

**RANCANG BANGUN COLOR RECOGNATION DENGAN
MENGUNAKAN METODE DYNAMIC COLOR THRESHOLDING
DALAM SISTEM INFORMASI GUDANG YANG TERINTEGRASI
DENGAN ORACLE**

SKRIPSI



Disusun Oleh:
Donny Eko Suprayogo
09.18.103

**PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNIK INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2013**

RECEIVED HONOLULU POLICE DEPARTMENT
DEPARTMENT POLICE CHIEF'S OFFICE
DEPARTMENT OF PUBLIC SAFETY
HONOLULU, HAWAII

RECEIVED

RECEIVED
DEPARTMENT OF PUBLIC SAFETY
HONOLULU, HAWAII

RECEIVED HONOLULU POLICE DEPARTMENT
DEPARTMENT POLICE CHIEF'S OFFICE
DEPARTMENT OF PUBLIC SAFETY
HONOLULU, HAWAII

LEMBAR PERSETUJUAN

RANCANG BANGUN COLOR RECOGNATION DENGAN MENGUNAKAN METODE DYNAMIC COLOR THRESHOLDING DALAM SISTEM INFORMASI GUDANG YANG TERINTEGRASI DENGAN ORACLE

SKRIPSI

*Disusun dan Diajukan untuk melengkapi dan memenuhi persyaratan guna
mencapai Gelar Sarjana Teknik Informatika Strata Satu (S-1)*

Disusun Oleh :

DONNY EKO SUPRAYOGO

09.18.103

Diperiksa dan Disetujui,

Dosen Pembimbing I

Dosen Pembimbing II

Ir. Sidik Noertjahjono, MT
NIP.Y. 1028700163

Ali Mahmudi, BEng, PhD
NIP.P. 1031000429

Ketua Jurusan Teknik Informatika S-1

Joseph Dedy Irawan, ST, MT
NIP. 197404162005031002

**PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG**

Skripsi. Ini saya persembahkan untuk Bapak dan Ibu
saya karena sudah rela setiap pagi bangun ketika
orang lain masih tidur untuk mencari Rezeki guna
Membayai Kuliah saya.

RANCANG BANGUN COLOR RECOGNATION DENGAN MENGUNAKAN METODE DYNAMIC COLOR THRESHOLDING DALAM SISTEM INFORMASI GUDANG YANG TERINTEGRASI DENGAN ORACLE

Donny Eko Suprayogo

Program Studi Teknik Informatika S-1
Fakultas Teknologi Industri
Institut Teknologi Nasional Malang
Jl. Raya Karanglo Km 2 Malang
Email : Donny.e.Suprayogo@gmail.com

**Dosen Pembimbing: 1. Ir. Sidik Noertjahjono, MT
2. Ali Mahmudi, BEng, PhD**

Abstrak

Kombinasi dari beberapa warna bisa diciptakan sebuah karakter baru untuk membantu kerahasiaan data dan efisiensi waktu dalam sebuah pekerjaan. Dengan menggunakan sebuah webcam kemudian akan diteruskan ke dalam program yang dibuat dan diolah sesuai dengan basis pengetahuan akan dideteksi karakter yang diwakili oleh kombinasi sebuah warna. Sistem ini menggunakan kombinasi dari 3 warna dasar yaitu Merah, Hijau, Biru yang akan mewakili sebuah karakter.

Proses Color Recognition dalam menentukan nilai warna apakah bernilai Merah, Hijau, biru akan digunakan metode Color Thresholding adalah proses menentukan sebuah warna dengan memberikan batas nilai intensitas. Dengan metode ini untuk menentukan warna merah jika nilai $R > 200$, $G < 64$, $B < 64$ untuk warna hijau $R < 64$, $G > 200$, $B < 64$ sedangkan warna biru $R < 64$, $G < 64$, $B > 200$. Jika diibaratkan nilai lebih dari 200 dianggap bernilai 1 sedangkan di bawah 64 dianggap 0 maka kombinasi RGB warna merah = 100, Hijau = 010, Biru = 001. Dari nilai RGB yang ada yang setiap nilai dasar 8 bit bisa diciptakan $2^4 = 16.777.216$ tetapi pada metode ini dipakai 3 posisi dengan 3 warna dasar sehingga hanya menghasilkan $3^3 = 27$ kombinasi dan dari 27 kombinasi yang dipakai adalah 10 kombinasi yang digunakan untuk memberikan informasi yang terhubung pada sistem informasi yang terintegrasi dengan database Oracle.

Setelah dilakukan proses pengujian, diperoleh kesimpulan yaitu tingkat keberhasilan program pada pengujian tahap pertama dengan 5 sampel yang dibaca secara berurutan hasilnya dari 45 karakter 100% berhasil terbaca sedangkan pada tahap kedua 5 sampel dilakukan pembacaan secara acak selama 20 kali dari 180 karakter terjadi kesalahan 1 karakter sehingga persentase keberhasilannya adalah 98,875%.

Kata kunci : Dynamic Color Thresholding, Recognition, Code, Oracle

LEMBAR PERNYATAAN

Yang bertanda tangan di bawah ini. Saya :

Nama : DONNY EKO SUPRAYOGO

Nim : 09.18.103

Program Studi : Teknik Informatika S-1, Institut Teknologi Nasional Malang

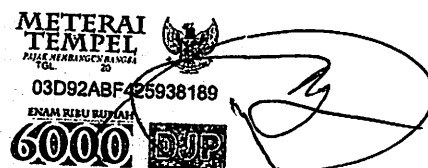
Menyatakan dengan sebenar-sebenarnya bahwa skripsi saya yang berjudul:

“RANCANG BANGUN COLOR RECOGNATION DENGAN MENGGUNAKAN METODE DYNAMIC COLOR THRESHOLDING DALAM SISTEM INFORMASI GUDANG YANG TERINTEGRASI DENGAN ORACLE”

Ini seluruhnya benar-benar merupakan hasil karya sendiri, Bukan merupakan pengambil alihan tulisan atau pemikiran orang lain kecuali sumber landasan teori sebagai penunjang penulisan yang telah dituliskan sebagaimana mestinya di akhir penulisan skripsi ini. Apabila dikemudian terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan skripsi ini merupakan hasil karya orang lain, saya bersedia menerima sanksi sesuai dengan hukum yang berlaku.

Malang, 20 Maret 2013

Yang Membuat Pernyataan



Donny Eko Suprayogo
0918103

KATA PENGANTAR

Puji dan Syukur penyusun panjatkan kehadiran Tuhan Yang Maha Esa, karena telah memberikan rahmat-Nya sehingga penyusun dapat menyelesaikan skripsi dengan judul RANCANG BANGUN COLOR RECOGNATION DENGAN MENGGUNAKAN METODE DYNAMIC COLOR THRESHOLDING DALAM SISTEM INFORMASI GUDANG YANG TERINTEGRASI DENGAN ORACLE sesuai dengan waktu yang ditentukan.

Penyusunan skripsi ini merupakan salah satu persyaratan untuk menyelesaikan program pendidikan Strata Satu (S-1) Teknik Informatika, Fakultas Teknologi Industri di Institut Teknologi Nasional Malang.

Pada kesempatan ini penyusun mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Ir. Soeparno, MT selaku Rektor ITN Malang.
2. Anang Subardi, MT selaku Dekan Fakultas Teknologi Industri ITN Malang.
3. Joseph Dedy Irawan, ST, MT selaku Ketua Jurusan Teknik Informatika ITN Malang.
4. Ir. Sidik Noertjahjono, MT selaku dosen pembimbing I.
5. Ali Mahmudi, BEng, PhD selaku dosen pembimbing II
6. Semua pihak yang telah membantu terselesaikannya skripsi ini.

Penyusun menyadari bahwa skripsi masih jauh dari sempurna, oleh karena itu penyusun mengharapkan kritik dan saran dari pembaca. Semoga skripsi ini bisa bermanfaat bagi para pembaca sekalian.

Malang, Februari 2013

Penyusun

DAFTAR ISI

Halaman Judul	i
Lembar Persetujuan Skripsi.....	ii
Abstrak.....	iii
Kata Pengantar	iv
Daftar Isi	v
Daftar Gambar	vii
Daftar Tabel	viii
 BAB I.....	 1
PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan dan Manfaat	2
1.4 Batasan Masalah	2
1.5 Metodologi.....	3
1.6 Sistematika Penulisan.....	3
 BAB II	 5
LANDASAN TEORI.....	5
2.1 Computer Vision.....	5
2.2 Pengolahan Citra.....	5
2.2.1 Citra Analog.....	6
2.2.2 Citra Digital.....	7
2.2.3 Digitalisasi Citra.....	7
2.3 Citra RGB	8
2.4 Threshold RGB	8
2.5 Sistem Informasi.....	9
2.6 Data Base Oracle	11
2.7 Delphi.....	12
2.7.1 Keunggulan Delphi:.....	12
2.7.2 Spesifikasi Minimum Sistem untuk Instalasi Delphi	13
2.7.3 Tampilan Delphi	13
2.8 Pengujian Black Box	18
2.9 Membuat Database Link menggunakan Oracle XE	20
 BAB III.....	 22
PERANCANGAN DAN PEMBUATAN SISTEM.....	22
3.1 Dynamic Color Thresholding	22
3.1.2 SEGMENTASI GAMBAR.....	25

3.1.3	Kombinasi Warna	26
3.2	Perancangan Perangkat Lunak (Software).....	29
3.2.2	Desain Data Flow Diagram	31
3.2.3	Perancangan Database	33
3.2.3	ERD	38
3.3	Perangkat Keras (Hardware).....	39
3.3.1	Desain Color Code.....	39
3.3.2	Perancangan Box Pembacaan Kode Warna.....	41
BAB IV		46
IMPLEMENTASI DAN PENGUJIAN		46
4.1	Implementasi Color Recognition.....	46
4.1.1	Pembacaan Kode Warna	46
4.2	Integrasi Sistem Informasi Gudang	48
4.2.1	Form Login.....	49
4.2.2	Menu Utama	49
4.2.3	Form Data Barang	50
4.2.4	Form Denah Lokasi	51
4.2.4	Form Penerimaan PO.....	52
4.2.5	Form Pengeluaran Barang	52
4.2.6	Cetak Laporan.....	53
4.2.7	History transaksi	55
4.2.8	Grafik Report.....	55
4.2.9	Form Import Data	57
4.2.10	Form Import Data	57
4.3	Pengujian	58
4.4	Kelebihan dan Kekurangan.....	62
BAB V		63
PENUTUP		63
5.1.	Kesimpulan.....	63
5.2.	Saran	64

DAFTAR GAMBAR

Gambar 2-1. Gambar Matrik Ukuran N Baris dan M Kolom.....	8
Gambar 2-2. Gambar Cara Menentukan Warna Dengan Menggunakan Metode <i>Dynamic Color Thresholding</i>	9
Gambar 2-3. Tampilan Interface Borland Delphi 7	14
Gambar 2-4. Tampilan Menu Bar	15
Gambar 2-5. Tampilan Component Palette.....	16
Gambar 2-6. Tampilan Lembar kerja Form.....	16
Gambar 2-7. Tampilan Lembar kerja Code Editor	17
Gambar 2-8. Tampilan Lembar kerja Object Inspector	18
Gambar 2-9. Tampilan Object Tree View.....	18
Gambar 2-10. Skema Client Server	20
Gambar 3-1. Diagram Blok Pembacaan Warna.....	23
Gambar 3-2. Skema Pembacaan <i>Color Code</i> dengan kode 000746008.....	25
Gambar 3-3. Gambar <i>Color Code</i> Yang Di Deteksi	28
Gambar 3-4. Diagram Blok Desain Sistem software yang akan dibuat	30
Gambar 3-5. Rancangan DFD Level 0	31
Gambar 3-6. Rancangan DFD Level 1	32
Gambar 3-7. Type Status Variabel Pada Database Setting	33
Gambar 3-8. Type Status Variabel Pada Database Stok Barang.....	34
Gambar 3-9. Type status variabel pada database Setting.....	35
Barang Belum Memiliki Barcode.....	35
Gambar 3-10. Type Status Variabel Pada Database Setting	35
History Penerimaan	35
Gambar 3-11. Type Status Variabel Pada Database History Penerimaan.....	36
Gambar 3-12. Type Status Variabel Pada Database Penerimaan Barang	36
Gambar 3-13. Tabel Pengeluaran Barang.....	37
Gambar 3-14. Type Status Variabel Pada Database User	37
Gambar 3-15. Desain ERD Database Aplikasi	38

Gambar 3-16. Tampilan Depan <i>Color Code</i>	40
Gambar 3-18. Tampilan Belakang <i>Color Code</i>	41
Gambar 3-19. Tampilan Box Pembaca Kartu.....	42
Gambar 3-20. Desain Box Pembaca Kartu.....	42
Gambar 3-21. Tampilan Box Pembaca Kartu yang Sudah Jadi.....	43
Gambar 3-22. Desain PCB Sistem Pencahayaan Dengan Lampu Led Di Dalam Box	44
Gambar 3-23. USB Hub Digunakan Untuk Menyambungkan Antara Web Cam Dengan Lampu Led.....	44
Gambar 4-1. Form Pembacaan <i>Color Code</i>	47
Gambar 4-2. Memasukan <i>Color Code</i> Ke Dalam Box Reader	48
Gambar 4-3. Memasukan <i>Color Code</i> Ke Dalam Box Reader	48
Gambar 4-3. Form Login Untuk Validasi Hak Akses.....	49
Gambar 4-4. Menu Utama Program	50
Gambar 4-5. Form pengolah data barang	51
Gambar 4-6. Form pengolah data barang	51
Gambar 4-7. Form pengolah PO.....	52
Gambar 4-8. Form penerimaan barang.....	53
Gambar 4-9. Form laporan penerimaan barang.....	53
Gambar 4-10. Hasil cetak laporan penerimaan.....	54
Gambar 4-11. Form Laporan Pengeluaran	54
Gambar 4-12. Hasil Cetak Laporan Penerimaan.....	55
Gambar 4-13. History Transaksi.....	56
Gambar 4-14. Laporan Grafik Stok Barang	56
Gambar 4-15. Import Database Dari File Excel	57
Gambar 4-16. Export Database Menjadi File Excel.....	58
Gambar 4-17. Tampilan Sampel Yang Akan Di Uji	59

DAFTAR TABEL

Tabel 3-2. Pola Urutan Pembacaan <i>Color Code</i>	26
Tabel 3-3. Tabel Kombinasi <i>Color Code</i>	27
Tabel 3-4. Tabel Pembacaan Pola warna	28
Tabel 4-1. Hasil Pengujian Tahap 1	60
Tabel 4-2. Hasil Pengujian Tahap 2.....	61

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dengan menggunakan komposisi warna yang kompleks bisa diciptakan sebuah kombinasi kode untuk memudahkan identifikasi. Metode ini bisa membantu dalam efisiensi waktu dalam sebuah pekerjaan. Dengan memanfaatkan sebuah citra digital dan menganalisa tiap piksel yang sudah ditentukan maka bisa dibuat sebuah kode yang mempercepat dalam pengenalan suatu kode. Selain itu kode yang tersambung dengan database lebih mempermudah dalam mengakses informasi dari kode yang telah dikenali. Sebuah sistem informasi dengan dukungan komputer vision sebagai identifikasi kode akan menambah efisiensi pekerjaan. Selain itu database yang saling terintegrasi antara client dan server akan lebih memudahkan dalam distribusi data dan menghindari redundansi data antara client dan server.

Metode yang di pakai dalam perancangan sistem ini memakai metode *Dynamic Color Threshold*. Metode ini dipilih karena termasuk metode sederhana dalam pendeteksian warna yaitu dengan cara memberikan ambang batas nilai RGB untuk menentukan nilai intensitas warna yang ada pada sebuah citra. Karena metode ini cukup sederhana maka akan lebih memaksimalkan waktu dalam pembacaan.

Dengan menggunakan metode pengenalan warna yang akan diimplementasikan pada pengenalan kode gudang. Perancangan ini akan mengambil studi kasus pada Gudang PLTA Sutami karena pencatatan disana masih menggunakan cara manual dengan pencatatan menggunakan buku. Dan dengan database yang saling terintegrasi akan bertujuan dalam distribusi data antara Gudang dan kantor pusat.

1.2 Rumusan Masalah

Yang menjadi rumusan masalah dalam penyusunan skripsi ini adalah bagaimana cara supaya warna bisa dijadikan sebuah kode yang berisi informasi detail dari setiap barang yang ada pada gudang dan dapat terintegrasi dengan kantor pusat yang menggunakan Database Oracle dalam melakukan kegiatannya.

1.3 Tujuan dan Manfaat

Tujuan perancangan sistem yang akan dibuat adalah dapat membuat sebuah jenis code baru yang berdasarkan pada intensitas warna yang digunakan untuk melakukan pembacaan sebuah karakter. Dengan kode warna yang masih baru maka tingkat kerahasiaan dalam merahasiakan suatu kode akan lebih terjamin. Selain itu dengan pembuatan sistem informasi ini diharapkan distribusi data yang terintegrasi dengan oracle akan membantu proses tukar data.

Manfaat dengan perancangan sistem ini dapat Mempercepat dalam pengenalan kode barang dan jenis barang sehingga lebih efisien. Selain itu Pencatatan data gudang akan lebih teratur dan mudah dalam melakukan pencarian data serta dengan adanya distribusi data antara kantor pusat dan gudang maka akan menghindari redundansi data.

1.4 Batasan Masalah

Batasan permasalahan dalam Implementasi color recognition dengan metode dynamic color threshold dalam sistem informasi gudang PLTA sutami:

1. Definisi warna yang mewakili suatu kode akan ditentukan langsung dalam script sehingga warna yang ditetapkan sebagai kode yaitu RGB untuk lebih mempermudah dalam proses identifikasi.
2. Karakter yang dipakai dalam sistem ini berjumlah 9 karakter yang bernilai bilangan desimal dari 0 sampai 9.

3. Database yang terintegrasi masih ada pada kantor distrik dan kantor sektor karena pada PT PJB Up Brantas terdapat lebih dari 4 kantor distrik dan pada implementasi ini hanya mengambil sampel satu kantor distrik dan satu kantor sektor.
4. Dalam pengaturan topologi tidak dibahas dikarenakan lebih fokus pada integrasi database link tidak membahas tentang jaringan yang digunakan.
5. Dalam pembacaan kartu menggunakan tempat yang sudah disediakan khusus yang telah dipasang sebuah web cam.

1.5 Metodologi

Metode yang akan digunakan dalam tugas akhir ini terdiri dari langkah-langkah berikut:

- a. Mengumpulkan referensi data untuk membangun sistem yang akan dibuat.
- b. Perancangan pembuatan sistem informasi yang akan ditambahkan dengan program color recognition.
- c. Mempersiapkan kamera dan membuat sebuah tempat khusus untuk melakukan pembacaan kode warna.
- d. Pembuatan desain dari code warna yang akan digunakan dalam program dan menentukan berapa maksimal kombinasi yang bisa dipakai dalam code warna yang akan dipakai
- e. Pengujian pembacaan code warna yang akan dibuat batas maksimal pembacaan pixel pada kamera
- f. Membuat kesimpulan dari pengujian pembacaan code warna apakah sudah sesuai dengan hasil yang diinginkan

1.6 Sistematika Penulisan

Untuk mempermudah dan memahami pembahasan penulisan skripsi ini, maka sistematika penulisan disusun sebagai berikut :

Bab I : Pendahuluan

Berisi Latar Belakang, Rumusan Masalah, Pembatasan Masalah, Tujuan dan manfaat, Metodologi Penelitian dan Sistematika Penulisan.

