjalan dan menangkap gambar obyek. Setelah program dijalankan, di dalam panel ini terdapat 2 kotak berwarna hijau dan merah, dimana kotak hijau mewakili keadaan kosong parkir dan kotak merah mewakili keadaan parkir yang terisi

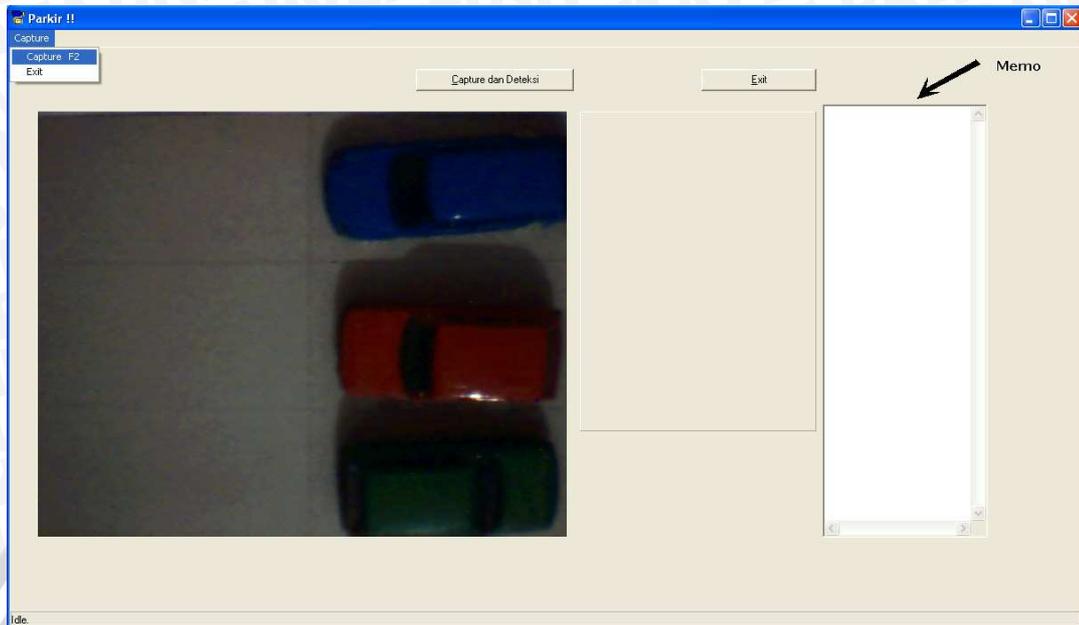


Gambar 5.4 Tampilan Panel 2

Sumber : Implementasi dan Pengujian

5.3.4 Tampilan Memo

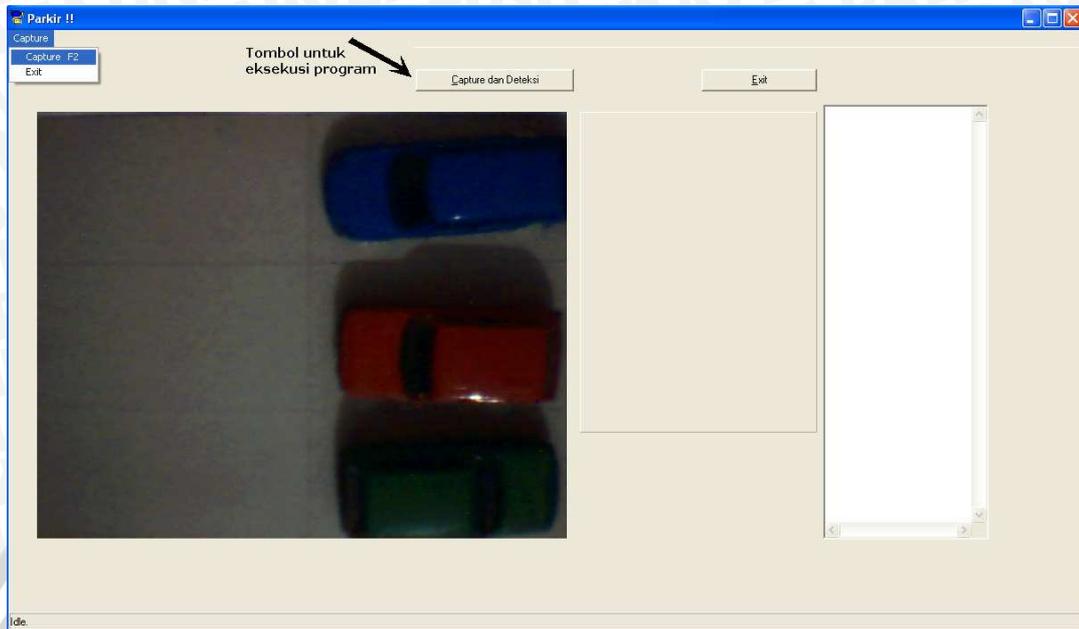
Pada fitur ini, ditampilkan hasil untuk perhitungan rata-rata Red, Green, Blue, serta ditampilkan pula hasil perbandingan RGB untuk mengetahui hasil perhitungan final nya setelah program dijalankan.



Gambar 5.5 Tampilan Memo
Sumber : Implementasi dan Pengujian

5.3.5 Tampilan Tombol Capture dan Deteksi

Tombol ini digunakan untuk meng-*capture* gambar sekaligus untuk mengeksekusi program secara keseluruhan.



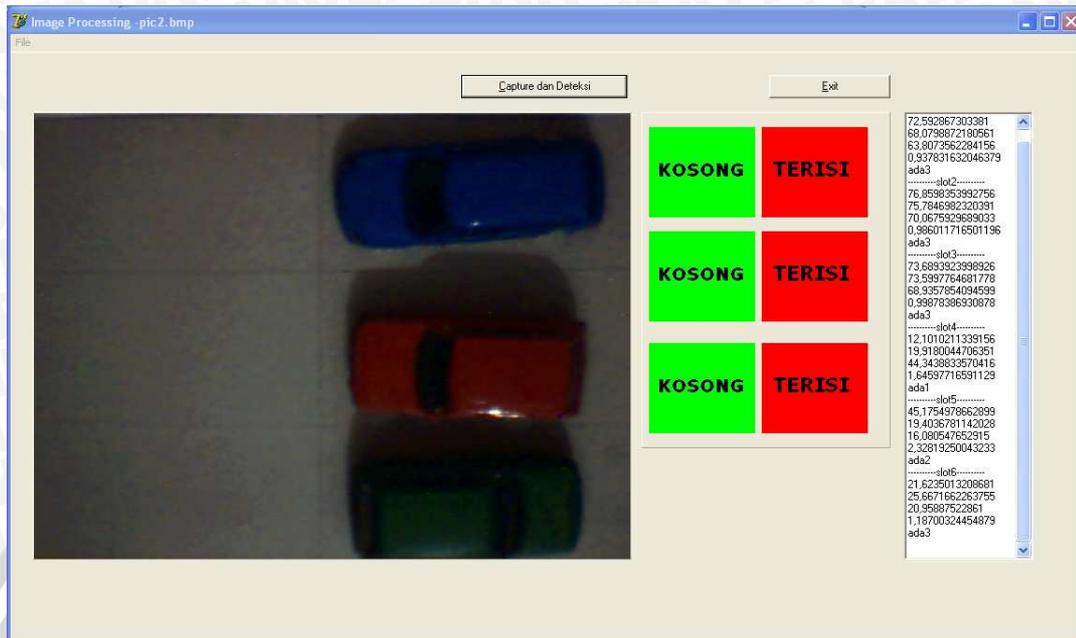
Gambar 5.6 Tampilan Tombol Capture dan Deteksi

Sumber : Implementasi dan Pengujian

5.3.6 Tampilan Saat Program berjalan

Setelah program berjalan, dan semua hasil perhitungan dimasukkan, maka didapatkan tampilan sebagai berikut.





Gambar 5.7 Tampilan Akhir Aplikasi

Sumber : Implementasi dan Pengujian

5.4 Pengujian

Sub-bab ini membahas proses pengujian terhadap aplikasi penuntun parkir dengan teknik pengolahan citra digital yang telah dibangun. Pengujian yang dilakukan meliputi pengujian *White-Box Testing* dan *Black-Box Testing* serta pengujian intensitas cahaya. Proses ini perlu dilakukan untuk mengetahui kinerja sistem apakah telah memenuhi kebutuhan yang ada.

5.4.1 *White-Box Testing*

Pengujian *white-box* berfokus pada struktur kontrol program. Test case dilakukan untuk memastikan bahwa semua statemen pada program telah dieksekusi paling tidak satu kali selama pengujian dan bahwa semua kondisi logis telah diuji. Pengujian basic path, tehnik pengujian white-box, menggunakan grafik (matriks grafiks) untuk melakukan serangkaian pengujian yang *independent* secara linear yang memastikan cakupan. Kesalahan logika dan asumsi yang tidak benar kebanyakan dilakukan ketika coding untuk “kasus tertentu”. Dibutuhkan kepastian bahwa

eksekusi jalur ini telah dites. Asumsi bahwa adanya kemungkinan terhadap eksekusi jalur yang tidak benar. Dengan *white box* testing dapat ditemukan kesalahan ini. Kesalahan penulisan yang acak. Seperti berada pada jalur logika yang membingungkan pada jalur normal.

5.4.2 *Black-Box Testing*

Pada pengujian *Black-Box Testing*, tidak perlu tahu apa yang sesungguhnya terjadi pada sistem atau perangkat lunak. Yang diuji adalah masukan dan keluarannya. Artinya, dengan berbagai masukan yang diberikan, apakah sistem atau perangkat lunak memberikan keluaran seperti yang diharapkan? Dalam pengujian ini, menggunakan skenario yang dikembangkan saat analisis sebagai panduan. Apakah keluaran sesuai dengan harapan serta kebutuhan pengguna.

Sistem atau perangkat lunak penuntun parkir dengan teknik pengolahan citra digital ini dibuat dengan gambar sebagai input dari sistem tersebut. Gambar yang digunakan sebagai input program adalah gambar yang dimanipulasi sedemikian rupa sehingga dapat menampilkan slot kosong pada tempat parkir. Pengambilan gambar dilakukan secara manual oleh user dengan bantuan kamera dengan posisi kamera sejajar dengan bidang parkir.

5.4.3 Pengujian pada Pengecekan Piksel

Pengujian ini dilakukan untuk mendapatkan resolusi yang tepat pada penggunaan kamera. Aplikasi tidak berjalan apabila kamera yang digunakan, mempunyai resolusi yang terlalu besar atau terlalu kecil. Pengujian pada proses ini menggunakan pengujian *Black-box*.

5.4.4 Pengujian intensitas warna pada obyek

Pengujian ini dilakukan dengan cara menjalankan aplikasi dan mengambil *sample* dari obyek untuk di-cek intensitas warnanya. Selain itu dilakukan juga perbandingan secara manual pada intensitas rata-rata RGB pada obyek yang di-

capture . Pengujian yang dilakukan pada proses manipulasi intensitas warna dilakukan untuk memeriksa apakah semua jalur independen yang ada pada algoritma proses manipulasi intensitas warna terlewati semua.