Bab II : Landasan Teori

Berisi teori-teori yang didapat dari studi literatur dan konsep-konsep yang terkait dengan tugas akhir ini, beserta dengan penyelesaian masalah yang diambil dalam penyusunan tugas akhir.

Bab III : Perancangan dan Pembuatan Sistem

Dalam bab ini berisi mengenai analisa dan perancangan dari sistem yang akan dibangun meliputi analisa sistem, komponen sistem pakar, rancangan basis data dan perancangan antarmuka

Bab IV : Implementasi dan Pengujian

Berisi tentang implementasi dan cara menjalankan aplikasi serta uji coba dari program yang telah dibuat tersebut.

Bab V : Penutup

Merupakan bab terakhir yang memuat intisari dari hasil pembahasan yang berisikan kesimpulan dan saran yang dapat digunakan sebagai pertimbangan untuk pengembangan penulisan selanjutnya.

BAB II

LANDASAN TEORI

2.1 Computer Vision

Computer Vision adalah ilmu dan teknologi mesin yang melihat, di mana mesin mampu mengekstrak informasi dari gambar yang diperlukan untuk menyelesaikan tugas tertentu. Sebagai suatu disiplin ilmu, visi komputer berkaitan dengan teori di balik sistem buatan bahwa ekstrak informasi dari gambar. Data gambar dapat mengambil banyak bentuk, seperti urutan video, pandangan dari beberapa kamera, atau data multi-dimensi dari scanner medis. Sedangkan sebagai disiplin teknologi, computer vision berusaha untuk menerapkan teori dan model untuk pembangunan sistem computer vision.

Computer Vision didefinisikan sebagai salah satu cabang ilmu pengetahuan yang mempelajari bagaimana komputer dapat mengenali obyek yang diamati. Cabang ilmu ini bersama Artificial Intelligence akan mampu menghasilkan Visual Intelligence System. Perbedaannya adalah Computer Vision lebih mempelajari bagaimana komputer dapat mengenali obyek yang diamati. Namun komputer grafik lebih ke arah pemanipulasian gambar (visual) secara digital. Bentuk sederhana dari grafik komputer adalah grafik komputer 2D yang kemudian berkembang menjadi grafik komputer 3D, pemrosesan citra, dan pengenalan pola. Grafik komputer sering dikenal dengan istilah visualisasi data.

2.2 Pengolahan Citra

Pengolahan citra merupakan proses pengolahan dan analisis citra yang banyak melibatkan persepsi visual. Proses ini mempunyai ciri data masukan dan informasi keluaran yang berbentuk citra. Istilah pengolahan citra digital secara umum

didefinisikan sebagai pemrosesan citra dua dimensi dengan komputer. Dalam definisi yang lebih luas, pengolahan citra digital juga mencakup semua data dua dimensi. Citra digital adalah barisan bilangan nyata maupun kompleks yang diwakili oleh bit-bit tertentu.

Umumnya citra digital berbentuk persegi panjang atau bujur sangkar (pada beberapa sistem pencitraan ada pula yang berbentuk segi enam) yang memiliki lebar dan tinggi tertentu. Ukuran ini biasanya dinyatakan dalam banyaknya titik atau piksel sehingga ukuran citra selalu bernilai bulat. Setiap titik memiliki koordinat sesuai posisinya dalam citra. Koordinat ini biasanya dinyatakan dalam bilangan bulat positif, yang dapat dimulai dari 0 atau 1 tergantung pada sistem yang digunakan. Setiap titik juga memiliki nilai berupa angka digital yang merepresentasikan informasi yang diwakili oleh titik tersebut.

Pengolahan citra digital digunakan untuk memproses citra sesuai dengan keperluan. Sebelum membahas tentang pengolahan citra digital, sebaiknya kita mengetahui terlebih dahulu apa yang dimaksud citra itu. Citra adalah representasi atau tiruan dari suatu benda atau objek. Citra dibagi menjadi dua, yaitu citra analog dan citra digital. Citra analog adalah citra yang dihasilkan sinyal kontinyu, misalnya foto yang dicetak di kertas foto, citra yang tampil di layar TV, citra yang dihasilkan oleh CT-scan, citra yang tersimpan dalam pita kaset, dll. Sedangkan citra digital adalah citra yang bisa diolah langsung oleh komputer dan tersimpan dalam media simpan digital misalnya memory, harddisk, CD, dll. Contoh dari citra digital ini adalah foto yang dihasilkan oleh kamera digital, citra yang dihasilkan oleh scanner, dll.

2.2.1 Citra Analog

Citra analog adalah citra yang bersifat kontinu, seperti gambar pada monitor televisi, foto sinar X, foto yang ter cetak di kertas foto, lukisan pemandangan alam, hasil CT scan, lembar gambar yang terekam pada pita kaset. Citra analog tidak dapat direpresentasikan dalam komputer sehingga tidak bisa diproses di komputer secara

langsung. Oleh sebab itu, agar citra analog ini dapat diproses di komputer, proses konversi analog ke digital harus dilakukan terlebih dahulu. Citra analog dihasilkan dari alat-alat analog, seperti video kamera analog, kamera foto analog, Web Cam, CT scan dan lain-lain.

2.2.2 Citra Digital

Citra digital adalah citra yang dapat diolah oleh komputer. Citra digital dapat di Definisikan sebagai fungsi dua variabel, $f(x, y)$, dimana x dan y adalah koordinat spasial dan nilai $f(x, y)$ adalah intensitas citra pada koordinat. Sebuah citra digital dapat diwakili oleh sebuah matrix yang terdiri dari kolom dan baris, dimana per potongan antara kolom dan baris disebut piksel (piksel = picture element), element terkecil dari sebuah citra. Piksel mempunyai dua parameter, yaitu koordinat dan intensitas atau warna.

2.2.3 Digitalisasi Citra

Agar dapat diolah dengan dengan komputer digital, maka suatu citra harus direpresentasikan secara numerik dengan nilai-nilai *diskrit*. Representasi citra dari fungsi malar (kontinu) menjadi nilai-nilai diskrit disebut *digitalisasi*. Citra yang dihasilkan inilah yang disebut citra digital (*digital image*). Pada umumnya citra digital berbentuk empat persegi panjang, dan dimensi ukurannya dinyatakan sebagai tinggi' lebar (atau lebar ' panjang).

Citra analog tidak bisa langsung diolah tetapi harus melalui proses digitalisasi karena komputer hanya mengerti bahasa 0 dan 1. Proses mengubah citra analog menjadi citra digital disebut digitalisasi citra. Ada dua hal yang harus dilakukan pada digitalisasi citra, yaitu digitalisasi spasial yang disebut juga sebagai sampling (penerokan) dan digitalisasi intensitas yang sering disebut sebagai kuantitas. Citra digital yang berukuran $N \times M$ lazim dinyatakan dengan matriks yang berukuran N baris dan M kolom sebagai berikut:

$$f(x,y) \approx \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M) \\ f(1,0) & f(1,1) & \dots & f(1,M) \\ \vdots & \vdots & \vdots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix}$$

Gambar 2-1. Gambar Matrik Ukuran N Baris dan M Kolom

2.3 Citra RGB

RGB adalah singkatan dari Red - Blue - Green adalah model warna pencahayaan (additive color mode) dipakai untuk "input devices" seperti scanner maupun "output devices" seperti display monitor, warna-warna primernya (Red, Blue, Green) tergantung pada teknologi alat yang dipakai. Setiap piksel pada citra warna mewakili warna yang merupakan kombinasi dari tiga warna dasar (RGB = Red Green Blue). Setiap warna dasar menggunakan penyimpanan 8 bit = 1 byte, yang berarti setiap warna mempunyai gradasi sebanyak 255 warna. Berarti setiap piksel mempunyai kombinasi warna sebanyak $28 \cdot 28 \cdot 28 = 224 = 16$ juta warna lebih. Itulah sebabnya format ini dinamakan true color karena mempunyai jumlah warna yang cukup besar sehingga bisa dikatakan hampir mencakup semua warna di alam.

2.4 Threshold RGB

Untuk warna-warna dasar, nilai RGB cukup efektif dalam melakukan deteksi. Nilai threshold dapat dipelajari berdasarkan pola warna, tetapi threshold RGB ini sangat rentan terhadap kestabilan cahaya. Aplikasi untuk menentukan nilai threshold terbaik dalam suatu permasalahan dapat dilakukan menggunakan teknik-teknik machine learning. Threshold dapat dipilih apakah threshold global atau threshold local. Threshold menggunakan distance threshold dari warna-warna di sekitar obyek yang dimaksud. Sebelumnya diambil gambar-gambar contoh sebagai acuan untuk menentukan thresholding dari warna yang diinginkan.

Untuk menentukan warna RGB dengan metode thresholding

IF $r > 200$ and $g < 64$ and $b < 64$ THEN warna = merah

Nilai – nilai diatas adalah nilai threholding yang menjadi batasan dari penentuan warna

Gambar 2-2. Gambar Cara Menentukan Warna Dengan Menggunakan Metode *Dynamic Color Thresholding*

Pada gambar 2-2, data warna-warna tersebut diambil rata-rata dari setiap elemen warna. Thresholding dilakukan dengan menghitung setiap rata-rata elemen warna. Apabila nilai di antara ketiga warna tersebut ada yang lebih dominan maka nilai dari piksel itu adalah warna yang dominan.

2.5 Sistem Informasi

Sistem informasi adalah suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan informasi yang diperlukan untuk pengambilan keputusan. Sistem informasi dalam suatu organisasi dapat dikatakan sebagai suatu sistem yang menyediakan informasi bagi semua tingkatan dalam organisasi tersebut kapan saja diperlukan. Sistem ini menyimpan, mengambil, mengubah, mengolah dan mengkomunikasikan informasi yang diterima dengan menggunakan sistem informasi atau peralatan sistem lainnya.

Sistem informasi terdiri dari komponen-komponen yang disebut blok bangunan (building blok), yang terdiri dari komponen input, komponen model, komponen output, komponen teknologi, komponen hardware, komponen software, komponen

basis data, dan komponen kontrol. Semua komponen tersebut saling berinteraksi satu dengan yang lain membentuk suatu kesatuan untuk mencapai sasaran.

1. Komponen input

Input mewakili data yang masuk ke dalam sistem informasi. Input disini termasuk metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2. Komponen model

Komponen ini terdiri dari kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

3. Komponen output

Hasil dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua pemakai sistem.

4. Komponen teknologi

Teknologi merupakan “tool box” dalam sistem informasi, Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran, dan membantu pengendalian dari sistem secara keseluruhan.

5. Komponen hardware

Hardware berperan penting sebagai suatu media penyimpanan vital bagi sistem informasi. Yang berfungsi sebagai tempat untuk menampung database atau lebih

mudah dikatakan sebagai sumber data dan informasi untuk memperlancar dan mempermudah kerja dari sistem informasi.

6. Komponen software

Software berfungsi sebagai tempat untuk mengolah, menghitung dan memanipulasi data yang diambil dari hardware untuk menciptakan suatu informasi.

2.6 Data Base Oracle

Oracle adalah relational database management system (RDBMS) untuk mengelola informasi secara terbuka, komprehensif dan terintegrasi. Oracle Server menyediakan solusi yang efisien dan efektif karena kemampuannya dalam hal sebagai berikut:

1. Dapat bekerja di lingkungan client/server (pemrosesan tersebar)
2. Menangani manajemen space dan basis data yang besar
3. Mendukung akses data secara simultan
4. Performansi pemrosesan transaksi yang tinggi
5. Menjamin ketersediaan yang terkontrol
6. Lingkungan yang terreplikasi

Database merupakan salah satu komponen dalam teknologi informasi yang mutlak diperlukan oleh semua organisasi yang ingin mempunyai suatu sistem informasi yang terpadu untuk menunjang kegiatan organisasi demi mencapai tujuannya.

2.7 Delphi

Borland Delphi merupakan program aplikasi database yang berbasis Object Pascal dari Borland. Selain itu, Delphi juga memberikan fasilitas pembuatan aplikasi visual. Delphi merupakan pilihan dalam pembuatan aplikasi visual karena memberikan produktivitas yang tinggi.

Delphi awalnya sebuah proyek penelitian rahasia di Borland yang berevolusi menjadi sebuah produk yang disebut AppBuilder. Tak lama sebelum rilis pertama dari Borland AppBuilder, Novell AppBuilder di rilis, meninggalkan Borland membutuhkan nama baru.

Pengembang Danny Thorpe memilih nama kode *Delphi* di referensi ke Oracle di Delphi. Salah satu tujuan asli dari Delphi adalah untuk menyediakan konektivitas database untuk programmer sebagai kunci dan sebuah paket database yang populer pada saat itu adalah database Oracle, maka, “Jika Anda ingin berbicara dengan Oracle, pergi ke Delphi”. Sebagai perkembangan lebih lanjut, nama mereka mulai berkembang dan ada dukungan yang tumbuh di dalam nama Borland Delphi.

2.7.1 Keunggulan Delphi:

Dalam perancangan sistem ini digunakan Borland Delphi 7. Karena Borland delphi7 memiliki beberapa keunggulan yang dibutuhkan dalam perancangan sistem ini. Berikut ini adalah keunggulan yang dimiliki oleh Borland Delphi7:

1. Memiliki IDE (*integrated Development Environment*) atau lingkungan pengembangan terintegrasi yang didalamnya terdapat menu-menu yang memudahkan programmer untuk membuat sebuah program aplikasi.
2. Proses kompilasi cepat, pada saat program dijalankan, secara otomatis akan dibaca sebagai sebuah program, tanpa dijalankan terpisah.
3. Mudah digunakan, karena source code Delphi merupakan turunan dari bahasa pemrograman Pascal.

4. Bersifat multi purpose, artinya bahasa pemrograman Delphi dapat digunakan untuk mengembangkan berbagai keperluan pengembangan aplikasi.
5. Dapat mengkompilasi menjadi single executable, memudahkan distribusi dan meminimalisir masalah yang terkait dengan versioning.

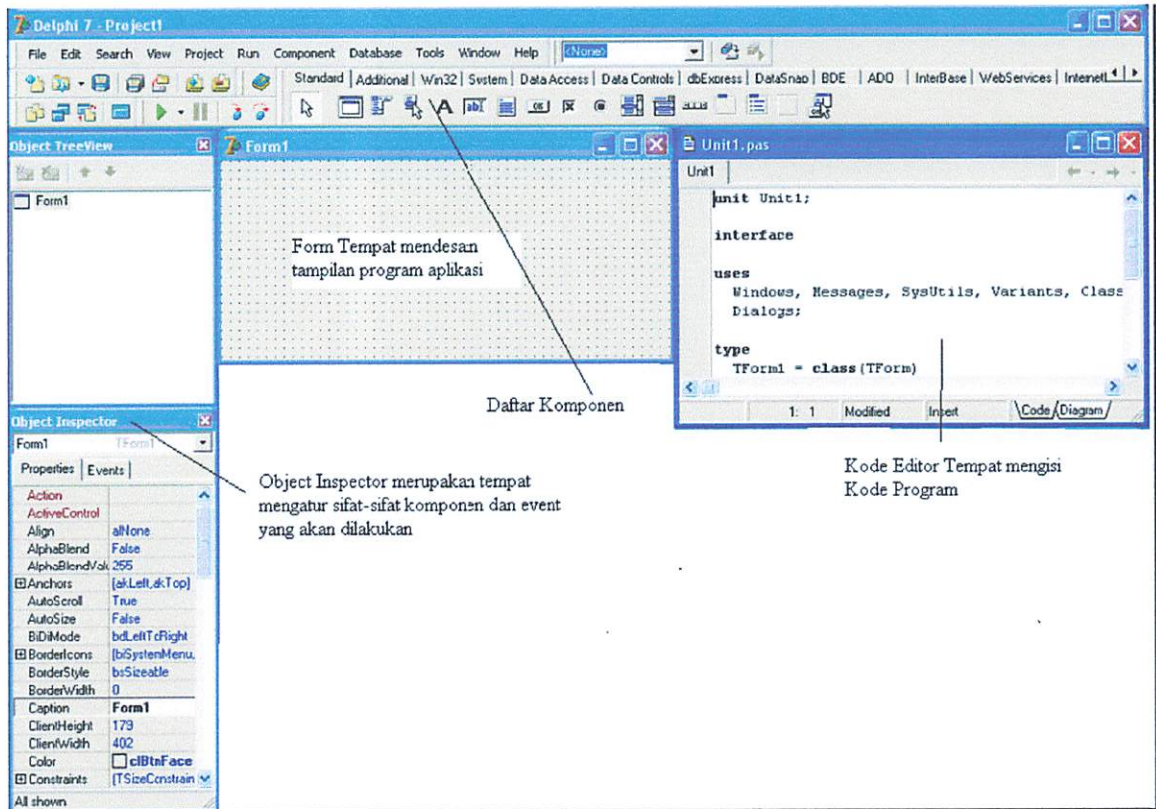
2.7.2 Spesifikasi Minimum Sistem untuk Instalasi Delphi

Sisitem yang dibuat akan memerlukan beberapa hardware untuk disinkronisasikan dengan software. Hardware digunakan juga untuk mengambil citra analog untuk bisa di olah oleh software. Berikut ini adala spesifikasi minimum yang dibutuhkan:

- a. Intel Pentium 166 MHz or higher (P2 400 MHz recommended)
- b. Microsoft Windows 98, 2000, and XP
- c. 256 Mb
- d. Approximate hard disk space required for a full install: 475 Mb (Enterprise edition)
- e. CD-ROM drive
- f. VGA or higher resolution monitor
- g. Mouse or other pointing device

2.7.3 Tampilan Delphi

Tampilan utama dalam merancang sistem pada program Borland Delphi 7 adalah seperti ditunjukkan pada gambar berikut:



Gambar 2-3. Tampilan Interface Borland Delphi 7

IDE (*Integrated Development Environment*) atau lingkungan pengembangan terpadu pada program Delphi terbagi menjadi delapan bagian utama, yaitu:

1. Main Window
2. ToolBar
3. Component Palette
4. Form Designer
5. Code Editor
6. Object Inspector
7. Object TreeView

Untuk lebih jelasnya dapat dilihat pada gambar 2-3 di atas. IDE merupakan sebuah lingkungan di mana semua tombol perintah yang diperlukan untuk mendesain aplikasi, menjalankan dan menguji suatu aplikasi disajikan dengan baik untuk memudahkan pengembangan program.

1. Menu Bar

Berfungsi untuk memilih tugas-tugas tertentu, seperti memulai, membuka, dan menyimpan *project*, mengompilasi *project* menjadi *file executable* (EXE), dan lain-lain



Gambar 2-4. Tampilan Menu Bar

2. Toolbar

Delphi memiliki beberapa toolbar yang masing-masing memiliki perbedaan fungsi dan setiap tombol pada bagian toolbar berfungsi sebagai pengganti suatu menu perintah yang sering digunakan.



Gambar 2-4. Tampilan Toolbar

Toolbar terletak pada bagian bawah baris menu. Pada kondisi default Delphi memiliki enam bagian toolbar, antara lain: Standart, View, Debug, Desktops, Custom dan Component Palette.