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
var gambar: TBitmap;
```

```
    O,P,Q,Z : TextFile ;
```

```
    FileGw1,FileGw2,FileGw3,isi : string ;
```

```
    i,j,R,G,B : integer;
```

```
    RGB : Tcolor;
```

```
    TotalR,totalG,totalB,totalRG : real ;
```

```
begin
```

```
R :=getRvalue(RGB);
```

```
G :=getGvalue(RGB);
```

```
B :=getBvalue(RGB);
```

```
totalR := totalR + (R/39368);
```

```
totalG := totalG + (G/39368);
```

```
totalB := totalB + (B/39368);
```

```
if totalR < totalG then
```

```
begin
```

```
totalRG := totalG/totalR;
```

```
isi:='adal';
```

```
end
```

```
else
```

```
begin
```

```
totalRG :=totalR/totalG;
```

```
isi:='ada2';
```

```
end;
```

```
if ((totalRG > 0.7) and (totalRG < 1.3)) then
```

```
begin
```

```
totalRG :=totalG/totalR;
```

```
isi:='ada3';
```

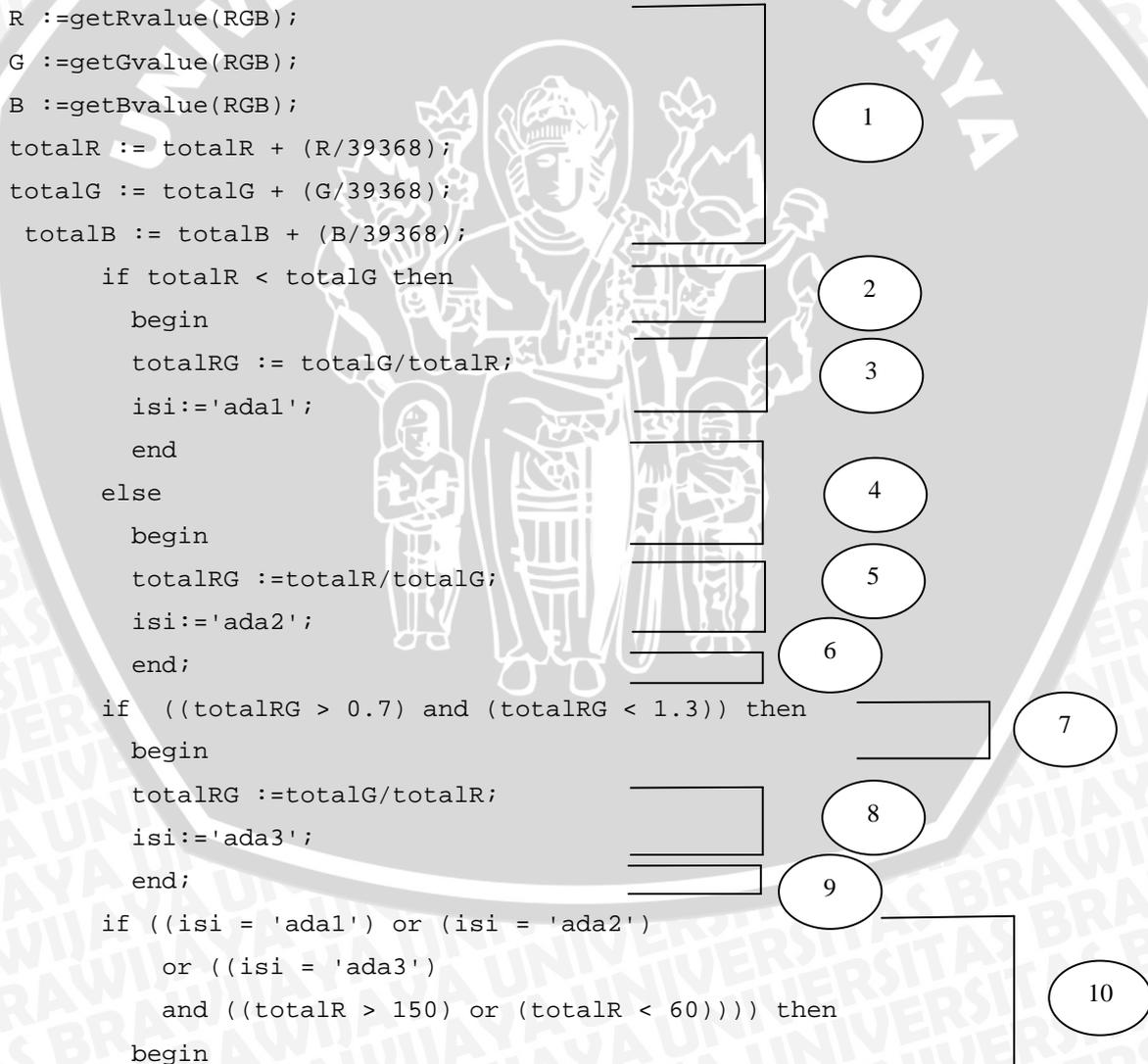
```
end;
```

```
if ((isi = 'adal') or (isi = 'ada2')
```

```
or ((isi = 'ada3')
```

```
and ((totalR > 150) or (totalR < 60))) then
```

```
begin
```



```
image2.Picture.LoadFromFile('terisi.bmp'); _____ 11  
end _____  
else _____ 12  
begin _____  
    image2.Picture.LoadFromFile('kosong.bmp'); _____ 13  
end; _____ 14
```





Gambar 5.8 Pemodelan algoritma prosedur manipulasi intensitas RGB ke dalam flow graph

Pemodelan ke dalam flow graph yang telah dilakukan terhadap fungsi labelling menghasilkan jumlah kompleksitas siklomatis (cyclomatic complexity) atau $V(G)$ melalui beberapa persamaan diantaranya :

$$V(G) = \text{Jumlah Region}$$

$$V(G) = E - N + 2$$

$$V(G) = P + 1$$

- dimana :
- E merupakan sisi (garis penghubung antar *node* (*Edge*))
 - N merupakan jumlah simpul (*Node*)
 - P merupakan simpul yang memiliki cabang (*Predicate Node*)

Berdasarkan persamaan di atas diperoleh perhitungan :

$$V(G) = \text{Jumlah Region} = (R1, R2, R3) = 3$$

$$V(G) = E - N + 2 = 15 - 14 + 2 = 3$$

$$V(G) = P + 1 = 2 + 1 = 3$$

Dari nilai *cyclomatic complexity* yang telah dihasilkan dari perhitungan yaitu tiga, ditentukan tiga buah basis set dari jalur independen yaitu:

Jalur 1 : 1-2-3-6-7-8-9-10-11-14

Jalur 2 : 1-2-4-5-6-7-8-9-10-11-14

Jalur 3 : 1-2-3-6-7-8-9-10-12-13-14

Jalur 4 : 1-2-4-5-6-7-8-9-10-12-13-14

Penentuan kasus uji (*test case*) untuk masing-masing jalur dan hasil eksekusi untuk masing-masing kasus uji adalah dijelaskan pada Tabel 5.1.

Tabel 5.1 Test case untuk pengujian fungsi manipulasi intensitas RGB

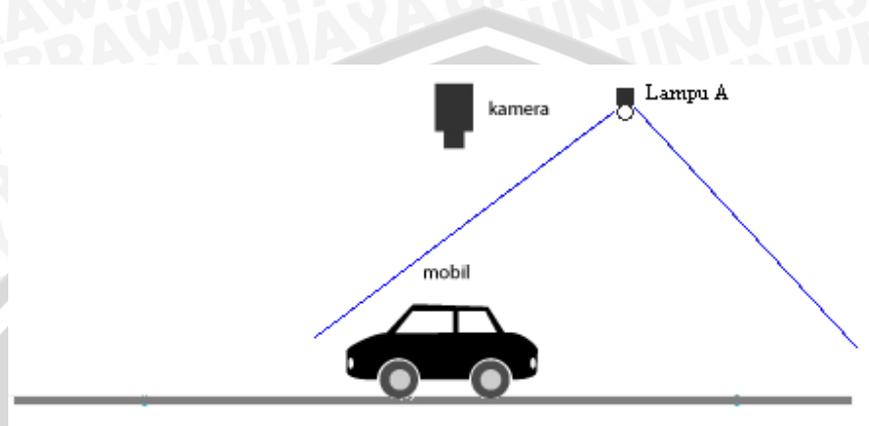
Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Gambar mobil dengan warna dominan merah	Indikasi ada1 dan Terisi	Indikasi ada1 dan Terisi
2	Gambar mobil dengan warna dominan biru	Indikasi ada2 dan Terisi	Indikasi ada2 dan Terisi
3	Gambar mobil dengan warna dominan hijau	Indikasi ada3 dan terisi	Indikasi ada3 dan terisi
4	Gambar diindikasikan tidak ada kendaraan	Indikasi ada3 dan kosong	Indikasi ada3 dan kosong

5.4.5 Pengujian Intensitas Cahaya

Pengujian ini adalah untuk menentukan hasil keluaran dari program secara keseluruhan dengan menambahkan parameter intensitas cahaya. Pengujian intensitas cahaya menggunakan 3 macam intensitas yaitu : rendah, sedang dan tinggi.

5.4.5.1 Pengujian intensitas cahaya sedang

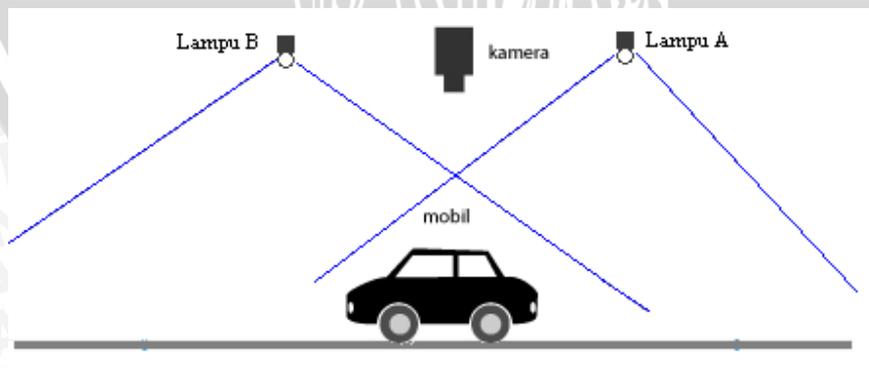
Langkah-langkah yang dilakukan pada pengujian ini adalah, meletakkan obyek mobil dalam media abu-abu, pengecekan akhir posisi kamera yang diletakkan tepat diatas media percobaan, mempersiapkan lampu A dengan daya sekitar 10 watt lalu di letakkan di sebelah kamera dengan jarak tertentu, proses capture dan deteksi dimulai. Cara pengambilan gambar obyek dan peletakan lampu dapat dilihat pada gambar 5.9.