3. Component Palette

Component Palette berisi kumpulan ikon yang melambangkan komponen-komponen yang terdapat pada *VCL (Visual Component Library)*. Pada Componen

Palette Anda akan menemukan beberapa page control, seperti Standart, Additional, Win32, System, Data Access dan lain-lain seperti tampak pada gambar 4

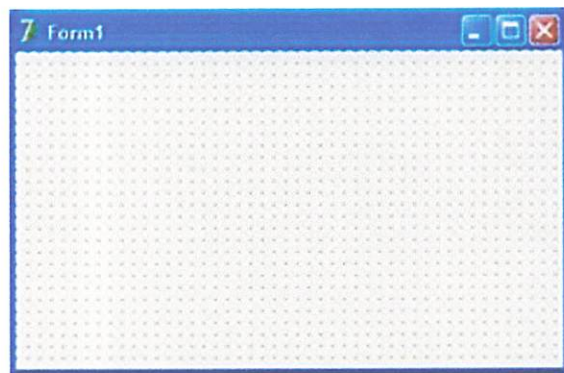


Gambar 2-5. Tampilan Component Palette

Ikon tombol Pointer terdapat di setiap page control. Tombol ini dipakai untuk menekan atau memilih posisi. Jika Anda memilih sebuah item dari sebuah page control, tombol pointer ini akan berada dalam keadaan tidak aktif. Hal ini berarti Anda akan meletakkan komponen pada form, Anda cukup klik pada form.

4. Form Designer

Merupakan suatu objek yang dapat dipakai sebagai tempat untuk merancang program aplikasi. Form berbentuk sebuah meja kerja yang dapat diisi dengan komponen-komponen yang diambil dari Component Palette. Pada saat Anda memulai Delphi, Delphi akan memberikan sebuah form kosong yang disebut form1, seperti gambar 5 di bawah ini.

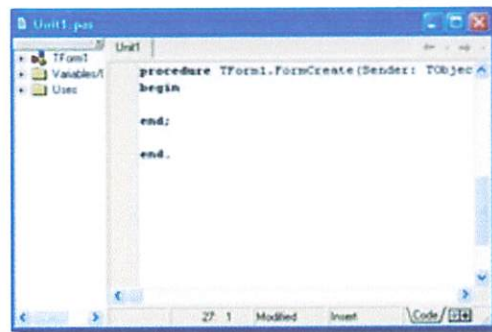


Gambar 2-6. Tampilan Lembar kerja Form

Sebuah form mengandung unit yang berfungsi untuk mengendalikan form dan Anda dapat mengendalikan komponen- komponen yang terletak dalam form dengan menggunakan Object Inspector dan Code Editor.

5. Code Editor

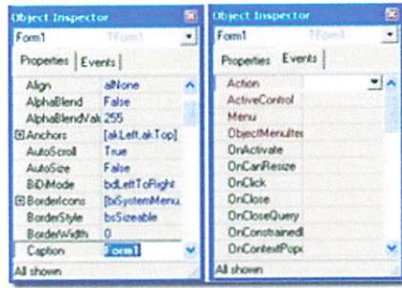
Code Editor merupakan tempat di mana Anda dapat menuliskan kode program. Pada bagian ini Anda dapat menuliskan pernyataan-pernyataan dalam Object Pascal. Satu diantara keuntungan bagi pengguna Delphi adalah bahwa Anda tidak perlu menuliskan kode-kode sumber, karena Delphi telah menyediakan kerangka penulisan sebuah program seperti pada gambar di bawah ini.



Gambar 2-7. Tampilan Lembar kerja Code Editor

6. Object Inspector

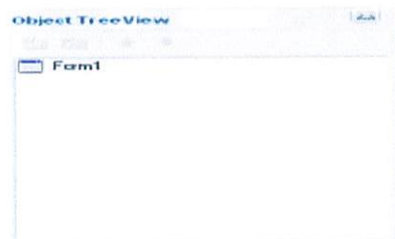
Digunakan untuk mengubah properti atau karakteristik dari sebuah komponen. Object Inspector terdiri dari dua tab, yaitu **Properties** dan **Events** seperti gambar 7 di bawah ini.



Gambar 2-8. Tampilan Lembar kerja Object Inspector

7. Object TreeView

Object TreeView menampilkan diagram pohon dari komponen-komponen yang bersifat visual maupun nonvisual yang telah terdapat dalam form, data module, atau frame. Object TreeView juga menampilkan hubungan logika antar komponen. Apabila Anda mengklik kanan salah satu item yang terdapat di dalam diagram pohon, Anda dapat melihat konteks menu komponen versi sebelumnya. Untuk mengakses menu secara penuh, klik kanan pada komponen yang sama dalam form, data module, atau frame.



Gambar 2-9. Tampilan Object Tree View

2.8 Pengujian Black Box

Metode uji coba black box memfokuskan pada keperluan fungsional dari software. Karena itu uji coba black box memungkinkan pengembang software untuk

membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program. Uji coba black box bukan merupakan alternatif dari uji coba white box, tetapi merupakan pendekatan yang melengkapi untuk menemukan kesalahan lainnya, selain menggunakan metode white box. Uji coba black box berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya :

1. Fungsi-fungsi yang salah atau hilang
2. Kesalahan interface
3. Kesalahan dalam struktur data atau akses database eksternal
4. Kesalahan performa
5. Kesalahan inisialisasi dan terminasi

Tidak seperti metode white box yang dilaksanakan di awal proses, uji coba black box diaplikasikan di beberapa tahapan berikutnya. Karena uji coba black box dengan sengaja mengabaikan struktur kontrol, sehingga perhatiannya difokuskan pada informasi domain. Uji coba didesain untuk dapat menjawab pertanyaan seperti berikut ini:

1. Bagaimana validitas fungsional nya diuji?
2. Jenis input seperti apa yang akan menghasilkan kasus uji yang baik?
3. Apakah sistem secara khusus sensitif terhadap nilai input tertentu?
4. Bagaimana batasan-batasan kelas data diisolasi?
5. Berapa rasio data dan jumlah data yang dapat di toleransi oleh sistem?
6. Apa akibat yang akan timbul dari kombinasi spesifik data pada operasi sistem?

Dengan mengaplikasikan uji coba black box, diharapkan dapat menghasilkan sekumpulan kasus uji yang memenuhi kriteria berikut:

1. Kasus uji yang berkurang, jika jumlahnya lebih dari 1, maka jumlah dari uji kasus tambahan harus didesain untuk mencapai uji coba yang cukup beralasan.

2. Kasus uji yang memberitahukan sesuatu tentang keberadaan atau tidaknya suatu jenis kesalahan

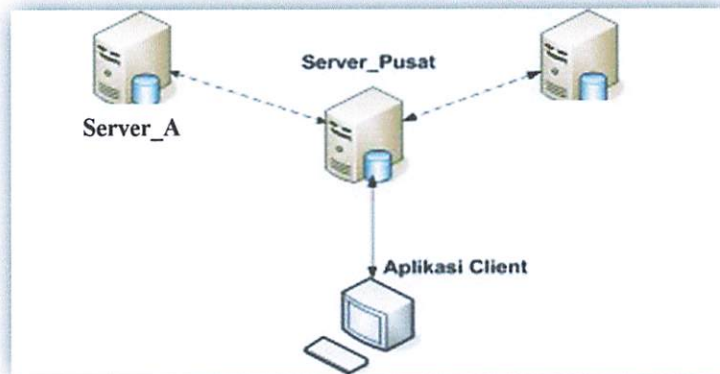
Pada siklus hidup pengembangan perangkat lunak (*System Development Life Cycle: SDLC*) salah satu proses yang harus dilakukan adalah proses pengujian (*testing*). Pengujian perangkat lunak adalah suatu teknik yang digunakan menguji apakah sebuah perangkat lunak yang dihasilkan telah memenuhi kebutuhan proses bisnis pengguna atau masih belum.

2.9 Membuat Database Link menggunakan Oracle XE

Konsep utama dari sistem basis data distribusi adalah database link. Database link adalah suatu koneksi antara dua server database fisik yang memungkinkan client untuk mengakses datanya sebagai satu database logic.

Apa sebenarnya keuntungannya kita menggunakan sistem seperti ini?

1. Aplikasi client tidak langsung terhubung dengan database sumbernya
2. Aplikasi client tidak tahu lokasi asal data sebenarnya
3. Lebih secure tentunya



Gambar 2-10. Skema Client Server

Pada gambar 2-10, diasumsikan bahwa kita ingin mengkomunikasikan antara Server_Pusat dengan Server_A, berikut detail informasi antar kedua database tersebut :

a. Server_Pusat

- User Schema : server
- User Password : server
- IP server_pusat : 192.168.0.1
- SID : XE
- Table : dosen

b. Server_A

- User Schema : client
- User Password : client
- IP server_pusat : 192.168.0.2
- SID : XE
- Table : mahasiswa

Contoh kasus, si Server_Pusat ingin mengetahui isi database dari si Server_A, dan sebaliknya

BAB III

PERANCANGAN DAN PEMBUATAN SISTEM

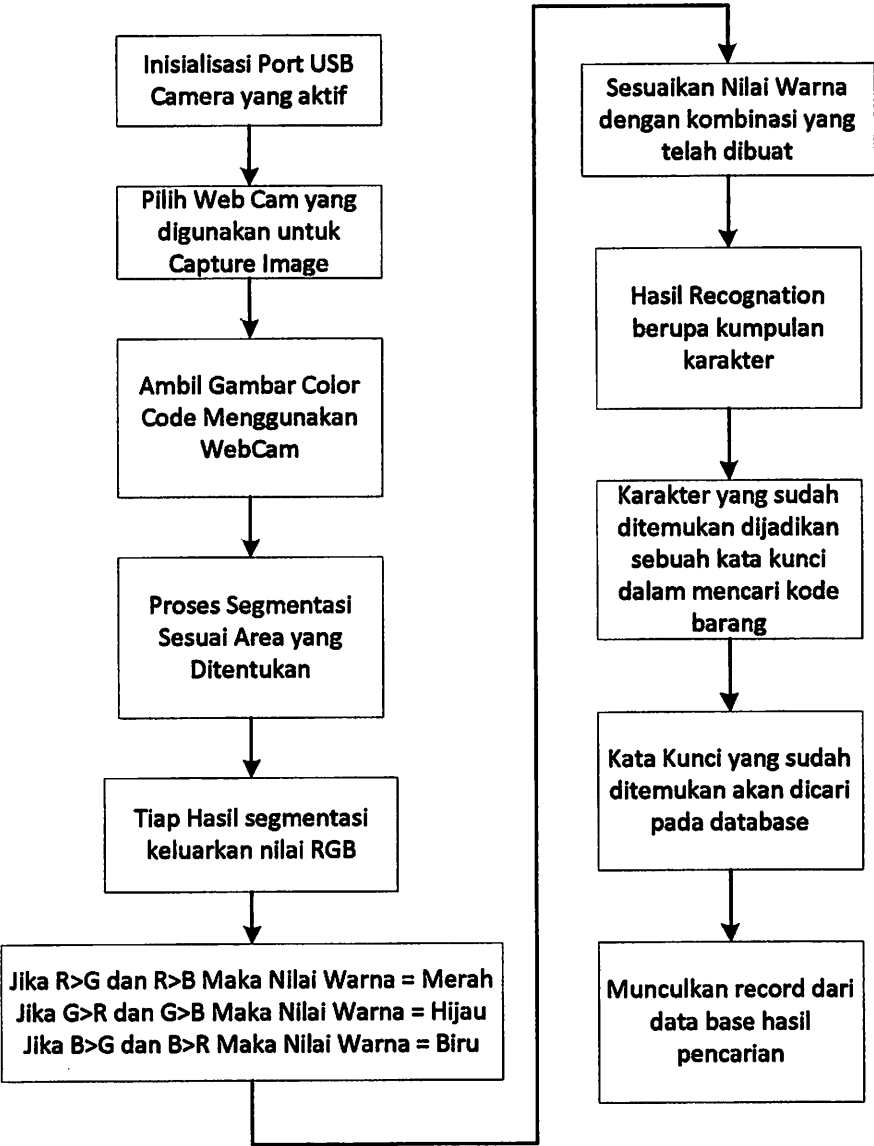
3.1 Dynamic Color Thresholding

Sistem yang akan di buat mengandung unsur *Computer Vision* karena menggunakan sebuah webcam untuk memasukan sebuah data. Dengan menggunakan media warna akan didapatkan sebuah karakter untuk mengetahui sebuah informasi tentang sesuatu barang. Warna adalah komponen yang akan diolah untuk menghasilkan sebuah karakter yang akan dijadikan sebuah kata kunci untuk diolah ke dalam database. Warna yang dipakai memiliki tipe RGB (*Red, Green, Blue*).

RGB adalah singkatan dari *Red - Green - Blue* adalah model warna pencahayaan (*additive color mode*) dipakai untuk "*input devices*" seperti scanner maupun "*output devices*" seperti display monitor, warna-warna primernya (*Red, Green, Blue*) tergantung pada teknologi alat yang dipakai seperti scanner atau digital camera, webcam. Warna RGB ini biasanya digunakan oleh monitor computer/TV. Warna yang dihasilkan berasal dari kombinasi 3 warna tersebut dan masing-masing memiliki nilai 8 bit merah, 8 bit hijau dan 8 bit untuk biru. Dalam komputer, nilai-nilai komponen sering disimpan sebagai angka integer antara 0 sampai 255, kisaran yang dapat ditampung sebuah bit (8-bit). Nilai ini dapat dituliskan dalam angka desimal maupun hexadecimal.

Sistem ini terbagi dalam beberapa proses yang saling berhubungan. Salah satu proses yang akan diproses yaitu proses recognition yaitu pengenalan *Color Code* sesuai dengan komposisi warna yang ditangkap oleh kamera webcam. Gambar 3-1 di bawah ini adalah gambaran diagram blok proses recognition.

DIAGRAM BLOK COLOR RECOGNITION



Gambar 3-1. Diagram Blok Pembacaan Warna

Pada langkah awal sistem akan inisialisasi port USB yang terhubung dengan webcam dan menampilkan pada combo box kemudian user akan memilih kamera yang dijadikan sebagai pembaca *color code*. Web cam akan mengambil gambar *Color Code* dan menampilkan berupa gambar digital pada komponen image selanjutnya akan di segmentasi sesuai dengan area yang ditentukan dan dikeluarkan nilai intensitas RGB yang ada pada setiap citra hasil segmentasi. Setelah diketahui nilai intensitas RGB setiap citra maka akan dilakukan metode *Dynamic Color Thresholding* untuk memunculkan nilai warna yang dominan pada citra. Dan dari nilai yang dihasilkan akan di cocok kan pada tabel kombinasi setiap 3 warna supaya diketahui karakter yang diwakili dari kombinasi warna. Setelah di hasil kan 9 karakter maka akan dijadikan sebagai kata kunci untuk menampilkan informasi sesuai dengan 9 karakter yang dianggap sebagai kode barang.

Pada sistem ini akan dilakukan pendeteksian warna dengan metode color thresholding yaitu dengan cara memberikan sebuah batas pada nilai RGB. Kombinasi warna yang akan dipakai untuk menentukan sebuah karakter ada 3 buah warna yaitu Merah, Hijau, Biru. Tabel 3-1 adalah tabel yang berisi tentang syarat nilai intensitas RGB pada sebuah image yang dianggap memiliki warna Merah, hijau dan Biru.

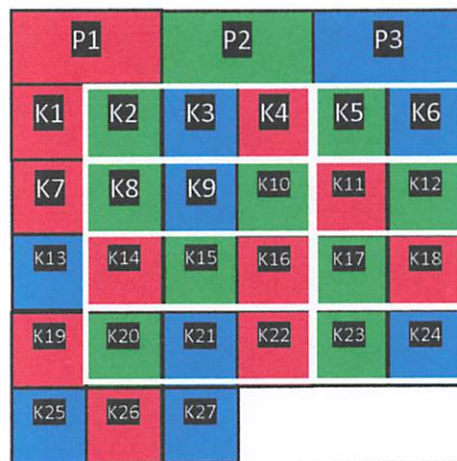
	Intensitas Nilai RGB		
Merah	$R > 180$	$G < 64$	$B < 64$
Hijau	$R < 64$	$R > 180$	$B < 64$
Biru	$R < 64$	$G < 64$	$R > 180$

Tabel 3-1. Tabel Penentuan nilai warna

Pada Tabel 3-1 adalah syarat nilai intensitas RGB untuk 3 warna yang dibutuhkan untuk dilakukan pengolahan citra digital. Sebelum dilakukan deteksi warna dilakukan dulu proses segmentasi. Nilai RGB yang tertera pada tabel adalah nilai batas yang dijadikan color threshold.

3.1.2 SEGMENTASI GAMBAR

Sebelum dilakukan proses pembacaan warna akan dilakukan proses segmentasi untuk memilih citra yang akan diproses dari hasil capture gambar, karena tidak semua image yang di capture akan diproses untuk mengurangi beban dari memori yang akan dipakai. Proses segmentasi ini akan dilakukan sebanyak 30 kali segmentasi, sehingga akan dihasilkan 30 gambar baru dari hasil segmentasi. 3 segmentasi pertama akan dilakukan untuk mengunci citra yang akan di proses. Setelah segmentasi yang pertama sudah dijalankan maka dilanjutkan dengan segmentasi kedua yaitu 27 kotak warna yang akan diterjemahkan ke dalam bilangan desimal. Hasil segmentasi akan ditampilkan ke dalam komponen Image milik Delphi. Ukuran citra yang dihasilkan dari 30×30 pixel untuk segmentasi pengunci ukurannya lebih besar yaitu 90×40 pixel. Dalam mengatur waktu dalam proses capture digunakan komponen timer yang akan mengatur interval dalam melakukan proses looping. Gambar 3-2 adalah gambaran dari proses segmentasi yang akan dilakukan



Gambar 3-2. Skema Pembacaan *Color Code* dengan kode 000746008

Pada gambar 3-2, tanda P1, P2, P3 adalah warna yang dijadikan pengunci ketika warna itu terdeteksi maka program akan melakukan capture gambar melalui

webcam. Dalam melakukan pembacaan ada pola tersendiri dalam melakukan pembacaan warna hasil segmentasi. Tabel 3-2 adalah pola pembacaan warna dari hasil segmentasi.

warna hasil segmentasi	Nomor Urut Karakter
P1	Tanda merah
P2	Tanda Hijau
P3	Tanda biru
K1,K2,K3	1
K4,K5,K6	2
K7,K8,K9	3
K10,K11,K12	4
K13,K14,K15	5
K16,K17,K18	6
K19,K20,K21	7
K22,K23,K24	8
25,K26,K27	9

Tabel 3-2. Pola Urutan Pembacaan *Color Code*

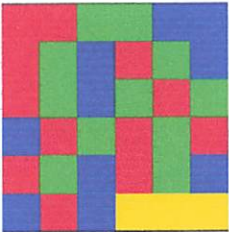
3.1.3 Kombinasi Warna

Untuk warna yang dibaca adalah warna RGB sehingga untuk menentukan suatu nilai maka dibutuhkan 3 kombinasi warna, Misalnya saja angka 1 di tentukan dengan kombinasi warna merah, biru, hijau. Sehingga dalam pembentukan kode warna di butuhkan matriks warna 5×6 agar bisa tersusun menjadi 9 angka yang mewakili suatu barang pada gudang. Tabel 3-3 adalah tabel kombinasi warna yang akan dilakukan sebagai basis pengetahuan dari proses recognition yang akan dilakukan. Tabel 3-3 adalah tabel kombinasi warna yang akan dipakai.

Kombinasi Warna	Warna	Karakter yang dihasilkan
	Merah, Hijau, Biru	0
	Merah, Biru, hijau	1
	Hijau, Merah, Biru	2
	Hijau, Biru, Merah	3
	Biru, Merah, Hijau	4
	Biru, Hijau, Merah	5
	Merah, hijau, Merah	6
	Hijau, Merah, Hijau	7
	Biru, Merah, Biru	8
	Merah, Biru, Merah	9

Tabel 3-3. Tabel Kombinasi *Color Code*

Pada tabel 3-3. Terdapat keterangan kombinasi angka yang dibutuhkan oleh sistem adalah karakter stok code. Kode yang dibutuhkan berupa angka maka dibutuhkan karakter 0-9. Dalam menentukan kombinasi ini belum ada aturan yang mengatur sebelumnya sehingga kombinasi ini menjadi kombinasi standard untuk *Color Code*. Contoh pembacaan *Color Code* yang telah di capture melalui webcam.