Gambar 5.9 Proses pengambilan gambar pada intensitas sedang

5.4.5.2 Pengujian intensitas cahaya tinggi

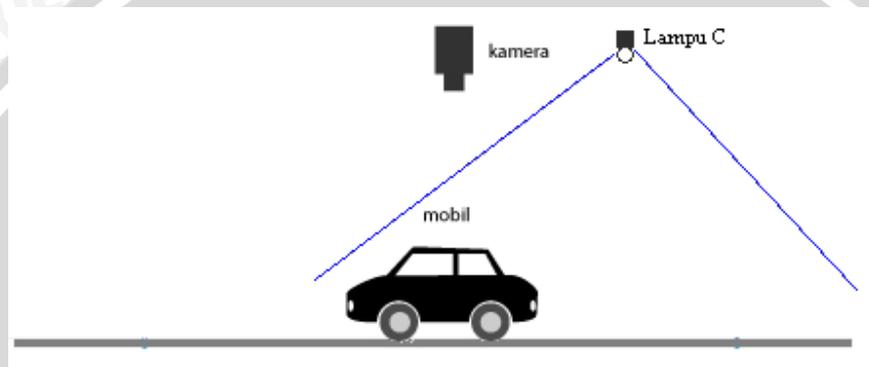
Langkah-langkah yang dilakukan pada pengujian ini adalah, meletakkan obyek mobil dalam media abu-abu, pengecekan akhir posisi kamera yang diletakkan tepat diatas media percobaan, mempersiapkan lampu A dan lampu B dengan daya masing-masing sekitar 10 watt lalu di letakkan di sebelah kamera dengan jarak tertentu, proses capture dan deteksi dimulai. Cara pengambilan gambar obyek dan peletakan lampu dapat dilihat pada gambar 5.10.



Gambar 5.10 Proses pengambilan gambar pada intensitas tinggi

5.4.5.3 Pengujian Intensitas cahaya rendah

Langkah-langkah yang dilakukan pada pengujian ini adalah, meletakkan obyek mobil dalam media abu-abu, pengecekan akhir posisi kamera yang diletakkan tepat diatas media percobaan, mempersiapkan lampu C dengan daya sekitar 5 watt lalu di letakkan di sebelah kamera dengan jarak tertentu, proses capture dan deteksi dimulai. Cara pengambilan gambar obyek dan peletakan lampu dapat dilihat pada gambar 5.11



Gambar 5.11 Proses pengambilan gambar pada intensitas rendah

5.4.6 Analisa Hasil Pengujian

Setelah dilakukan serangkaian pengujian pada berbagai *sample* obyek, maka dapat ditentukan beberapa kondisi yang memungkinkan agar aplikasi ini berjalan. Pada pengujian awal, resolusi kamera yang diperbolehkan pada aplikasi ini adalah 640 x 480. Intensitas cahaya tidak boleh terlalu terang ataupun terlalu gelap. Pada intensitas cahaya yang terlalu terang, nilai rata-rata intensitas RGB secara keseluruhan dapat melebihi 200, sehingga melampaui batas yang diperbolehkan pada aplikasi. Pada Intensitas cahaya yang terlalu gelap, aplikasi cenderung mengenali obyek sebagai warna hitam, sehingga terdapat kecenderungan tempat parkir terlihat terisi. Kesesuaian antara jumlah dan posisi mobil terhadap intensitas cahaya dapat dilihat pada tabel 5.2 dan tabel 5.3 , sedangkan kesesuaian antara kondisi asli dan kondisi yang ditampilkan oleh program dapat dilihat pada tabel 5.4 .

Tabel 5.2 Pengujian pada kesesuaian posisi mobil dan jumlah mobil

No	Intensitas cahaya	Jumlah mobil	Jumlah mobil output program	Kesesuaian posisi dan jumlah mobil
1	Sedang	3	3	Sesuai
2	Sedang	3	3	Sesuai
3	Sedang	2	2	Sesuai
4	Sedang	2	2	Sesuai
5	Sedang	2	2	Sesuai
6	Sedang	2	2	Sesuai
7	Sedang	2	4	Tidak Sesuai
8	Sedang	2	2	Sesuai
9	Sedang	1	1	Sesuai
10	Sedang	4	4	Sesuai
11	Sedang	4	5	Tidak Sesuai
12	Sedang	4	5	Tidak Sesuai
13	Sedang	4	4	Sesuai
14	Sedang	4	4	Sesuai
15	Sedang	3	4	Tidak Sesuai
16	Sedang	3	3	Sesuai

Prosentase kesalahan = 25%

Tabel 5.3 Pengujian pada kesesuaian posisi mobil dan jumlah mobil dengan intensitas cahaya berbeda

No	Intensitas cahaya	Jumlah mobil	Jumlah mobil output program	Kesesuaian posisi dan jumlah mobil
1	Rendah	3	6	Tidak Sesuai (jumlah)
2	Sedang	3	3	Sesuai
3	Tinggi	3	3	Tidak Sesuai (posisi)
4	Rendah	1	6	Tidak Sesuai (jumlah)
5	Sedang	1	1	Sesuai
6	Tinggi	1	1	Sesuai
7	Rendah	2	6	Tidak Sesuai (jumlah)
8	Sedang	2	2	Sesuai
9	Tinggi	2	2	Tidak Sesuai (posisi)
10	Rendah	1	6	Tidak Sesuai (jumlah)
11	Sedang	1	6	Sesuai
12	Tinggi	1	6	Sesuai

Prosentase kesalahan = 50%



Tabel 5.4 Rerata nilai RGB dengan nilai ambang dan hasil akhir

Percobaan	Slot	TotalR	TotalG	TotalB	Nilai Ambang	Hasil
1	1	72.5928	68.0798	63.8073	0.9378	Kosong
	2	76.8598	75.7846	70.0675	0.9860	Kosong