Gambar 3-3. Gambar *Color Code* Yang Di Deteksi

Pada gambar 3-3 adalah sebuah cuplikan gambar dari *Color Code*. Untuk pembacaanya bisa dilihat seperti pada tabel di bawah ini.

warna hasil segmentasi	Nomor Urut Karakter	Nilai warna	Hasil Karakter
K1,K2,K3	1	Merah, Hijau, Biru	0
K4,K5,K6	2	Merah, Hijau, Biru	0
K7,K8,K9	3	Merah, Hijau, Biru	0
K10,K11,K12	4	Hijau, Merah, Hijau	7
K13,K14,K15	5	Biru, Merah, Hijau	4
K16,K17,K18	6	Merah, Hijau, merah	6
K19,K20,K21	7	Merah, Hijau, Biru	0
K22,K23,K24	8	Merah, Hijau, Biru	0
25,K26,K27	9	Biru, Merah, Biru	8

Tabel 3-4. Tabel Pembacaan Pola warna

Pada Tabel 3-4 dapat disimpulkan bahwa gambar 3-3 memiliki nilai 000746008 yang disusun sesuai urutan yang telah ditentukan. Hasil angka yang dihasilkan sudah melalui proses kombinasi warna sehingga dari 27 warna bisa didapatkan 9 karakter yang baru.

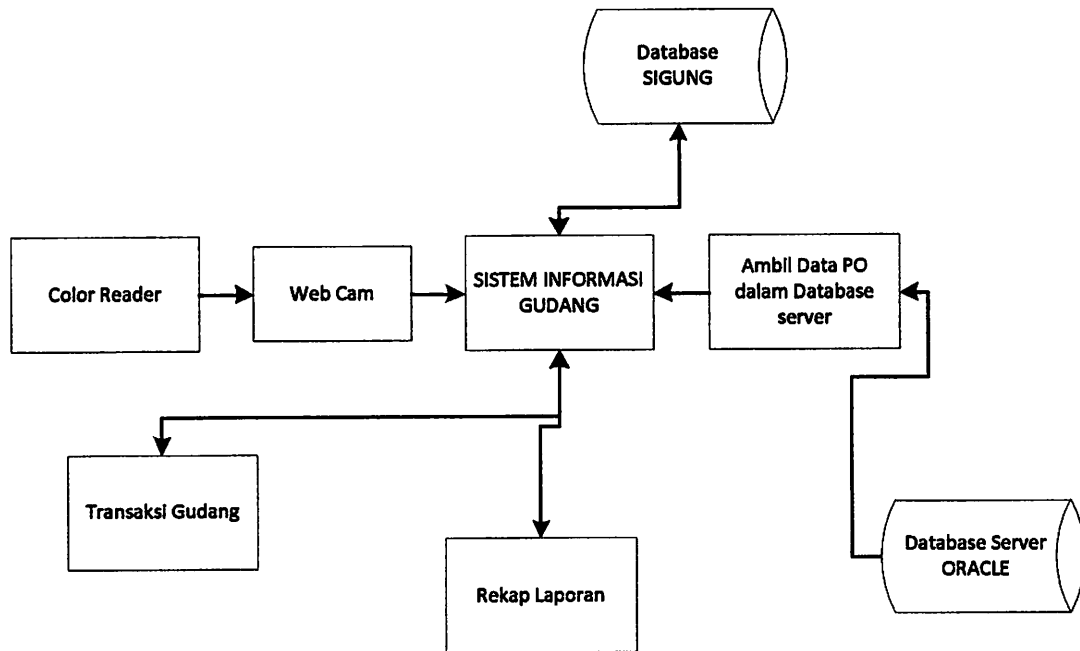
3.2 Perancangan Perangkat Lunak (*Software*)

Perancangan perangkat lunak ini adalah perancangan software yang akan digunakan untuk proses pengolahan citra pembacaan *Color Code*. Data yang dihasilkan dari pengolahan citra ke dalam Sistem informasi yang terintegrasi dengan database sehingga mengeluarkan informasi yang dibutuhkan.

Program yang digunakan untuk membuat program aplikasi ini adalah Borland Delphi7. Alasan pemakaian Delphi7 karena Delphi7 termasuk Pemrograman berorientasi objek disingkat OOP merupakan pemrograman yang berorientasi kan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam kelas-kelas atau objek-objek. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya. Program Delphi7 ini juga dapat dipakai sebagai Pengolahan Citra digital dan dapat mengambil Citra melalui webcam sehingga bisa langsung diolah. Gambar 3-4 adalah gambaran umum sistem yang akan dibuat.

Program yang akan di buat dalam proses capture image akan melakukan capture image sebanyak 5 kali dalam 1 detik. Jika warna kunci yang memiliki pixel yang lebih besar sudah terdeteksi maka akan dilakukan pembacaan pixel yang kecil. Fungsi dari image yang berukuran besar yang ada di atas *Color Code* adalah sebagai penandaan untuk melakukan proses segmentasi dan recognition. Pada gambar 3-4 adalah gambaran umum tentang desain sistem yang akan dibuat.

Pada gambar 3-4 *Color Code* akan dibaca melalui webcam dan akan dikirimkan citra digital ke dalam sistem informasi gudang untuk diolah menjadi sebuah karakter yang mewakili sebuah kode barang dan akan diolah untuk mendapatkan informasi yang sudah tersimpan pada database sigung. Pada sistem informasi ini dapat dilakukan proses transaksi gudang serta rekap laporan. Dan untuk menerima PO diambil dari database oracle yang terhubung oleh server pada kantor pusat.



Gambar 3-4. Diagram Blok Desain Sistem software yang akan dibuat

Spesifikasi perangkat lunak yang dibutuhkan dalam mendukung pembuatan rancang bangun yang digunakan adalah sebagai berikut:

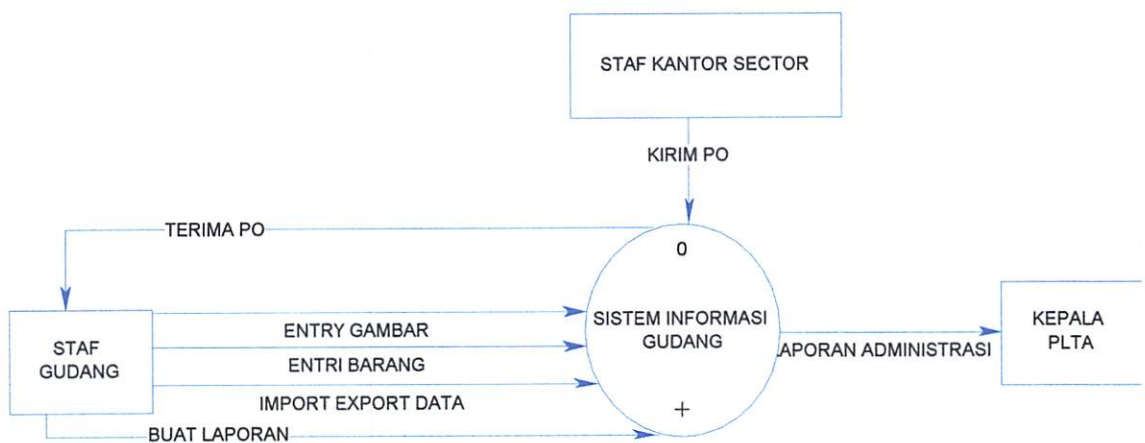
- a. XAMPP – win32-1.7.2. Database yang digunakan
- b. *Borland Delphi 7*, untuk membangun aplikasi.
- c. *Oracle XE* sebagai Database server

Sebelum membuat program penangkapan citra melalui webcam menggunakan Delphi, maka kita harus terlebih dahulu menginstalasi DSpack. DSpack yang kita gunakan adalah DSpack versi 2.3.4. DSpack merupakan sekumpulan komponen dan class untuk menuliskan berbagai aplikasi multimedia menggunakan MS Direct Show dan DirectX technologies. DSpack didesain untuk bekerja dengan DirectX 9 pada sistem operasi Win9X, ME, 2000, dan Windows XP. DSpack didesain juga untuk bekerja dengan Delphi 5, 6, 7 dan CPP Builder 6.

3.2.2 Desain Data Flow Diagram

Dalam perancangan ini akan dibutuhkan juga sistem informasi yang terhubung langsung dengan Color reader yang telah dibuat sehingga data yang akan didapatkan akan langsung diolah pada sistem informasi yang telah dibuat untuk mempercepat proses pencarian informasi. Gambar 3-5 menunjukkan gambaran umum dengan menggunakan blok diagram dari sistem informasi yang akan dibuat.

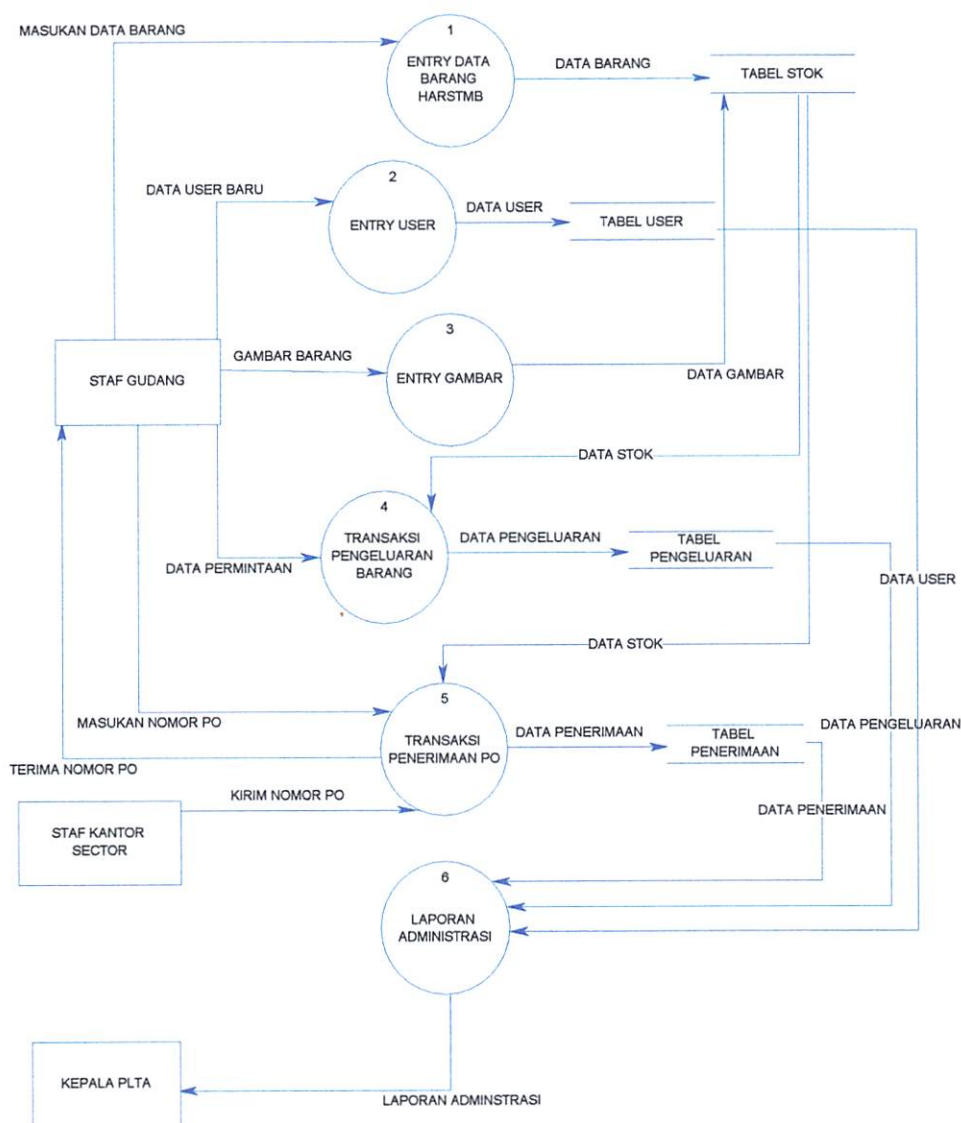
Pada perancangan DFD level 0 ini pada untuk pelaku pada sistem informasi gudang ini ada 3 yaitu kepala PLTA yang bertugas sebagai penerima laporan dari staf gudang dimana progress yang terjadi pada gudang baik itu transaksi penerimaan dan pengeluaran akan di laporkan oleh staf gudang. Pelaku selanjutnya adalah staf kantor sektor dimana yang berhak menerbitkan nomor PO untuk diberikan kepada staf gudang dan di lakukan pendataan sebagai inventory gudang. Dan yang terakhir adalah staf gudang yang melakukan semua transaksi yang ada di gudang baik penerimaan dan pengeluaran barang serta melakukan perubahan data sesuai kebutuhan.



Gambar 3-5. Rancangan DFD Level 0

Pada DFD Level 1 adalah perincian dari proses yang terjadi pada sistem informasi gudang yang berjumlah 7 proses yaitu: entry data Stok semua jenis

HARSTM, Entry User, entry gambar, transaksi pengeluaran barang, transaksi penerimaan barang, dan laporan administrasi yang ditujukan kepada kepala PLTA. Sebagai gambaran tentang DFD level 1 adalah seperti gambar 3-6 di bawah ini.



Gambar 3-6. Rancangan DFD Level 1

3.2.3 Perancangan Database

Pada database MySQL akan di buat beberapa tabel yang akan dijadikan sebagai media penyimpan data gudang. Untuk itu akan dibuat beberapa tabel yang akan disambungkan dengan Delphi yaitu:

- a. seting
- b. t_barang
- c. t_barang_tanpa_barcode
- d. t_history_penerimaan
- e. t_history_pengeluaran
- f. t_penerimaan_barang
- g. t_pengeluaran_barang
- h. t_user

Keterangan dari tabel-tabel tersebut sebagai berikut:

1. Setting

Type status variabel pada database setting ini digunakan untuk menyimpan data untuk pengaturan nama Kepala PLTA, Staf Gudang dan Admin Gudang sehingga pengaturan ini akan muncul ketika akan mencetak laporan dan nama yang bersangkutan bisa di load otomatis dari gambar 3-7 ini.

Field	Jenis
Kepala_PLTA	varchar(100)
Staf_Gudang	varchar(100)
Admin_PLTA	varchar(100)

Gambar 3-7. Type Status Variabel Pada Database Setting

2. t_barang

Pada gambar ini berisi semua type status variabel pada database stok barang yang berjenis HARSTMB, HARSTMT, HARSTMP, HARSTMTR yang membedakan antara setiap jenis barang adalah pada field jenis yang berbeda-beda. Untuk menyimpan gambar pada database digunakan type data BLOB sehingga pada setiap stock code yang menjadi primary key memiliki 1 data gambar yang mewakili setiap data.

Field	Jenis
<u>STOCK_CODE</u>	varchar(12)
ITEM_NAME	varchar(90)
DESC_LINEX1	varchar(35)
DESC_LINEX2	varchar(35)
UNIT_OF_ISSUE	varchar(3)
ESOHINVENT_COST_PR	varchar(25)
LAST_REC_DATE	varchar(10)
CRITICALITY	varchar(3)
STATUS	varchar(3)
ROP	varchar(10)
REORDER_QTY	varchar(7)
SOH	varchar(7)
VOLUME_KARTU	varchar(5)
VOLUME_FISIK	varchar(5)
SELISIH_LEBIH	varchar(10)
SELISIH_KURANG	varchar(10)
BIN_LOKASI_LAMA	varchar(10)
BIN_LOKASI_BARU	varchar(10)
KETERANGAN	varchar(25)
JENIS	varchar(10)
Gambar	longblob

Gambar 3-8. Type Status Variabel Pada Database Stok Barang

3. t_barang_tanpa_barcode

Tabel ini sama dengan tabel stok tetapi perbedaan dengan stok di atas adalah tabel ini tidak memiliki stock code yang menjadi primary key jadi pada tabel ini tidak

menggunakan primary key karena data barang yang disimpan pada tabel ini hanya sementara sebelum mendapatkan stock code dari pejabat yang berwenang.

Field	Jenis
NAMA_BARANG	varchar(90)
BARANG_MILIK	varchar(7)
GROUP_CLASS	varchar(20)
STAT_CODE	varchar(20)
SAT	varchar(5)
SOH	varchar(7)
BANYAKNYA_KARTU	varchar(5)
BANYAKNYA_FISIK	varchar(5)
SELISIH_LEBIH	varchar(10)
SELISIH_KURANG	varchar(10)
BIN_LOKASI	varchar(10)
KETERANGAN	varchar(25)

Gambar 3-9. Type status variabel pada database Setting

Barang Belum Memiliki Barcode

4. t_history_penerimaan

History penerimaan ini digunakan untuk mengetahui kapan saja terjadi penerimaan

Field	Jenis
STOCK_CODE	varchar(10)
NAMA_BARANG	varchar(25)
TANGGAL	date
JUMLAH_TERIMA	varchar(4)
STOCK_KARTU	varchar(4)
STOCK_FISIK	varchar(4)
NOMOR_PO	varchar(10)

Gambar 3-10. Type Status Variabel Pada Database Setting

History Penerimaan

5. t_history_pengeluaran

Gambar dibawah adalah Type Status Variabel Pada Database fungsinya sama dengan t_history_penerimaan tetapi bedanya adalah yang dicatat adalah data transaksi keluar barang.

Field	Jenis
STOCK_CODE	varchar(10)
NAMA_BARANG	varchar(90)
TANGGAL	date
JUMLAH_KELUAR	varchar(4)
STOCK_KARTU	varchar(4)
STOCK_FISIK	varchar(4)
NOMOR_PERMINTAAN	varchar(10)

Gambar 3-11. Type Status Variabel Pada Database History Penerimaan

6. t_penerimaan_barang

Type Status Variabel Pada Database ini mencatat semua data barang yang telah di receipt dari database kantor pusat. Dimana semua transaksi

Field	Jenis
STOCK_CODE	varchar(12)
NAMA_BARANG	varchar(90)
SATUAN	varchar(15)
JUMLAH	varchar(15)
NOMOR_PO_SP	varchar(15)
TANGGAL_PO_SP	date
TANGGAL_LEVERING	date
NOMOR_BAPB	varchar(15)
TANGGAL_DATANG	date
TANGGAL_DITERIMA	date
TANGGAL_DIPAKAI	date
KETERANGAN	varchar(25)

Gambar 3-12. Type Status Variabel Pada Database Penerimaan Barang

7. t_pengeluaran_barang

Type status variabel pada database ini mencatat segala pengeluaran barang yang telah dilakukan sehingga dapat memungkinkan user mengetahui transaksi transaksi barang yang di keluarkan.