	3	73.6893	73.5997	68.9357	0.9987	Kosong
	4	12.1010	19.9180	44.3438	1.6459	Terisi
	5	45.1754	19.4036	16.0805	2.3281	Terisi
	6	21.6235	25.6671	20.9588	1.1870	Terisi
2	1	8.3121	20.4933	38.5385	2.4654	Terisi
	2	11.1776	26.7097	18.7095	2.3895	Terisi
	3	23.2813	19.9340	14.9466	0.8562	Terisi
	4	62.2188	77.5564	62.0987	1.2465	Kosong
	5	65.2986	83.6304	66.6811	1.2807	Kosong
	6	24.5550	21.3727	16.3965	0.8703	Terisi
3	1	8.1409	20.3902	37.9210	2.5046	Terisi
	2	10.9511	26.7476	18.3632	2.4424	Terisi
	3	23.2855	20.2461	15.2153	0.8694	Terisi
	4	60.5684	75.5377	60.5312	1.2471	Kosong
	5	26.5522	20.0022	14.1540	1.3274	Terisi
	6	62.0221	81.1231	65.8747	1.3079	Terisi

Tabel 5.5 Perbandingan antara kondisi real dengan keluaran aplikasi yang disertai error

Percobaan	Slot	Kondisi Slot Sebenarnya	Kondisi Slot di Software	Error (%)
1	1	Kosong	Kosong	0
	2	Kosong	Kosong	
	3	Kosong	Kosong	
	4	Terisi	Terisi	
	5	Terisi	Terisi	
	6	Terisi	Terisi	
2	1	Terisi	Terisi	0
	2	Terisi	Terisi	
	3	Terisi	Terisi	
	4	Kosong	Kosong	
	5	Kosong	Kosong	
	6	Terisi	Terisi	
3	1	Terisi	Terisi	16,66
	2	Terisi	Terisi	
	3	Terisi	Terisi	
	4	Kosong	Kosong	
	5	Terisi	Terisi	
	6	Kosong	Terisi	

B A B VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Teknik capture yang digunakan, yang jauh lebih sederhana dan lebih menghemat biaya dalam pembuatan sistem. Dengan resolusi kamera 640 x 480, aplikasi ini dapat membedakan antara slot parkir yang kosong atau terisi. Program pendeteksi parkir dibuat dengan cara membandingkan intensitas rata-rata antara nilai *Red*, *Green*, dan *Blue* pada suatu obyek tempat parkir yang disorot oleh kamera.

6.2 Saran

Aplikasi ini mempunyai keunggulan dibanding dengan metode-metode sebelumnya, namun diantara keunggulan tersebut, pasti ada kelemahan yang dimiliki oleh sistem. Intensitas cahaya yang dinamis membuat aplikasi ini tidak berjalan, karena nilai dari perbandingan antara *Red*, *Green*, *Blue* juga berganti ganti, sehingga dibutuhkan satu tetapan intensitas yang *fix* agar aplikasi ini berjalan.

Hal lain yang patut pula menjadi perhatian adalah, ketinggian kamera. Dengan sudut pandang dan resolusi kamera yang terbatas, maka tidak dimungkinkan pemasangan kamera pada ketinggian maksimal yang disyaratkan pada suatu bangunan.

Untuk mengatasi beberapa masalah ini, disarankan agar dilakukan penyempurnaan pada program sehingga bisa dipakai pada intensitas cahaya yang dinamis. Selain itu, diperlukan juga penggunaan kamera dengan resolusi dan sudut pandang yang lebih besar, sehingga dapat menurunkan ketinggian kamera untuk suatu bangunan.

DAFTAR PUSTAKA

Fadlisya, Taufiq, Zulfikar & Fauzan . Pengolahan Citra Menggunakan Delphi, Graha Ilmu.Yogyakarta, 2008.

Achmad Basuki, Jozua F. Palandi, Fatchurrochman, “Pengolahan Citra Digital Menggunakan Visual Basic”, Graha Ilmu,Yogyakarta, 2005

Pranata Antony, Borland Delphi 6 edisi 4, Andi, Yogyakarta, 2003

Usman Ahmad, “Pengolahan Citra Digital dan Teknik Pemrogramannya”, Graha ilmu, yogyakarta, 2005

Achmad Basuki, “Image Enhacement”, Politeknik Elektronika Negeri Surabaya, Surabaya, 2005.

Dadet Pramadihanto, “Machine Learning”, Politeknik Elektronika Negeri Surabaya, Surabaya, 2005.

Anonymous. Image Capturing. <http://www.softpedia.com/get/Internet/Servers/Other-Servers/Capture-Em.shtml>

Diakses tanggal 24 April 2010.

LAMPIRAN

Lampiran 1 Gambar Simulasi dan Cara Pengambilan Data



Gambar Cara Pengambilan Rekaman Tampak Depan



Gambar Cara Pengambilan Rekaman Tampak Atas

Lampiran 2 Source Code Program