Field	Jenis
STOCK_CODE	varchar(12)
NAMA_BARANG	varchar(90)
SATUAN	varchar(15)
JUMLAH	varchar(15)
NOMOR_PERMINTAAN	varchar(15)
NOMOR_WO	varchar(15)
TANGGAL_DIKELUARKAN	date
NOMOR_RETUR	varchar(15)
TANGGAL_RETUR	date
DARI	varchar(25)
KODE	varchar(15)
KETERANGAN	varchar(25)

Gambar 3-13. Tabel Pengeluaran Barang

8. t_user

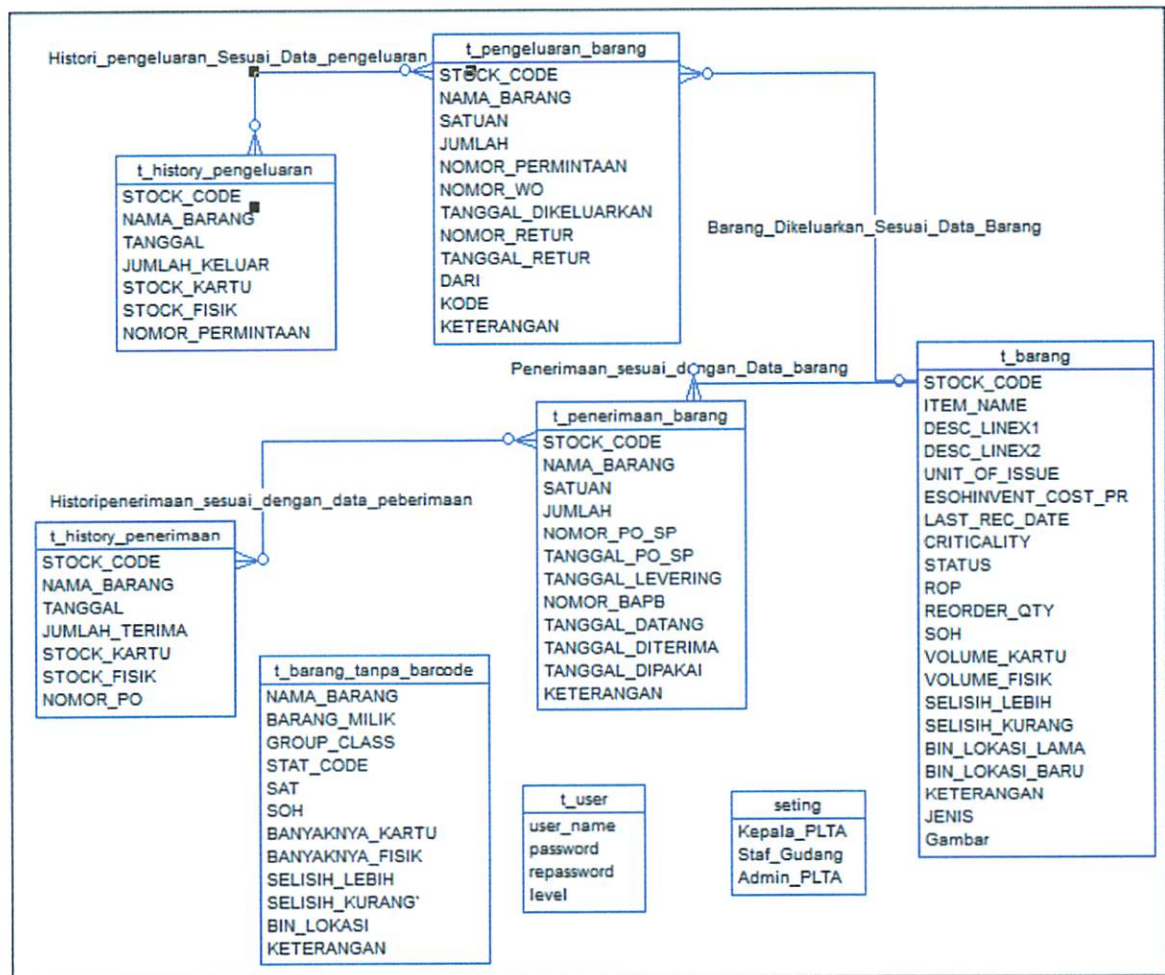
Type status variabel pada database user ini mencatat semua data user yang berhak masuk pada program. Sehingga tabel ini berisi user name dan password semua user yang telah terdaftar. Pendefinisian user level juga dicatat pada tabel ini.

Field	Jenis
user_name	varchar(20)
password	varchar(20)
repassword	varchar(20)
level	varchar(2)

Gambar 3-14. Type Status Variabel Pada Database User

3.2.3 ERD

Pada perancangan database ada beberapa tabel yang memiliki relasi yang saling berhubungan. Tabel yang berhubungan yaitu tabel pengeluaran barang dan tabel barang karena setiap barang yang dikeluarkan harus sesuai dengan data barang yang sudah di data sehingga barang keluar dan masuk dapat di kontrol. Begitu pula antara history pengeluaran dengan tabel penerimaan harus sesuai. Relasi tersebut juga berlaku dengan tabel penerimaan dan history penerimaan. Pada gambar



Gambar 3-15. Desain ERD Database Aplikasi

3.3 Perangkat Keras (*Hardware*)

Dalam perancangan program yang akan dibuat dibutuhkan perangkat keras dan perangkat lunak yang saling berinteraksi. Seperti contoh bahwa kamera yang dipakai harus bias sinkron dengan PC atau laptop yang akan digunakan. Berikut ini adalah daftar spesifikasi hardware dan software yang dibutuhkan:

Spesifikasi perangkat keras yang dibutuhkan dalam rancang bangun yang di gunakan adalah sebagai berikut:

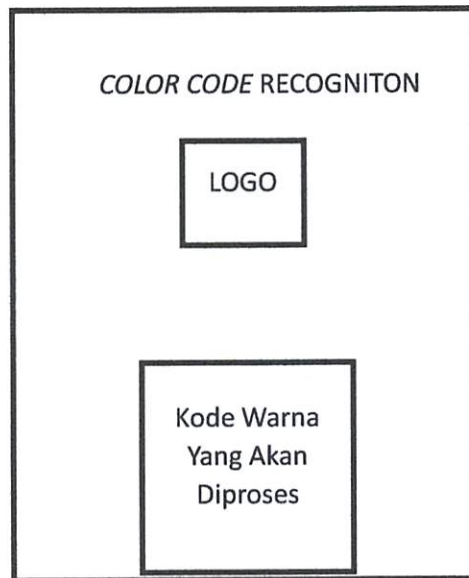
- a. Camera Webcam
- b. *Color Code*
- c. Processor Intel® Core™ i3 CPU M380 @2,53GHz 2,53GHz
- d. RAM 4 GB
- e. Harddisk 320 GB
- f. Sistem operasi Microsoft Windows 7 Profesional

3.3.1 Desain *Color Code*

Color Code yang akan digunakan terbuat dari mika. Untuk background yang dipakai pada desain ini adalah hitam dan posisi yang digunakan adalah portrait sehingga berdiri untuk posisi desain *Color Code*. Kombinasi warna yang akan dibaca ada pada bagian bawah *Color Code*. Desain belakang *Color Code* hanya berisi logo dari *Color Code*.

Gambar 3-15 adalah rancangan desain *Color Code* dimana warna RGB yang akan diproses untuk dijadikan sebuah karakter. Letak *Color Code* yang akan diproses berada pada bagian bawah. Untuk kartu *Color Code* nya dibuat secara portrait. Bentuk kombinasi warna berbentuk persegi sehingga dalam melakukan segmentasi setiap wilayah lebih mudah karena memiliki luas wilayah yang sama. Di atas kombinasi warna diberi logo untuk mengetahui bahwa kartu ini adalah *Color Code*.

Desain kartu ini disamakan dengan ukuran ID card seperti KTP, SIM, ATM bertujuan supaya dalam pengembangan selanjutnya diharapkan dapat diimplementasikan pada ID card yang lain.



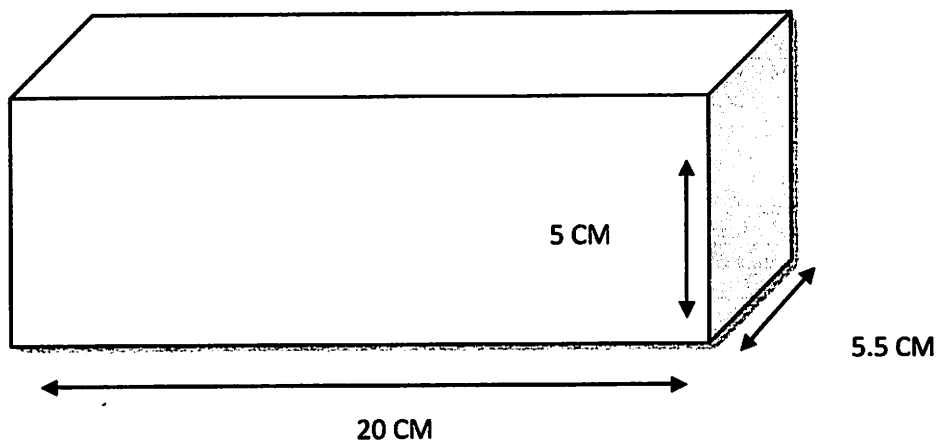
Gambar 3-16. Tampilan Depan *Color Code*



Gambar 3-17. Tampilan Depan Dan Belakang Kartu *Color Code*

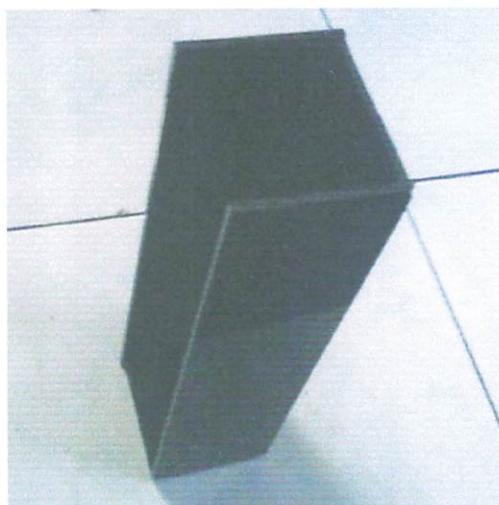
3.3.2 Perancangan Box Pembacaan Kode Warna

Proses pertama yang dilakukan dalam pembacaan kode warna adalah menempatkan kartu pada sebuah box yang sudah disediakan sehingga pembacaan akan lebih akurat. Pada box ini sudah ditentukan berapa jarak antara kamera dengan kartu kode warna yang akan dibaca. Gambar 3-17 adalah desain box untuk membaca *Color Code*.



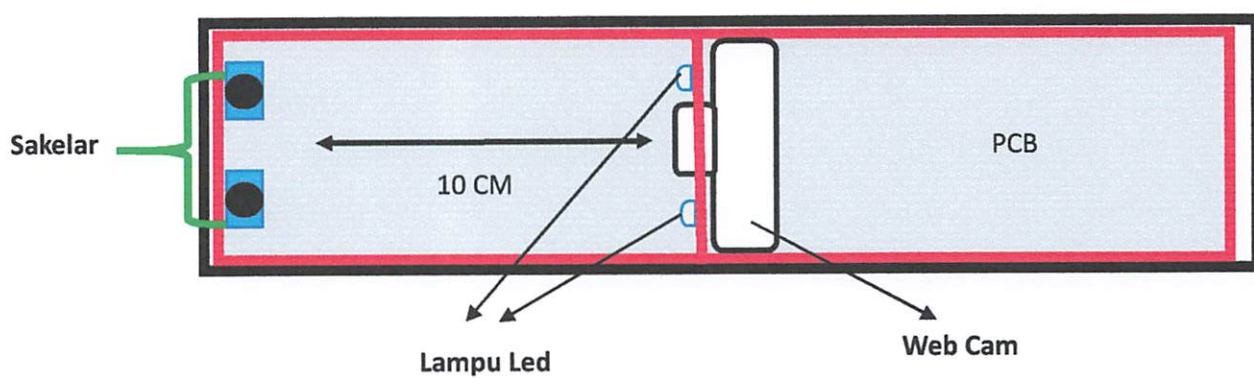
Gambar 3-18. Tampilan Belakang *Color Code*

Pada gambar 3-17 adalah desain box yang berbentuk balok yang memiliki panjang, lebar dan tinggi yang berbeda-beda. Box ini memiliki tinggi 5 cm, panjang 20 cm dan lebar 5.5 cm. untuk lebar dari box ini disesuaikan dengan lebar *Color Code*.



Gambar 3-19. Tampilan Box Pembaca Kartu

Gambar 3-18 tampilan box yang telah jadi yang terbuat dari mika. Warna yang dipilih adalah warna hitam karena mencegah masuknya cahaya yang akan mengganggu proses pembacaan *Color Code* karena dengan warna hitam kondisi di dalam box akan gelap dan satu-satunya sumber cahaya ada pada lampu led



Gambar 3-20. Desain Box Pembaca Kartu

Pada gambar 3-19 menunjukkan bahwa di dalam box terdapat sebuah webcam untuk menangkap citra analog sehingga bias dikenali oleh computer. Selain webcam box ini juga mempunyai sebuah switch dan USB terminal untuk menggabungkan antara lampu led yang ada pada box dengan webcam. Jadi ketika kartu menempel pada dasar box maka otomatis akan menekan switch sehingga akan menyalakan lampu led dan cahaya akan mengarah kepada kartu kode warna. Lampu led ini berfungsi untuk memberi cahaya yang tetap sehingga dalam pembacaan tidak terganggu pada intensitas cahaya yang ada pada sekelilingnya.



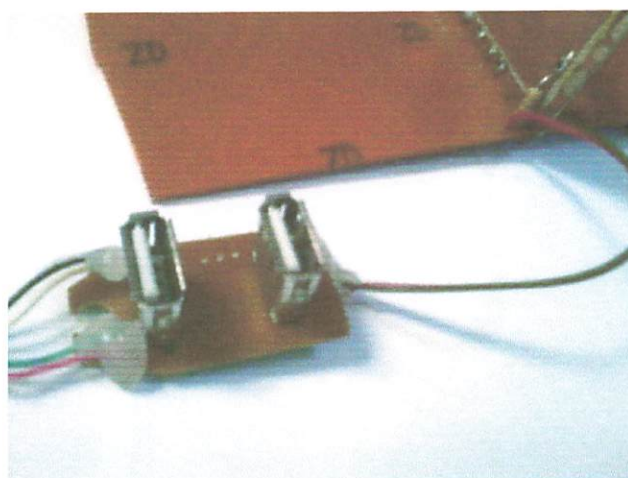
Gambar 3-21. Tampilan Box Pembaca Kartu yang Sudah Jadi

Gambar 3-20 adalah hasil jadi dari desain box yang dibuat dari mika tebal dengan ketebalan 1 cm. pada bagian belakang webcam terdapat kabel webcam dan USB hub yang menghubungkan antara camera webcam dengan lampu led.



Gambar 3-22. Desain PCB Sistem Pencahayaan Dengan Lampu Led Di Dalam Box

Lampu led yang dipasang mengambil catu daya lewat USB sehingga tidak memerlukan sumber tenaga tambahan karena lampu led menyala ketika proses pembacaan kode warna saja sehingga menambah efisiensi tenaga yang dikeluarkan. Melakukan pembacaan tanpa menggunakan box bias dilakukan akan tetapi lebih sulit karena harus menempatkan pada titik yang telah ditentukan karena itu diperlukan box untuk membantu pengambilan gambar.



Gambar 3-23. USB Hub Digunakan Untuk Menyambungkan Antara Web Cam Dengan Lampu Led

Pada USB hub yang diambil oleh lampu led hanya jalur dari tegangan USB saja sedangkan untuk jalur data USB tidak dipakai. Karena pada USB ada 4 jalur yaitu 2 jalur untuk tegangan dan 2 jalur untuk lalu lintas data.

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Color Recognition

Pada bagian implementasi ini akan dilakukan proses pengambilan gambar dari kode warna yang sudah didesain pada sebuah kartu yang telah didesain. Kemudian akan dilakukan proses recognition yang akan menghasilkan sebuah kumpulan kode barang yang akan menampilkan informasi dari kode barang tersebut. Untuk menampilkan data atau informasi dari hasil pembacaan *Color Code* perlu disambungkan ke dalam Sistem Informasi Gudang untuk mengolah data hasil pembacaan.

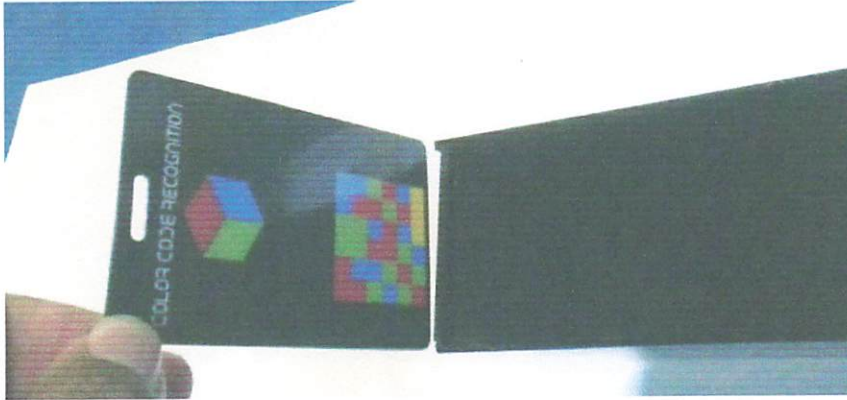
4.1.1 Pembacaan Kode Warna

Hasil dari pembacaan kode adalah sebuah bilangan desimal yang mewakili sebuah kode barang. Dari kode barang hasil dari pembacaan akan dijadikan menjadi sebuah kata kunci untuk mencari informasi data barang. Dalam proses capture kode warna hal yang pertama di proses adalah pemilihan jenis kamera yang aktif. Dalam combo box akan ditampilkan semua hardware kamera yang dideteksi oleh komputer. Selanjutnya ketika dipilih salah satu kamera yang sudah terinstal maka kamera yang dipilih akan aktif. Karena menggunakan komponen DSpack maka gambar yang ditangkap oleh kamera akan ditampilkan pada komponen Video Windows. Komponen ini adalah salah satu komponen dari DSpack yang berfungsi meneruskan gambar yang ditangkap oleh kamera supaya bisa ditampilkan pada layout Delphi. Pada gambar 4-1 ditampilkan form yang digunakan untuk membaca kombinasi warna.

Gambar 4-1. Form Pembacaan *Color Code*

Setelah memilih kamera yang aktif langkah selanjutnya adalah memasukan *Color Code* pada box reader yang telah dibuat. Tunggu sampai suara bel berbunyi yang mengisyaratkan bahwa pembacaan *Color Code* telah berhasil. Dan kartu sudah bisa diambil dari box reader karena data kombinasi yang diinginkan sudah terbaca oleh system.

Ketika memasukan *Color Code* posisi kartu harus sejajar seperti pada gambar 4-2 supaya 2 switch yang ada pada dasar box tertekan dan memulai proses pembacaan. Fungsi dari kedua switch tersebut untuk menyalakan lampu led untuk melakukan proses pencahayaan dalam box. Ketika terdengar suara alarm maka itu menandakan bahwa proses pembacaan *Color Code* sudah selesai dan Kartu *Color Code* boleh diambil. Gambar 4-3 adalah tampilan *color code* ketika menempel pada switch dimana color code tenggelamkan ke dalam box reader sedalam $\frac{3}{4}$ bagian.



Gambar 4-2. Memasukan *Color Code* Ke Dalam Box Reader



Gambar 4-3. Memasukan *Color Code* Ke Dalam Box Reader

4.2 Integrasi Sistem Informasi Gudang

Setelah mendapatkan kata kunci yang terdapat pada kode warna maka akan di masukan ke dalam data base dan akan ditampilkan informasi mengenai barang yang di maksudkan.