```

unit FilmStrip;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, StdCtrls, Menus, IdBaseComponent, IdComponent,
  ShellAPI, VFW, INIFiles, IdIPWatch, Clipbrd;
const
  WM_ICONTRAY = WM_USER + 1;
  ENTER = Chr(13) + Chr(10);
type
  TfrmMain = class(TForm)
    pnlVideo: TPanel;
    MainMenu1: TMainMenu;
    mnuPopUp: TPopupMenu;
    mnuShow: TMenuItem;
    N1: TMenuItem;
    tmrUpdateTray: TTimer;
    N3: TMenuItem;
    stbStatus: TPanel;
    dlgSave: TSaveDialog;
    mnuCapture: TMenuItem;
    mnuSave: TMenuItem;
    mnuCapture2: TMenuItem;
    tmrResetStatus: TTimer;
    tmrCapture: TTimer;
    Panel1: TPanel;
    Memo5: TMemo;
    Button1: TButton;
    Exit1: TMenuItem;
    Image2: TImage;
  end;

```

```
Image3: TImage;
Image4: TImage;
Image5: TImage;
Image6: TImage;
Image7: TImage;
Image1: TImage;
Edit1: TEdit;
Label1: TLabel;
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure FormCreate(Sender: TObject);
procedure WMSizing(var Message: TMessage); message WM_SIZING;
procedure Status(const NewStatus: string);
procedure mnuShowClick(Sender: TObject);
procedure mnuQuitClick(Sender: TObject);
procedure PrepareTray;
procedure AppMinimize(Sender: TObject);
procedure tmrUpdateTrayTimer(Sender: TObject);
procedure mnuSaveClick(Sender: TObject);
procedure mnuCapture2Click(Sender: TObject);
procedure tmrResetStatusTimer(Sender: TObject);
procedure tmrCaptureTimer(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Exit1Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
private
  { Private declarations }
  a:NOTIFYICONDATA;
public
  { Public declarations }
procedure TrayMessage(var Msg: TMessage); message WM_ICONTRAY;
end;
```

```
var
    frmMain: TfrmMain;
    Video : TVideo;
    tmpFile: String;
    Port : Integer;
    AppPath : String;
    INIFile : TINIFile;
    INIFileName : String;
    SaveFileName : String;
    picCount : Integer;
    CaptureTime : Integer;
    Interval : Integer;
    function ReadFromINI(const Section : String; const Key : String; const Default : String)
: String; overload;
    function ReadFromINI(const Section : String; const Key : String; const Default : LongInt)
: LongInt; overload;
    function WriteToINI(const Section : String; const Key : String; const Value : String)
: Boolean; overload;
    function WriteToINI(const Section : String; const Key : String; const Value : LongInt)
: Boolean; overload;
implementation
//uses About, Log;
{$R *.dfm}
{$R click.res}
function ReadFromINI(const Section : String; const Key : String; const Default : String)
: String;
var
    ReadINI: TIniFile;
begin
    try
        ReadFromINI := ReadINI.ReadString(Section, Key, Default);
    finally
        ReadINI.Free;
```

```
end;
end;
function ReadFromINI(const Section : String; const Key : String; const Default : LongInt)
:LongInt;
var
ReadINI: TIniFile;
begin
try
    ReadINI := TIniFile.Create(INIFilename);
finally
    ReadINI.Free;
end;
end;
function WriteToINI(const Section : String; const Key : String; const Value : String)
:Boolean;
var
WriteINI: TIniFile;
begin
try
    WriteINI.WriteString(Section, Key, Value);
    WriteToINI := True;
finally
    WriteINI.Free;
end;
end;
function WriteToINI(const Section : String; const Key : String; const Value : LongInt)
:Boolean;
var
WriteINI: TIniFile;
begin
try
    WriteINI.WriteInteger(Section, Key, Value);
    WriteToINI := True;
```

```
finally
    WriteINI.Free;
end;
end;
procedure TfrmMain.PrepareTray;
begin
    Application.OnMinimize := AppMinimize;
    a.cbSize := sizeof(a);
    a.Wnd := Handle;
    a.hIcon := Application.Icon.Handle;
    a.uFlags := NIF_ICON + NIF_MESSAGE + NIF_TIP;
end;
procedure TfrmMain.TrayMessage(var Msg: TMessage);
var
    curPos : TPoint;
begin
    case Msg.lParam of
        WM_RBUTTONDOWN:
            begin
                if (Self.Visible) then mnuShow.Caption := '&Hide' else mnuShow.Caption := '&Show';
            end;
        WM_LBUTTONDOWN:
            begin
                Self.Show;
                Application.BringToFront;
            end;
    end;
end;
end;
procedure TfrmMain.AppMinimize(Sender: TObject);
begin
    Self.Hide;
end;
procedure TfrmMain.tmrUpdateTrayTimer(Sender: TObject);
```

```

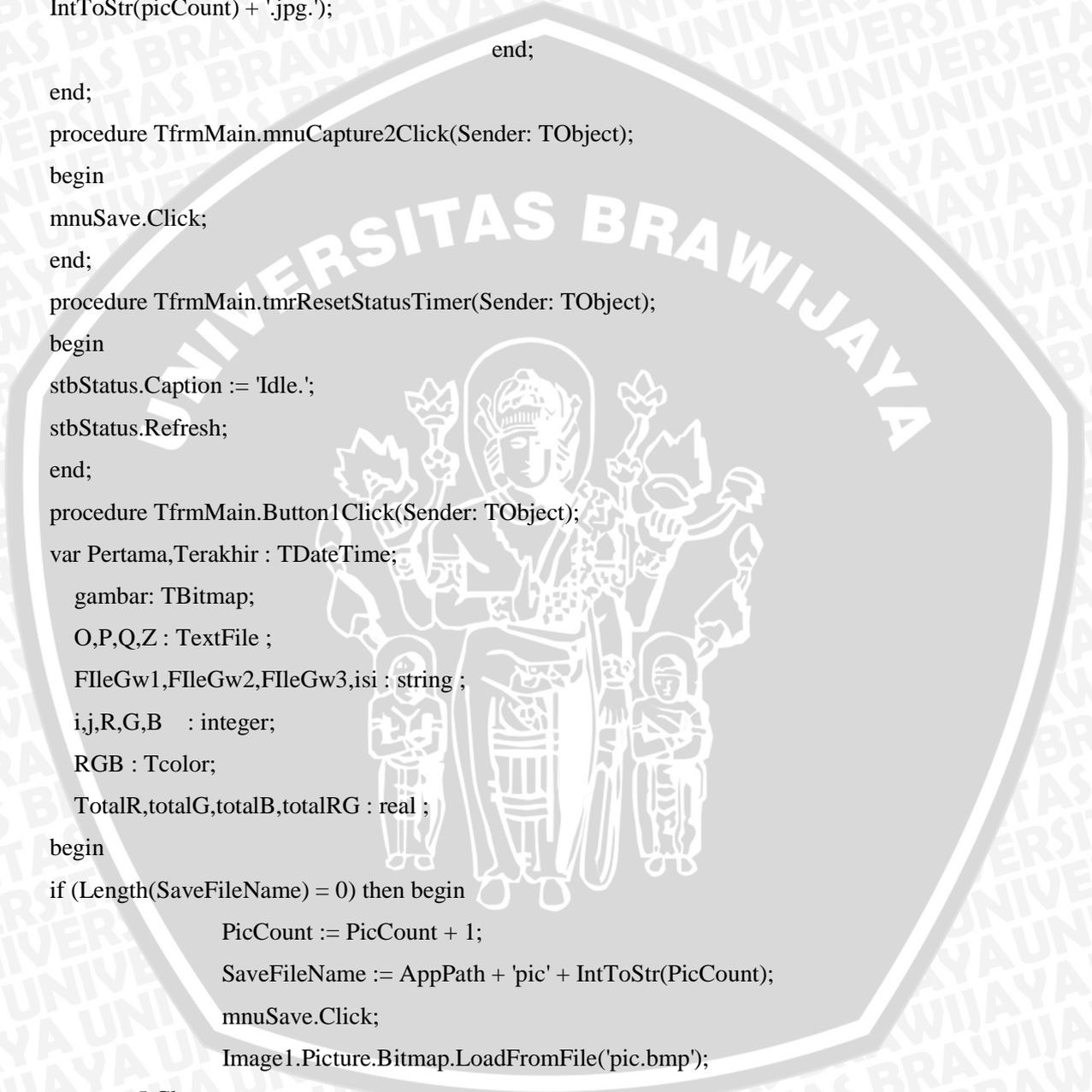
begin
Shell_NotifyIcon(NIM_ADD, @a);
end;
procedure TfrmMain.FormCreate(Sender: TObject);
begin
if (Pos('/T', UpperCase(CMDLine)) > 0) or (Pos('/M', UpperCase(CMDLine)) > 0) then begin
    Self.WindowState := wsMinimized;
    Self.Hide;
end;
if (Pos('/?', UpperCase(CMDLine)) > 0) or (Pos('/H', UpperCase(CMDLine)) > 0) or (Pos('-
H', UpperCase(CMDLine)) > 0) or (Pos('-?', UpperCase(CMDLine)) > 0) then begin
    Application.Terminate;
    exit;
end;
AppPath := ExtractFilePath(Application.ExeName);
INIFileName := AppPath + 'WebCam.INI';
PrepareTray;
Video := TVideo.Create(pnlVideo);
end;
procedure TfrmMain.FormClose(Sender: TObject; var Action: TCloseAction);
begin
FreeAndNil(Video);
end;
procedure TfrmMain.WMSizing(var Message: TMessage);
var
    lprc: PRect;
    TitleBarHeight: Integer;
begin
    TitleBarHeight := Height - ClientHeight;
    lprc^.Bottom := lprc^.Top + Round(ClientWidth * 0.75) + TitlebarHeight +
    stbStatus.Height;
end;
procedure TfrmMain.Status(const NewStatus: string);

```