4.2.1 Form Login

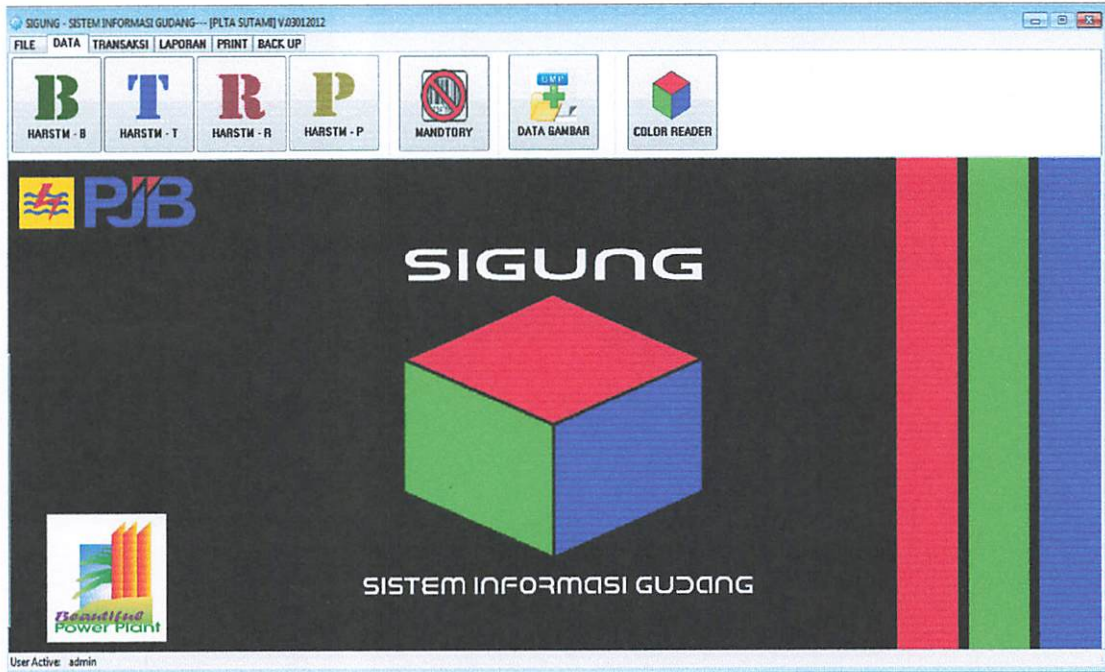
Gambar 4-3 adalah form yang akan di tampilkan pertama kali adalah form login. Form validasi ini adalah sebagai penyeleksi user yang masuk karena tidak semua user berhak merubah data. User disini akan di bagi menjadi 2 level yaitu level 1 dan 2. Yang membedakan antara user level 1 dan 2 adalah pada level 2 bisa membuka semua form dan mengakses tombol simpan, edit, hapus dan cari, sedangkan pada level 1 yakni level terendah hanya bisa melihat beberapa form dan tidak mempunyai hak untuk melakukan simpan, edit, hapus, cari hanya bisa melihat beberapa laporan saja.



Gambar 4-3. Form Login Untuk Validasi Hak Akses

4.2.2 Menu Utama

Dalam rancangan awal menu akan tampak seperti gambar 4-4 terlihat beberapa menu utama dimana setiap menu akan berhubungan dalam mengolah database. Pilihan untuk melakukan pembacaan warna ada pada menu Data.



Gambar 4-4. Menu Utama Program

4.2.3 Form Data Barang

Form barang ini adalah form yang sering di lihat karena pada form ini akan dapat dilihat karena kode B berarti Baru Bisa Dipakai. Pada gambar 4-5 form ini bisa melakukan simpan, edit, hapus, cari hanya untuk user level 2. Pada form ini juga bisa melihat gambar planogram dari rak gudang serta posisi dari barang yang dipilih dengan cara mengklik tombol lokasi. Pada pojok sebelah kanan terlihat juga gambar dari sebuah barang yang dipilih. Program ini di beri tambahan gambar karena ada beberapa barang yang masih dalam kardus sehingga tak perlu membuka-buka kardus untuk mengetahui bentuk barangnya. Selain itu juga terdapat ratusan barang sehingga tidak bisa menghafal kan bentuk barangnya satu per satu.

Volume 10, Number 1, Spring 1975
 Editor: J. M. J. van der Kamp
 Editor: J. M. J. van der Kamp
 Editor: J. M. J. van der Kamp

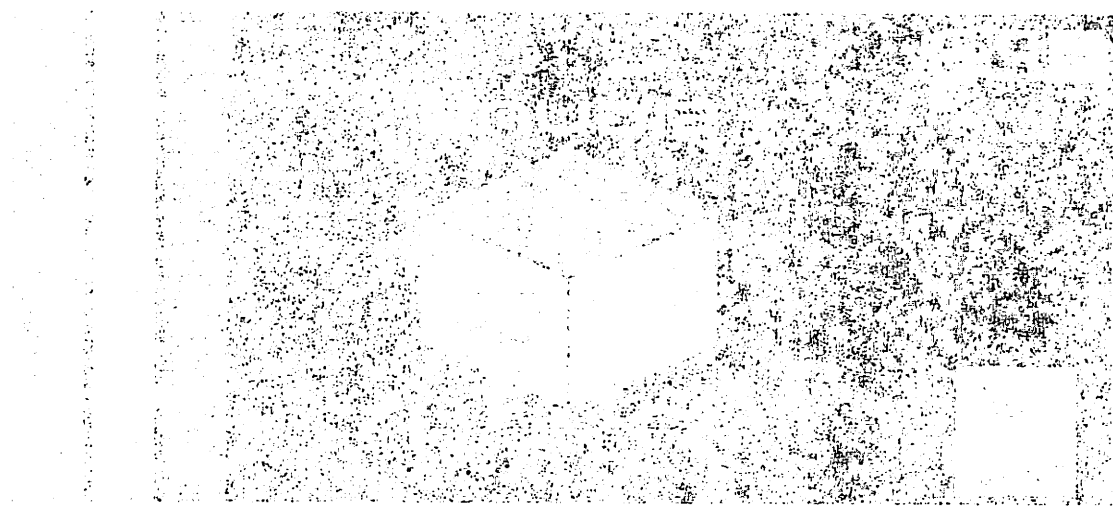


Figure 1. The experimental setup.

2.1.2. Stimulus Material

The stimuli were 16 short stories, each 100 words long, describing a situation in which a person was faced with a choice between two actions. The stories were written by a group of 16 students, 8 male and 8 female, who were given a list of 16 situations and asked to write a short story for each. The stories were then read aloud to a group of 16 students, 8 male and 8 female, who were asked to rate the stories on a scale of 1 to 5, where 1 was "very boring" and 5 was "very interesting". The stories were then ranked according to their average rating, and the top 16 stories were selected for use in the experiment. The stories were then read aloud to a group of 16 students, 8 male and 8 female, who were asked to rate the stories on a scale of 1 to 5, where 1 was "very boring" and 5 was "very interesting". The stories were then ranked according to their average rating, and the top 16 stories were selected for use in the experiment.

Gambar 4-5. Form pengolah data barang

4.2.4 Form Denah Lokasi

Form ini adalah salah satu fitur yang diunggulkan dimana barang yang di cari digambarkan pada sebuah peta planogram gudang. Disini barang yang dipilih akan di tandai dengan warna titik yang berkedip-kedip titik yang sebelumnya warnanya hijau akan berubah menjadi merah.

Gambar 4-6. Form pengolah data barang

4.2.4 Form Penerimaan PO

Form ini digunakan untuk melihat laporan penerimaan yang terjadi. Pada form ini dapat melihat penerimaan barang berdasarkan Nomor PO, Nama Barang, Stock Code kemudian tinggal di klik preview maka akan tampilan semua data penerimaan sesuai dengan jenis yang dipilih.

Gambar 4-7. Form pengolah PO

4.2.5 Form Pengeluaran Barang

Form pengeluaran barang ini berhubungan dengan transaksi pengeluaran barang. Untuk melakukan pengeluaran barang pertama kali harus memilih barang yang akan dikeluarkan dengan cara mengklik 2 kali pada nama barang atau tekan F11 kemudian akan muncul form data barang kemudian pilih salah satu barang yang ada pada stok laku klik OK atau klik 2 kali pada barang yang dipilih.

Gambar 4-8. Form penerimaan barang

4.2.6 Cetak Laporan

Untuk form ini di gunakan untuk mencetak laporan mulai dari tanggal awal yang diinginkan sampai tanggal akhir yang diinginkan. Disini akan menampilkan transaksi penerimaan barang sesuai range tanggal yang diinginkan. Setelah itu klik preview maka akan muncul semua transaksi yang diinginkan. Setelah muncul data yang sudah dicari kemudian pilih cetak sampai muncul tampilan print preview dan tinggal pilih menu print untuk mencetak laporan yang diinginkan

STOCK_CODE	NAMA BARANG	TANGGAL	JUMLAH	NOMOR PO	TANGGAL PO	TANGGAL LEVEL
0003290	WALL AND FILLING	EA	15	884003	18/04/2011	02/05/2011
0003676	BRUSH, PAINT	EA	6	884003	18/04/2011	02/05/2011
0003987	BRUSH	EA	4	884003	18/04/2011	02/05/2011
00012469	DETERGENT, LAUNDRY	KG	4	884003	18/04/2011	02/05/2011
00012823	BRUSH, SCRUB	EA	6	884003	18/04/2011	02/05/2011
00012297	PENETRATING OIL	CAR	6	884003	18/04/2011	02/05/2011
00047909	BAG, TEXTILE	EA	6	884003	18/04/2011	02/05/2011
00062099	BAG, WIPING	KG	12	884003	18/04/2011	02/05/2011
00062071	TURNER, PAINT PRODUCTS	L	6	884003	18/04/2011	02/05/2011
00022546	TAPE, ANTISTATIC	ROLL	12	884003	18/04/2011	02/05/2011
00024346	PAPER, ABRASIVE	SHET	15	884003	18/04/2011	02/05/2011
00047146	GLUERS	PAIR	15	884003	18/04/2011	02/05/2011
00074008	TAPE, INSULATION, ELECTRICAL	ROLL	6	884003	18/04/2011	02/05/2011
00012103	BALLAST LAMP	EA	2	884007	11/02/2011	25/02/2011
00012303	LIGHT INDICATOR	EA	2	884007	11/02/2011	25/02/2011
00012823	BRUSH SCRUB	EA	2	884004	26/12/1999	26/12/1999

Gambar 4-9. Form laporan penerimaan barang

Print Preview

PERENCANAAN DAN PROSES PELAKSANA TRANSAKSI PERGUDANGAN

HALAMAN

NO. 000
TGL. BERLAKU
REVISI
HALAMAN

UP BRASTAS

PENERIMAAN (IN)

NO	STOCK CODE	NAMA BARANG	SAT	JMLH	NOMOR POSP	TANGGAL		NOMOR BAPB	TANGGAL			KETERANGAN
						POSP	LEVERING		DATANG	DITERIMA	DIPAKAI	
0001208	MINICH SOCKET	SET	3	884803	28082011	05092011	ACH77	28042012	28042012	28192012	PERCOBAAN	
0001209	TERMINAL BLOC	EA	3	884803	28082011	05092011	ACH77	28042012	28042012	28192012	PERCOBAAN	
22	PLA	EA	4	884803	28082011	05092011	ACH77	28042012	28042012	28192012	PERCOBAAN	
0001210	RELAY, THERMAL	EA	3	884803	28082011	05092011	ACH77	28042012	28042012	28192012	PERCOBAAN	
0001211	RELAY, ELECTROMAGNETIC	EA	4	884803	28082011	05092011	ACH77	28042012	28042012	28192012	PERCOBAAN	
0001212	BALLAST LAMP	EA	1	884803	28082011	05092011	ACH77	28042012	28042012	28192012	PERCOBAAN	
0001213	BALLAST LAMP	EA	1	884803	28082011	05092011	ACH77	28042012	28042012	28192012	PERCOBAAN	
0001214	BALLAST LAMP	EA	20	884803	28082011	05092011	ACH77	28042012	28042012	28192012	PERCOBAAN	
0001215	BALLAST LAMP	EA	20	884803	28082011	05092011	ACH77	28042012	28042012	28192012	PERCOBAAN	
0001216	LIHTE INDICATOR	EA	10	884803	28082011	05092011	ACH77	28042012	28042012	28192012	PERCOBAAN	
0001217	LIHTE INDICATOR	EA	4	884803	28082011	05092011	ACH77	28042012	28042012	28192012	PERCOBAAN	
0001218	LIHTE INDICATOR	EA	3	884803	28082011	05092011	ACH77	28042012	28042012	28192012	PERCOBAAN	
0001219	LIHTE INDICATOR	EA	3	884803	28082011	05092011	ACH77	28042012	28042012	28192012	PERCOBAAN	
0001220	LIHTE INDICATOR	EA	2	884803	28082011	05092011	ACH77	28042012	28042012	28192012	PERCOBAAN	

50% Page 1 of 9

Gambar 4-10. Hasil cetak laporan penerimaan

Pada form ini juga sama seperti laporan penerimaan pengguna bisa mendapatkan data pengeluaran sesuai dengan range tanggal yang diinginkan kemudian dicetak ke dalam hard copy.

SUTAM-----SIGUNG-----[LAPORAN TRANSAKSI PENERIMAAN]

LAPORAN PENGELUARAN BARANG

TENTUKAN TANGGAL TRANSAKSI

01/01/2012 5/0 14/01/2013

Preview Cetak

STOCK CODE	NAMA BARANG	SATUAN	JUMLAH	NOMOR PERMINTAAN	NOMOR WD	TANGGAL DIF
000013169	BALLAST, LAMP	EA	5	5	5	15/03/2012
000320130	TERMINAL BLOCK	EA	3	457	YU78	17/04/2012

Gambar 4-11. Form Laporan Pengeluaran

												No. DOC : TGL BERLAKU : REV ISI : PALAMAN :	
PERENCANAAN DAN PROSES PELAKSANA TRANSAKSI PERGUDANGAN													
PENGELUARAN BARANG (OUT) DAN RETUR													
NO	STOCK CODE	NAMA BARANG	SAT	JMLH	NOMOR PERMINTAAN	NOMOR WO	TANGGAL DIKELUARKAN	NOMOR RETUR	TANGGAL RETUR	DARI	KODE	KETERANGAN	
	000013109	BALLA ST. LAMP	EA	5	5		19/03/2012	5	0:00:00	55	5		

Mengetahui,
 KEPALA PLTA SUTAMI

Disetujui,
 ADMIN PLTA SUTAMI

Dibuat Oleh,
 STAF GUDANG PLTA SUTAMI

KUSNADI

BAWON

BRAHMANTYO

Gambar 4-12. Hasil Cetak Laporan Penerimaan

4.2.7 History transaksi

Pada gambar 4-13 menunjukkan Form history ini adalah rangkuman dari penerimaan dan pengeluaran dengan memasukan jangka waktu transaksi. Langkah pertama masukan barang yang akan di ketahui history nya kemudian masukan laporan transaksi dari tanggal awal sampai tanggal akhir kemudian klik preview. setelah di klik preview maka barang yang di tunjukkan akan diketahui umur barang tersebut di gudang sudah berapa lama kemudian dari tanggal awal dan akhir tersebut akan di ketahui ada transaksi apa saja.

4.2.8 Grafik Report

Form ini adalah laporan tentang perkembangan stock dengan transaksi penerimaan dan pengeluaran tinggal klik grafik kemudian akan muncul grafik perbulan seperti yang ditunjukkan pada gambar 4-14. Disini dapat diketahui antara stok dan transaksi apakah seimbang sehingga bisa diketahui kapan waktunya menambah stock barang. Form ini ada pada menu laporan. Laporan grafik lebih disukai untuk melihat kondisi gudang apakah overload ataukah sebaliknya. Disini ada detail yang berfungsi untuk mengetahui grafik yang detail setiap bulannya.

SUTAMI-----SIGUNG-----[HISTORY PENERIMAAN TIAP ITEM]

HISTORY TRANSAKSI BARANG

STOCK CODE

NAMA BARANG

DARI TANGGAL SAMPAI

UMUR BARANG DI GUDANG

0

HARI

 PREVIEW

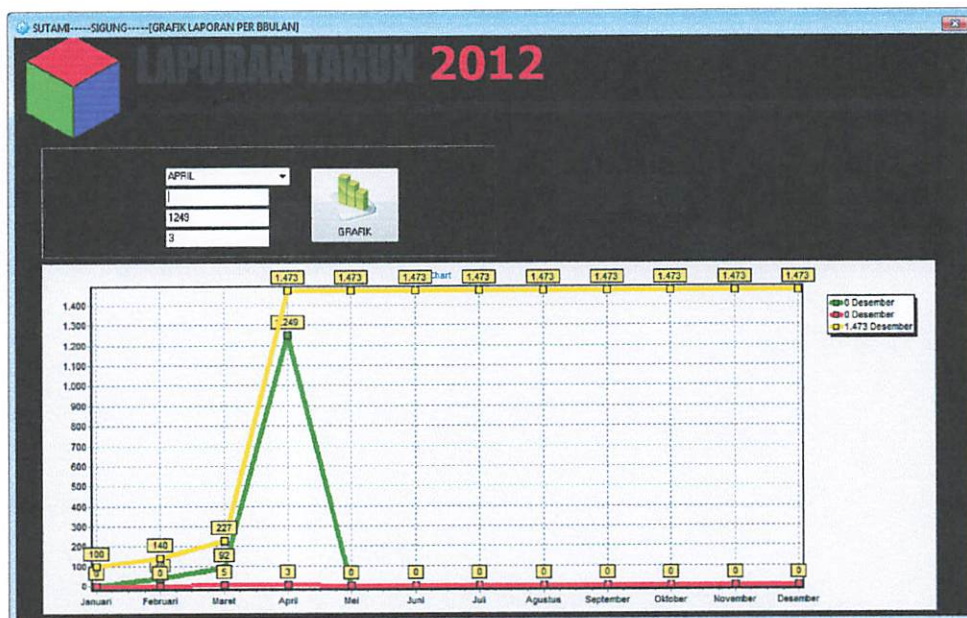
TRANSAKSI PENERIMAAN

STOCK_CODE	NAMA_BARANG	TANGGAL	JUMLAH_TERIMA	STOCK_KARTU	STOCK_FISIK	NOMOR_PO
000002741	PAPER, ABRASIVE	21/02/201:5	11	11		dd3
000013383	LIGHT, INDICATOR	21/03/201:3	16	16		
000013383	LIGHT, INDICATOR	21/03/201:3	16	16		
000002741	PAPER, ABRASIVE	22/03/201:2	16	16		dd4
000013169	BALLAST, LAMP	14/03/201:5	24	24		p0
000013169	BALLAST, LAMP	09/03/201:3	29	29		dd3

TRANSAKSI PENGELUARAN

STOCK_CODE	NAMA_BARANG	TANGGAL	JUMLAH_KELUAR	STOCK_KARTU	STOCK_FISIK	NOMOR_PERMINTA
000013383	LIGHT, INDICATOR	30/12/189:5	17	17		bc18
000328245	SWITCH, THERMOSTATIC	30/12/189:5	1	1		tt66
000318964	BEARING, PLAIN, SPHERICAL	30/12/189:6	1	1		n2
000013383	LIGHT, INDICATOR	30/12/189:3	6	6		sd
000328245	SWITCH, THERMOSTATIC	30/12/189:3	-4	-4		1212

Gambar 4-13. History Transaksi



Gambar 4-14. Laporan Grafik Stok Barang

4.2.9 Form Import Data

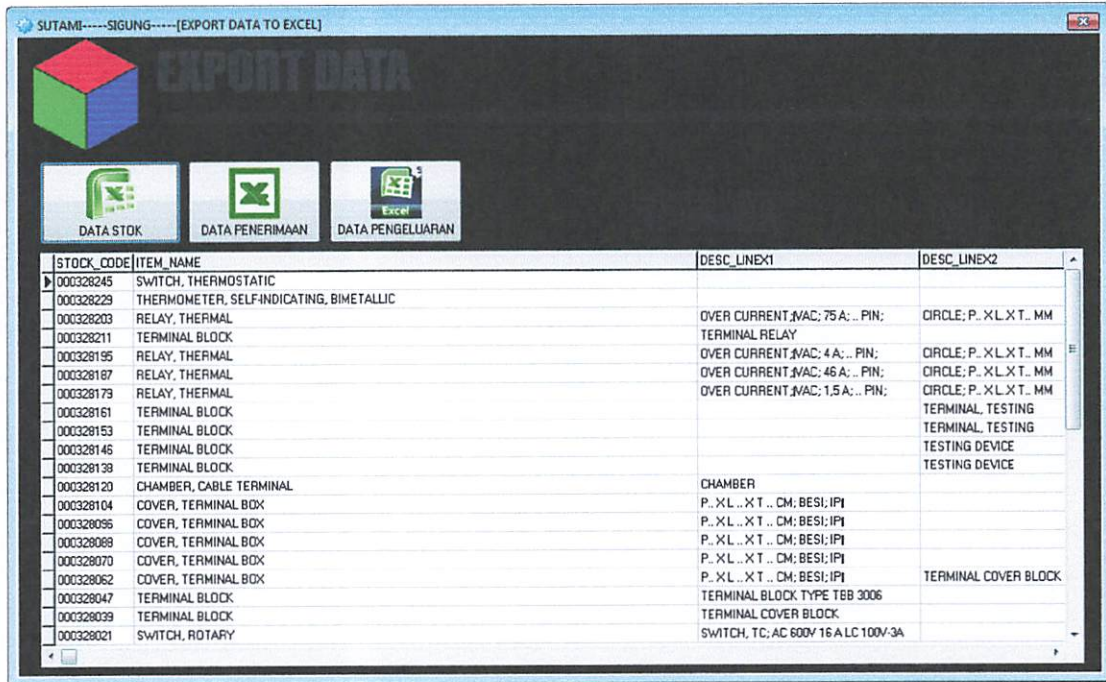
Form ini berguna untuk melakukan import data dari excel menjadi DBMS. Form ini berguna untuk melakukan perpindahan dari sistem yang masih menggunakan excel di ubah menjadi DBMS. Untuk mengimport ke dalam DBMS format pada excel harus sama seperti tabel dalam database. Langkah selanjutnya tentukan file excel yang ingin di masukan ke dalam kemudian masukan nama database dan nama tabel tujuan yang berada dalam database. Kemudian import maka data yang ada pada excel akan langsung masuk ke dalam database tanpa mengetik satu persatu data yang ada pada excel.