```
var
  strTime : String;
  TheTime : TDateTime;
begin
  TheTime := Time;
  stbStatus.Caption := 'Request sedang diproses';
  stbStatus.Refresh;
  tmrResetStatus.Enabled := True;
end;
procedure TfrmMain.mnuShowClick(Sender: TObject);
begin
  if (Self.Visible) then Self.Hide else Self.Show;
end;
procedure TfrmMain.mnuQuitClick(Sender: TObject);
begin
  FreeAndNil(Video);
  Application.Terminate;
end;
procedure TfrmMain.mnuSaveClick(Sender: TObject);
begin
  if (Length(SaveFileName) = 0) then begin
    PicCount := PicCount + 1;
    SaveFileName := AppPath + 'pic' + IntToStr(PicCount);
  end;
  if DirectoryExists(SaveFileName) then SaveFileName := SaveFileName + 'pic';
  if (FileExists(SaveFileName + IntToStr(picCount) + '.jpg')) then begin
    picCount := picCount + 1;
    exit;
  end;
  if not Video.TakePicture(SaveFileName + IntToStr(picCount) + '.jpg') then begin
    Status('Error!');
```

```

end
else begin
    Status('Captured ' + SaveFileName +
IntToStr(picCount) + '.jpg.');
```



```

    end;
end;
procedure TfrmMain.mnuCapture2Click(Sender: TObject);
begin
    mnuSave.Click;
end;
procedure TfrmMain.tmrResetStatusTimer(Sender: TObject);
begin
    stbStatus.Caption := 'Idle.';
    stbStatus.Refresh;
end;
procedure TfrmMain.Button1Click(Sender: TObject);
var Pertama, Terakhir : TDateTime;
    gambar: TBitmap;
    O,P,Q,Z : TextFile ;
    FileGw1,FileGw2,FileGw3,isi : string ;
    i,j,R,G,B : integer;
    RGB : Tcolor;
    TotalR,totalG,totalB,totalRG : real ;
begin
    if (Length(SaveFileName) = 0) then begin
        PicCount := PicCount + 1;
        SaveFileName := AppPath + 'pic' + IntToStr(PicCount);
        mnuSave.Click;
        Image1.Picture.Bitmap.LoadFromFile('pic.bmp');
        memo5.Clear;
    end;
    if DirectoryExists(SaveFileName) then SaveFileName := SaveFileName + 'pic';
```

```
if (FileExists(SaveFileName + IntToStr(picCount) + '.jpg')) then begin
```

```
    picCount := picCount + 1;
```

```
    mnuSave.Click;
```

```
    Image1.Picture.Bitmap.LoadFromFile('pic.bmp');
```

```
    memo5.Clear;
```

```
    Pertama:=now;
```

```
    for i := 17 to 312 do
```

```
        begin
```

```
            RGB := image1.Canvas.Pixels [i,j];
```

```
            R :=getRvalue(RGB);
```

```
            G :=getGvalue(RGB);
```

```
            B :=getBvalue(RGB);
```

```
            totalR := totalR + (R/20894);
```

```
            totalG := totalG + (G/20894);
```

```
            totalB := totalB + (B/20894);
```

```
        end;
```

```
        if totalR < totalG then
```

```
            begin totalRG := totalG/totalR;
```

```
                isi:='ada1';
```

```
            end
```

```
        else
```

```
            begin
```

```
                totalRG :=totalR/totalG;
```

```
                isi:='ada2';
```

```
            end;
```

```
        if totalRG > 0.7 then
```

```
            begin totalRG :=totalG/totalR;
```

```
                isi:='ada3';
```

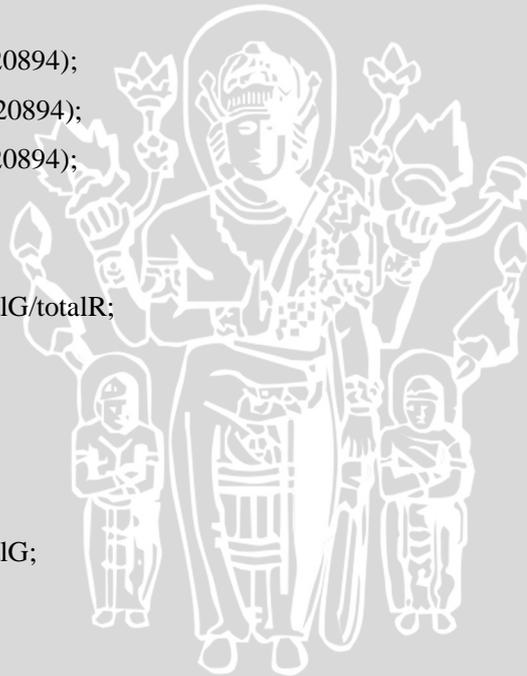
```
            end;
```

```
        if ((totalR > 150) or (totalR < 60))then
```

```
            begin
```

```
                image2.Picture.LoadFromFile('terisi.bmp');
```

```
            end
```



```
else
begin
    image2.Picture.LoadFromFile('kosong.bmp');
end;
memo5.Lines.Add('-----');
memo5.Lines.Add(floattostr(totalR));
memo5.Lines.Add(floattostr(totalG));
memo5.Lines.Add(floattostr(totalB));
memo5.Lines.Add(floattostr(totalRG));
memo5.Lines.Add(isi);
Terakhir := now;
Edit1.Text:= FormatDateTime("Respon: " ss.zzz', Terakhir-Pertama);
    exit;
end;

end;
procedure TfrmMain.Exit1Click(Sender: TObject);
begin
Close;
end;
procedure TfrmMain.Button3Click(Sender: TObject);
begin
memo5.Clear;
end;
end.
```