Gambar 4-15. Import Database Dari File Excel

4.2.10 Form Import Data

Form ini berfungsi sebagai pemindahan data dari tabel pada DBMS ke tabel excel kemudian bisa di simpan lagi dalam bentuk excel. Form ini sederhana hanya

tinggal menyimpan kembali dalam bentuk excel lagi. Import data excel ini dibutuhkan karena ada beberapa divisi pada perusahaan yang masih mengolah secara manual data base sehingga dibutuhkan import data excel yang bisa digunakan pada waktu yang diinginkan. Gambar 4-16 adalah tampilan form export.



Gambar 4-16. Export Database Menjadi File Excel

4.3 Pengujian

Salah satu teknik yang digunakan pada pengujian Color Recognition adalah *Black-Box Testing*. *Black-Box Testing* adalah suatu teknik pengujian yang mengamati proses masukan dan keluaran dari sistem perangkat lunak tanpa memperhatikan apa yang terjadi di dalam sistem. Salah satu cara yang terdapat pada *Black-Box Testing* adalah dengan membuat tabel-tabel yang di dalamnya berisi skenario, *output* yang diharapkan dan validasi. Hal tersebut dilakukan untuk menguji kesesuaian antara desain dengan implementasi.

Proses pengujian adalah proses yang penting karena dalam proses ini akan diketahui apakah program yang dibuat sudah sesuai dengan apa yang diharapkan. Dalam pengujian ini dilakukan beberapa langkah yang berbeda untuk memastikan bahwa program berjalan sesuai keinginan. Sebelumnya telah di sediakan 5 sampel kartu *Color Code*.



Gambar 4-17. Tampilan Sampel Yang Akan Di Uji

Pengujian pertama dilakukan dengan cara pembacaan pada beberapa sampel kode warna yang telah dibuat. Dalam pengujian ini dilakukan 10 kali pengujian kartu dengan kode yang berbeda-beda dan dilakukan dalam beberapa kali perulangan. Dengan pengujian ini bisa diperoleh berapa tingkat akurasi pembacaan kode warna.

Hasil pengujian akan ditampilkan pada sebuah tabel untuk mengetahui hasilnya seperti gambar di bawah ini:

Sample	Nilai Kode Warna	Hasil Pembacaan	Presentase keberhasilan
1	000746008	000746008	100%
2	000791913	000791913	100%
3	000805481	000805481	100%
4	123456780	123456780	100%
5	000327304	000327304	100%
Persentase Keberhasilan			100%

Tabel 4-1. Hasil Pengujian Tahap 1

Pengujian kedua dengan menggunakan sampel acak dan dilakukan dalam beberapa kali pembacaan. Hal ini bertujuan untuk mengetahui akurasi jika menggunakan data yang sama dan acak secara berkala apakah akan mempengaruhi tingkat akurasi atau tidak. Hal ini akan dilakukan sebanyak 20 kali pada pengujian ini akan mendapatkan beberapa sampel yang memiliki nilai Sama.

Sample	Nilai Kode Warna	Hasil Pembacaan	Persentase keberhasilan
1	000327304	000327304	100%
2	000746008	000746008	100%
3	000327304	000327304	100%
4	000791913	000791913	100%

5	000805481	000805481	100%
6	123456780	123456780	100%
7	000746008	000746008	100%
8	000327304	000327304	100%
9	000791913	000791913	100%
10	000746008	000746008	100%
11	123456780	123456780	100%
12	000746008	000746008	100%
13	000791913	000791913	100%
14	123456780	123456780	100%
15	000805481	000806481	98,875%
16	000746008	000746008	100%
17	000805481	000805481	100%
18	000791913	000791913	100%
19	000746008	000746008	100%
20	000805481	000805481	100%
Persentase Keberhasilan			99,944%

Tabel 4-2. Hasil Pengujian Tahap 2

Dari hasil pengujian pada Tabel 4-1. dengan menggunakan sampel yang berturut selama 5 kali menghasilkan keberhasilan pembacaan 100% hal ini menandakan bahwa pembacaan *Color Code* berhasil 100% pada sampel yang digunakan. Pada pengujian kedua pada Tabel 4-7. dari 5 sampel yang akan di uji secara

acak sebanyak 20 kali, dalam pengujian kedua ini terdapat salah satu kode yang salah membaca kode yaitu pada *Color Code* 000805481 salah pada angka 5 menjadi angka 6 sehingga hasil pembacaanya menjadi 000806481. Tapi ketika dilakukan pengujian lagi dengan memasukan *Color Code* yang terjadi kesalahan hasilnya sudah sesuai dengan yang diharapkan. Kesalahan mungkin terjadi karena cahaya yang masuk dan faktor posisi kartu saat dimasukkan pada color reader.

Dari hasil pengujian yang didapatkan, bisa diketahui bahwa selama proses pengujian pembacaan *Color Code* dari sampel yang telah disediakan, semua hasil kondisi yang dipersyaratkan telah sesuai dengan desain yang diharapkan, walaupun ada beberapa kesalahan dan bisa langsung diperbaiki.

4.4 Kelebihan dan Kekurangan

Pada sistem yang dibuat ini memiliki kelebihan dan kekurangan dibandingkan dengan barcode batang yang biasa dipakai. Kelebihan dari *Color Code* ini kombinasi warna masih belum dikenal secara luas sehingga bisa menjaga suatu kerahasiaan kode yang disimpan dalam *Color Code*. Berbeda dengan kode batang yang biasa ditemui yang bisa diketahui secara mudah. Kekurangan dari *Color Code* ini adalah bergantung dengan jenis warna saat pencetakan *Color Code* karena tidak bisa menggunakan tinta hitam dan putih.

Color Code ini juga memiliki bentuk yang menarik karena terdiri dari campuran warna. Kombinasi ini juga masih bisa ditambahkan lagi dengan karakter yang berbeda lagi karena kombinasi yang dihasilkan dari gabungan 3 warna ini masih jutaan karakter kombinasi.

BAB V

PENUTUP

5.1. Kesimpulan

1. Dengan Menggunakan Metode Dynamic Color Thresholding dapat melakukan pembacaan warna sehingga menjadi sebuah informasi yang dapat terintegrasi dengan database.
2. Color Code ini bisa menjadi sebuah jenis code baru dengan memanfaatkan kombinasi warna sehingga dapat mempercepat pengenalan kode barang sehingga lebih efisien.
3. Setiap warna dapat mewakili sebuah kode sehingga dengan warna dapat dibuat sebuah kata kunci khusus untuk menghindari redundansi data.
4. Pendeteksian warna ini dinyatakan 100% berhasil dan dalam pengujian dengan sampel acak sebanyak 20 kali tingkat keberhasilannya 179:180 atau secara persentase dapat dinyatakan sebesar 98,875% .
5. Dalam melakukan deteksi warna intensitas cahaya disekitar lingkungan pembacaan akan mempengaruhi dari nilai pembacaan intensitas nilai warna dasar RGB yang akan di inisialisasi
6. Kombinasi yang tercipta dari 3 warna dasar RGB bisa tercipta jutaan karakter baru, jika ditambah beberapa warna lagi maka bisa di manfaat kan untuk mewakili data yang sangat banyak.
7. *Color Code* ini bisa diimplementasikan bukan hanya untuk mengenali karakter numeric saja, tetapi dengan penambahan kombinasi semua karakter maupun motion bisa diwakili oleh *Color Code* ini.

5.2. Saran

1. Penambahan pendeteksian pola dengan menggunakan webcam akan menambah efektivitas dalam melakukan pembacaan *Color Code* ini karena jika menggunakan pendeteksian pola maka tidak perlu menggunakan Box Reader untuk melakukan pembacaan.
2. Untuk kedepannya program ini diharapkan bisa dijalankan dengan menggunakan device dari perangkat gadget yang lain, seperti kamera seluler, kamera tablet pc dan berbagai macam gadget lainnya akan menambah nilai fleksibilitas dari program yang dibuat ini karena berbasis mobile.
3. Dalam pengembangannya aplikasi ini diharapkan tidak hanya mengenali karakter 0-9 tetapi bisa semua karakter dengan cara menambahkan jumlah warna yang dikombinasikan akan menambah karakter yang bisa dibaca oleh *Color Code* sehingga tidak terbatas pada karakter *numeric*.
4. *Color Code* ini untuk kedepannya bisa digunakan pada semua aplikasi seperti: sistem absen pegawai, sistem sensus penduduk, ISBN buku dan masih banyak yang lain yang berhubungan dengan kode yang tidak boleh terjadi *redundancy*.

DAFTAR PUSTAKA

- [1] Balza Achmad, Kartika Firdausy, 2005, *Teknik Pengolahan Citra Digital Menggunakan Delphi*, Ardi Publishing.
- [2] Budi Susanto, 2012, *Membangun Sistem Basis Data Dengan ORACLE XE*, Andi Publisher.
- [3] Distributed Database Using Oracle.pdf (diakses tanggal 02 Maret 2012).
- [4] Magaline Ferdinand, 2012, *SI*, [pdf], (<http://apr11-si.comuf.com/SI.pdf>, diakses tanggal 2 Januari 2013).
- [5] Noerdityo Iswanto¹⁾, Koredianto Usman²⁾, Ledy Novamizanti³⁾ “ *Desain dan Implementasi Color Code Untuk Verifikasi Nomor Kendaraan Bermotor pada Sistem Parkir*”, 2012, Program Studi Telekomunikasi, Institut Teknologi Telkom.
- [6] Putri Juwita Utami, 2010, *MATERI DELPHI*, [pdf], (<http://jutami.staff.gunadarma.ac.id/Downloads/files/29491/Sejarah+Delphi.doc>, diakses tanggal 15 Januari 2013)
- [7] Rafael C. Gonzalez and Richard E. Woods, 2002, *Digital Image Processing*, Addison-Wesley Publishing.
- [8] Sembiring Abdul Sani, 2012, *Pembentukan Citra*, [pdf], (http://asanisembiring.files.wordpress.com/2012/02/bab-2_pembentukan-citra.pdf, diakses tanggal 19 November 2013)
- [9] Setiaji Hari, 2012, *Distributed Database Using Oracle*, [pdf], (<http://www.slideshare.net/harisetiaji31/distributed-database-using-oracle>, diakses tanggal 2 Maret 2012)
- [10] Sirait Hasanuddin, 2012, *TEORI DASAR CITRA DIGITAL*, [pdf], (<http://sirraith.files.wordpress.com/2009/02/citra-digital-13.pdf>, diakses tanggal 19 November 2012)
- [11] T.Sutoyo, Edy Mulyanto, Vincent Suhartono, Oky Dwi Nurhayati, Wijanarto, 2009, *Teori Pengolahan Citra Digital*, Andi Publisher.
- [12] Tata Laksana Gudang PLTA SUTAMI, 2010.



INSTITUT TEKNOLOGI NASIONAL MALANG

Fakultas Teknologi Industri

Program Studi Teknik Informatika S1

Jl. Raya KarangloKm. 2 Malang

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

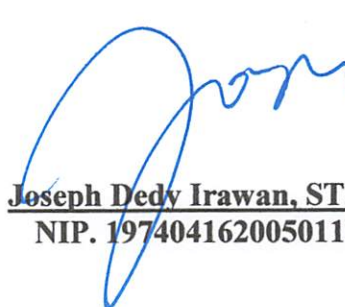
Nama : Donny Eko Suprayogo
NIM : 09.18.103
Jurusan : Teknik Informatika S-1
Judul : Rancang Bangun Color Recognition Dengan Menggunakan Metode
Dynamic Color Thresholding Dalam Sistem Informasi Gudang Yang
Terintegrasi Dengan Oracle

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :

Hari : Selasa
Tanggal : 19 Februari 2013
Nilai : 87.15 (A)

Panitia Ujian Skripsi :

Ketua Majelis Penguji



Joseph Dedy Irawan, ST. MT.
NIP. 197404162005011002

Anggota Penguji :

Dosen Penguji I



Sandy Nataly Mantja, S.Kom.
NIP.P. 1030800418

Dosen Penguji II



Suryo Adi Wibowo, ST
NIP.P. 1031000438



FORMULIR PERBAIKAN SKRIPSI

Nama : Donny Eko Suprayogo
NIM : 09.18.103
Prodi : Teknik Informatika S-1
Judul : Rancang Bangun Color Recognition Dengan Menggunakan
Metode Dynamic Color Thresholding Dalam Sistem Informasi
Gudang Yang Terintegrasi Dengan Oracle

Tanggal	Penguji	Uraian	Paraf
19 Februari 2013	I	- Kata Pengantar - Tidak boleh menggunakan button numbering harus memakai abjad - Tidak ada tabel relasi - Kesimpulan dan Saran	
19Februari2013	II	- Kesimpulan - Nilai Error - ERD - DFD	

Anggota Penguji :

Dosen Penguji I

Sandy Nataly Mantja, S.Kom.
NIP.P. 1030800418

Dosen Penguji II

Suryo Adi Wibowo, ST
NIP.P. 1031000438

Mengetahui

Dosen Pembimbing I

Ir. Sidik Noertjahjono, MT.
NIP.Y. 1028700163

Dosen Pembimbing II

Ali Mahmudi, BEng, PhD
NIP.P. 1031000429



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

T. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Nomor : ITN-147/T.INF/TA/2012
2012

17 Oktober

Lampiran : -
Perihal : Bimbingan Skripsi

Kepada : Yth. Sdr. Ir. Sidik Noertjahjono, MT
Dosen Pembimbing Program Studi Teknik Informatika S1
Institut Teknologi Nasional
M a l a n g

Dengan hormat

Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa :

Nama : DONNY EKO SUPRAYOGO
Nim : 0918103
Prodi : Teknik Informatika S1
Fakultas : Teknologi Industri

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu 6 (enam) bulan, terhitung mulai tanggal ;

17 Oktober 2012 – 17 April 2013

Sebagai satu syarat untuk menempuh Ujian Sarjana Teknik, Program Studi Teknik Informatika S1.

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.



Mengetahui

Program Studi Teknik Informatika S1
Ketua,

Joseph Dedy Irawan, ST, MT
NIP. 197404162005021002

Form S-4a



INSTITUT TEKNOLOGI NASIONAL MALANG
Fakultas Teknologi Industri
Program Studi Teknik Informatika S1
Jl. Raya Karanglo Km. 2 Malang

FORMULIR BIMBINGAN SKRIPSI

Nama : Donny Eko Suprayogo
Nim : 09.18.103
Masa Bimbingan : 17 Oktober 2012 s/d 17 April 2013
Judul Skripsi : Rancang Bangun Color Recognition Dengan Menggunakan Metode Dynamic Color Thresholding Dalam Sistem Informasi Gudang Yang Terintegrasi Dengan Oracle

NO	TANGGAL	URAIAN	PARAF PEMBIMBING
1	18-10-2012	Konsultasi judul skripsi	
2	26-11-2012	Konsultasi Metode Color Recognition (tambahan alat penerangan internal)	
3	14-12-2012	System pembacaan kode perbaikan pada USB Splitter	
4	9-1-2013	Perbaikan penulisan pada BAB IV	
5	16-1-2013	Perbaikan penulisan pada BAB III	
6	26-1-2013	Perbaikan penulisan BAB I & BAB II	
7	30-1-2013	Penambahan metode pengujian yang dipakai untuk pengujian program	
8	1-2-2013	Perbaikan penulisan pada kesimpulan abstrak pada makalah seminar hasil	
9	12-2-2013	Penyempurnaan metode penerangan box reader untuk mengoptimalkan intensitas cahaya	
10	15-2-2013	Konsultasi keseluruhan laporan skripsi	

Malang, 8 Maret 2013
Dosen Pembimbing I



Ir. Sidik Noertjahjono, MT
NIP.Y. 1028700163



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

**FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK**

BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Nomor : ITN-147/T.INF/TA/2012
2012
Lampiran : -
Perihal : Bimbingan Skripsi

17 Oktober

Kepada : Yth. Sdr. Ali Mahmudi, BEng, PhD
Dosen Pembimbing Program Studi Teknik Informatika S1
Institut Teknologi Nasional
M a l a n g

Dengan hormat

Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa :

Nama : DONNY EKO SUPRAYOGO
Nim : 0918103
Prodi : Teknik Informatika S1
Fakultas : Teknologi Industri

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu 6 (enam) bulan, terhitung mulai tanggal ;

17 Oktober 2012 – 17 April 2013

Sebagai satu syarat untuk menempuh Ujian Sarjana Teknik, Program Studi Teknik Informatika S1.

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.



Mengetahui

Program Studi Teknik Informatika S1
Ketua,

Joseph Bedy Irawan, ST, MT
NIP : 197404162005021002

Form S-4a



INSTITUT TEKNOLOGI NASIONAL MALANG
Fakultas Teknologi Industri
Program Studi Teknik Informatika S1
Jl. Raya Karanglo Km. 2 Malang

FORMULIR BIMBINGAN SKRIPSI

Nama : Donny Eko Suprayogo
Nim : 09.18.103
Masa Bimbingan : 17 Oktober 2012 s/d 17 April 2013
Judul Skripsi : Rancang Bangun Color Recognition Dengan Menggunakan Metode Dynamic Color Thresholding Dalam Sistem Informasi Gudang Yang Terintegrasi Dengan Oracle

NO	TANGGAL	URAIAN	PARAF PEMBIMBING
1	15-11-2012	Konsultasi BAB I dan BAB II	
2	29-11-2012	Konsultasi Implementasi Program	
3	19-12-2012	Konsultasi BAB III dan BAB IV	
4	5-1-2013	Demo program keseluruhan	
5	25-1-2013	Konsultasi makalah seminar hasil	
6	4-2-2013	Konsultasi keseluruhan skripsi	

Malang, 8 Maret 2013
Dosen Pembimbing II

Ali Mahmudi, BEng, PhD
NIP.P. 1031000429



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

**FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK**

3NI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Nomor : ITN - 230 /III.FTI/18/12
Lampiran : -
Perihal : **Survey**

Malang, 28 Nopember 2012

Kepada : Yth. Pimpinan
PT. PJB UP BRANTAS PLTA SUTAMI
Jl. Basuki Rachmat No. 271 Karangates
Di – Sumber Pucung - Malang

Bersama ini dengan hormat kami mohon kebijaksanaan Bapak/Ibu agar Mahasiswa kami dari **Fakultas Teknologi Industri Jurusan Teknik Informatika S-1** dapat diijinkan untuk melaksanakan survey pada PT. PJB UP BRANTAS PLTA SUTAMI yang Bapak/Ibu pimpin untuk mendapatkan data – data guna penyusunan Skripsi .

Mahasiswa tersebut adalah :

Donny Eko Suprayogo Nim. 09.18.103

Adapun lamanya Survey adalah : 30 Hari

Demikian agar maklum dan atas perhatian serta bantuannya kami ucapkan terima kasih.

DEKAN
Fakultas Teknologi Industri

Ir. H. Sidik Noertjahjono, MT
Nip. 1018100036

LAMPIRAN I – SOURCE CODE PROGRAM

1. Source code menampilkan semua port webcam

```
procedure TForm23.FormActivate(Sender: TObject);
var
i:integer;
begin
Bersihkan;
ComboBox1.Items.Clear;
capenum:=Tsysdevenum.create(CLSID_VideoInputDeviceCategory);
  for i :=0 to CapEnum.CountFilters-1 do
    begin
      ComboBox1 .items.add(CapEnum.Filters[i].friendlyName ) ;
    end
  end;
end;
```

2. Source code pilih camera webcam yang di aktifkan dan ditampilkan ke grab pada komponen DSPack

```
procedure TForm23.ComboBox1Select(Sender: TObject);
begin
Image2.Picture:=A1.Picture;
FilterGraph1 .ClearGraph ;
FilterGraph1 .Active :=False ;
Filter1 .BaseFilter .Moniker := CapEnum.getmoniker(ComboBox1 .ItemIndex ) ;
FilterGraph1 .Active :=True ;
with FilterGraph1 as IcaptureGraphBuilder2 do
Renderstream(@PIN_CATEGORY_PREVIEW, nil, Filter1 as IBaseFilter,SampleGrabber1
as IBaseFilter,
VideoWindow1 as IbaseFilter);
FilterGraph1 .Play ;
Timer1.Enabled:=True;
PageControl1.TabIndex:=0;
Timer1.Enabled:=True;
EdStockCode.Clear;
end;
```

3. Source code crooping image

```
procedure TForm23.CropBitmap(InBitmap : TBitmap; X, Y, W, H :Integer);
begin
  BitBlt(InBitmap.Canvas.Handle, 0, 0,w, h, InBitmap.Canvas.Handle, x, y, SRCCOPY);
  InBitmap.Width :=W;
  InBitmap.Height:=H;
end;
```

```

procedure TForm23.GetRGB(Col: TColor; var R, G, B: Byte);
var
  Color: $0..$FFFFFFFF;
begin
  Color := ColorToRGB(Col);
  R := ($000000FF and Color);
  G := ($0000FF00 and Color) Shr 8;
  B := ($00FF0000 and Color) Shr 16;

end;

```

```

Var
  Bmp : TBitmap;
  ColNumb: TColor;
  R, G, B, TR, TG, TB: Byte;
  x, y, Tx, Ty : integer;

begin
  Tanda1.Picture.Bitmap:=Image2.Picture.Bitmap;
  Bmp:=Tanda1.Picture.Bitmap;
  CropBitmap(Bmp,156,104, 70, 35);
  Tanda1.Picture.Assign(Bmp);

  Tanda2.Picture.Bitmap:=Image2.Picture.Bitmap;
  Bmp:=Tanda2.Picture.Bitmap;
  CropBitmap(Bmp,252,104, 70, 35);
  Tanda2.Picture.Assign(Bmp);

```

4. Source code dan pendeteksian nilai warna

```

procedure TForm23.GetRGB(Col: TColor; var R, G, B: Byte);
var
  Color: $0..$FFFFFFFF;
begin
  Color := ColorToRGB(Col);
  R := ($000000FF and Color);
  G := ($0000FF00 and Color) Shr 8;
  B := ($00FF0000 and Color) Shr 16;
end;

for Ty:=0 to I2.Picture.Bitmap.Height-1 do
begin
  for Tx:=0 to I2.Picture.Bitmap.Width-1 do
  begin
    TR:=GetRValue(I2.Picture.Bitmap.Canvas.Pixels[Tx,Ty]);

```

```

TG:=GetGValue(I2.Picture.Bitmap.Canvas.Pixels[Tx,Ty]);
TB:=GetBValue(I2.Picture.Bitmap.Canvas.Pixels[Tx,Ty]);
label36.Caption:=CekWarna(TR,TG,TB);
end;
end;

```

5. Source code proses pendeteksian kode

```

function CekKode(a,b,c : String) : string;
begin
if (a='merah')and (b='biru') and(c='hijau') then
Cekkode:='1';
if (a='hijau')and (b='merah') and(c='biru') then
Cekkode:='2';
if (a='hijau')and (b='biru') and(c='merah') then
Cekkode:='3' ;
if (a='biru')and (b='merah') and(c='hijau') then
Cekkode:='4';
if (a='biru')and (b='hijau') and(c='merah') then
Cekkode:='5';
if (a='merah')and (b='hijau') and(c='merah') then
Cekkode:='6';
if (a='hijau')and (b='merah') and(c='hijau') then
Cekkode:='7';
if (a='biru')and (b='merah') and(c='biru') then
Cekkode:='8';
if (a='merah')and (b='biru') and(c='merah') then
Cekkode:='9';
if (a='merah')and (b='hijau') and(c='biru') then
Cekkode:='0'
end;

```

6. Source code simpan

```

procedure TDataBarang.BtnSimpanDtaBrangClick(Sender: TObject);
begin
if ((trim(EdStockCode.Text)='')
or (trim(MeItemName.Text)='')
or (trim(EdUnit.Text)='')
or (trim(EdSoh.Text=''))) then
ShowMessage('Input Data Barang Gagal.Periksa Lagi Inputan Data')
else
begin
DataModule2.QStock.close;
DataModule2.QStock.SQL.Clear;
DataModule2.QStock.SQL.add('INSERT INTO T_barang');

```



```

DataModule2.QStock.SQL.add('VALUES
(:Stock_code,:Item_Name,:Desc_Linux1,:Desc_Linux2,:UNIT_OF_ISSUE,:ESOHINVEN
T_COST_PR,'
+':LAST_REC_DATE,:CRITICALITY,:STATUS,:ROP,:REORDER_QTY,:SOH,:VOLU
ME_KARTU,:VOLUME_FISIK,:SELISIH_LEBIH,:SELISIH_KURANG,:BIN_LOKASI_
LAMA,:BIN_LOKASI_BARU,:KETERANGAN,:JENIS,:Gambar'));
DataModule2.QStock.ParamByName('Stock_Code').Value:=EdStockCode.Text;
DataModule2.QStock.ParamByName('Item_Name').Value:=MeItemName.Text;
DataModule2.QStock.ParamByName('Desc_Linux1').Value:=MeDescLinux1.Text;
DataModule2.QStock.ParamByName('Desc_Linux2').Value:=MeDescLinux2.Text;
DataModule2.QStock.ParamByName('UNIT_OF_ISSUE').Value:=EdUnit.Text;
DataModule2.QStock.ParamByName('ESOHINVENT_COST_PR').Value:=EdESoh
.Text;
DataModule2.QStock.ParamByName('LAST_REC_DATE').Value:=EdLastRec.Text;
DataModule2.QStock.ParamByName('CRITICALITY').Value:=EdCriticaly.Text;
DataModule2.QStock.ParamByName('STATUS').Value:=EdStatus.Text;
DataModule2.QStock.ParamByName('ROP').Value:=EdRop.Text;
DataModule2.QStock.ParamByName('REORDER_QTY').Value:=EdReorder.Text;
DataModule2.QStock.ParamByName('SOH').Value:=EdSoh.Text;
DataModule2.QStock.ParamByName('VOLUME_KARTU').Value:=EdKartu.Text;
DataModule2.QStock.ParamByName('VOLUME_FISIK').Value:=EdFisik.Text;
DataModule2.QStock.ParamByName('SELISIH_LEBIH').Value:=EdLebih.Text;
DataModule2.QStock.ParamByName('SELISIH_KURANG').Value:=EdKurang.Text;
DataModule2.QStock.ParamByName('BIN_LOKASI_LAMA').Value:=EdLama.Text;
DataModule2.QStock.ParamByName('BIN_LOKASI_BARU').Value:=EdBaru.Text;
DataModule2.QStock.ParamByName('KETERANGAN').Value:=MeKet.Text;
DataModule2.QStock.ParamByName('JENIS').Value:=Label16.Caption;
DataModule2.QStock.ParamByName('gambar').Assign(Image3.Picture);
DataModule2.QStock.ExecSQL;
refreshBarang;
BersihkanBarang;
Showmessage('Data sudah berhasil disimpan');
end;
end;

```

7. Source code Edit

```

procedure TDataBarang.BtnEditDtaBrangClick(Sender: TObject);
begin
if ((trim(EdStockCode.Text)='')
or (trim(MeItemName.Text)='')
or (trim(EdUnit.Text)='')
or (trim(EdSoh.Text)='')) then
ShowMessage('Edit Data Barang Gagal.Periksa Lagi Inputan Data' )
else

```

```

begin
DataModule2 .QStock .Close;
DataModule2 .QStock.SQL.Clear;
DataModule2 .QStock.SQL.add('UPDATE t_barang SET
Item_Name=:a,Desc_Linex1=:b,Desc_Linex2=:c,UNIT_OF_ISSUE=:d,ESOHINVENT_C
OST_PR=:e,LAST_REC_DATE=:f,CRITICALITY=:g,STATUS=:h,ROP=:i,REORDER_
QTY=:j,SOH=:k,VOLUME_KARTU=:l,VOLUME_FISIK=:m,'
+'SELISIH_LEBIH=:n,SELISIH_KURANG=:o,BIN_LOKASI_LAMA=:p,BIN_LOKASI
_BARU=:q,KETERANGAN=:r Where Stock_code=:s ');
DataModule2 .QStock.ParamByName('a').AsString:=MeItemName.Text;
DataModule2 .QStock.Params.ParamByName('b').AsString:= MeDescLinex1.Text;
DataModule2 .QStock.Params.ParamByName('c').AsString:=MeDescLinex2.Text;
DataModule2 .QStock.ParamByName('d').AsString:=EdUnit.Text;
DataModule2 .QStock.Params.ParamByName('e').AsString:= EdESoh.Text;
DataModule2 .QStock.Params.ParamByName('f').AsString:=EdLastRec.Text;
DataModule2 .QStock.Params.ParamByName('g').AsString:=EdCriticaly.Text;
DataModule2 .QStock.Params.ParamByName('h').AsString:=EdStatus.Text;
DataModule2 .QStock.Params.ParamByName('i').AsString:=EdRop.Text;
DataModule2 .QStock.Params.ParamByName('j').AsString:=EdReorder.Text;
DataModule2 .QStock.Params.ParamByName('k').AsString:=EdSoh.Text;
DataModule2 .QStock.Params.ParamByName('l').AsString:=EdKartu.Text;
DataModule2 .QStock.Params.ParamByName('m').AsString:=EdFisik.Text;
DataModule2 .QStock.Params.ParamByName('n').AsString:=EdLebih.Text;
DataModule2 .QStock.Params.ParamByName('o').AsString:=EdKurang.Text;
DataModule2 .QStock.Params.ParamByName('p').AsString:=EdLama.Text;
DataModule2 .QStock.Params.ParamByName('q').AsString:=EdBaru.Text;
DataModule2 .QStock.Params.ParamByName('r').AsString:=MeKet.Text;
DataModule2 .QStock.Params.ParamByName('s').AsString:=EdStockCode.Text;
DataModule2 .QStock.ExecSQL;
refreshBarang;
BersihkanBarang;
DataModule2 .QStock .Open;
Showmessage('Data Berhasil Update');
end;
end;

```

8. Source code Cari

```

procedure TDataBarang.BtnCariDtaBrangClick(Sender: TObject);
begin
if ((trim(EdStockCode.Text)='') ) then
ShowMessage('Pencarian Data Gagal Anda Belum Masukan Stock Code' )
else
begin
DataModule2 .QStock.Close;

```

```

DataModule2 .QStock.SQL.Clear;
DataModule2 .QStock.SQL.add('select * from t_barang where stock_code like :a');
DataModule2 .QStock.ParamByName('a').AsString:='%'+EdStockCode.Text+'%';
DataModule2 .QStock.Open;
EdStockCode.Text:=DataModule2 .QStock.FieldName('Stock_Code').AsString;
MeItemName.Text:=DataModule2 .QStock.FieldName('Item_Name').AsString;
MeDescLinex1 .Text:=DataModule2 .QStock.FieldName('Desc_Linex1').AsString;
MeDescLinex2.Text:=DataModule2 .QStock.FieldName('Desc_Linex2').AsString;
EdUnit.Text:=DataModule2 .QStock.FieldName('UNIT_OF_ISSUE').AsString;
EdESoh.Text:=DataModule2
.QStock.FieldName('ESOHINVENT_COST_PR').AsString;
EdLastRec.Text:=DataModule2 .QStock.FieldName('LAST_REC_DATE').AsString;
EdCriticaly.Text:=DataModule2 .QStock.FieldName('CRITICALITY').AsString;
EdStatus.Text:=DataModule2 .QStock.FieldName('STATUS').AsString;
EdRop.Text:=DataModule2 .QStock.FieldName('ROP').AsString;
EdReorder.Text:=DataModule2 .QStock.FieldName('REORDER_QTY').AsString;
EdSoh.Text:=DataModule2 .QStock.FieldName('SOH').AsString;
EdKartu.Text:=DataModule2 .QStock.FieldName('VOLUME_KARTU').AsString;
EdFisik.Text:=DataModule2 .QStock.FieldName('VOLUME_FISIK').AsString;
EdLebih.Text:=DataModule2 .QStock.FieldName('SELISIH_LEBIH').AsString;
EdKurang.Text:=DataModule2 .QStock.FieldName('SELISIH_KURANG').AsString;
EdLama.Text:=DataModule2 .QStock.FieldName('BIN_LOKASI_LAMA').AsString;
EdBaru.Text:=DataModule2 .QStock.FieldName('BIN_LOKASI_BARU').AsString;
MeKet.Text:=DataModule2 .QStock.FieldName('KETERANGAN').AsString;
end;
end;

```

9. Source code Hapus

```

procedure TDataBarang.BtnHapusDtaBrangClick(Sender: TObject);
begin
if messagedlg('Yakin '+MeItemName .Text +' ini akan
dihapus',mtconfirmation,MBOkCancel,0)=mrOK then
begin
DataModule2 .QStock.Close;
DataModule2 .QStock.SQL.clear;
DataModule2 .QStock.SQL.Add('DELETE FROM t_Barang');
DataModule2 .QStock.SQL.Add('WHERE Stock_Code = '"+EdStockCode .Text +" '");
DataModule2 .QStock.ExecSQL;
refreshBarang;
BersihkanBarang;
Showmessage('Data '+MeItemName .Text +' sudah terhapus');
end;

end;

```

10. Source code Import Excel To MySQL

```

procedure TKonvert.DisplayException(Sender: TObject; E: Exception);
begin
  StatusBar1.SimpleText := E.Message;
  MessageDlg(E.Message,mtError,[MBOK],0);
end;

function TKonvert.GiveRowHeader(x: Integer) : String;
begin
  if (x > 26) then
    Result := GiveRowHeader(x div 26) + GiveRowHeader(x mod 26)
  else
    Result := Chr(Ord('A') + x - 1);
  end;
end;

procedure TKonvert.LoadFromXLFile(AFileName : String; ASheetNum: Integer;
AStringGrid:TStringGrid; var OSheetName : String);
var
  ARow,
  ACol,
  i,
  j      : Integer;
  XLApp,
  XLBook,
  XLSheet : Variant;
begin
  XLApp := CreateOLEObject('Excel.Application');
  XLApp.Visible := False;
  XLApp.Workbooks.Open(AFileName,
                        EmptyParam,
                        EmptyParam,
                        EmptyParam,
                        EmptyParam,
                        EmptyParam,
                        EmptyParam,
                        EmptyParam,
                        EmptyParam,
                        EmptyParam,
                        EmptyParam,
                        EmptyParam,
                        EmptyParam,
                        EmptyParam,
                        EmptyParam,
                        EmptyParam,
                        0
                      );
  XLBook := XLApp.ActiveWorkbook;
  XLSheet := XLBook.WorkSheets[ASheetNum];
  XLSheet.Activate;
  XLSheet.Cells.SpecialCells(xlCellTypeLastCell,EmptyParam).Activate;
  OSheetName := XLSheet.Name;
  ARow := XLApp.ActiveCell.Row;
  AStringGrid.RowCount := ARow + 1;
end;

```

```

ACol := XLApp.ActiveCell.Column;
AStringGrid.ColCount := ACol + 1;

for i := 1 to ARow do
begin
  AStringGrid.Cells[0, i] := IntToStr(i);
  for j := 1 to ACol do
  begin
    AStringGrid.Cells[j, 0] := GiveRowHeader(j);
    AStringGrid.Cells[j, i] := XLSheet.Cells[i, j];
  end;
end;

if not VarIsEmpty(XLApp) then
begin
  XLApp.DisplayAlerts := False;
  XLApp.Quit;
  XLApp := Unassigned;
  XLBook := Unassigned;
  XLSheet := Unassigned;
end;
end;

procedure TKonvert.Button1Click(Sender: TObject);
var
  server,
  user,
  password,
  db_name,
  SheetName : string;
  ConnStr,
  StrSQL : WideString;
  ACol,
  ARow,
  NRows : Integer;
begin
  //try
  ADOConnection1.Close;
  server := 'localhost'; //Sesuaikan, bisa diisi IP Address Server MySQL
  user := 'root'; //Sesuaikan dengan username mysql Anda
  password := ''; //Sesuaikan dengan password mysql Anda
  db_name := Edit3.Text ;
  ConnStr := 'Provider=MSDASQL;DRIVER={MySQL ODBC 3.51 Driver}; '
    + 'SERVER='+server
    + '; UID='+user
    + '; PASSWORD='+password
    + '; DATABASE='+db_name;
  ADOConnection1.ConnectionString := ConnStr;
  try
    AdoConnection1.Open;
  except

```

```

    raise;
end;
if ADOConnection1.Connected then begin
    Memo1.Clear;
    Memo1.Lines.Add('Connection established to '+server+'. Logged as '+user+' at
'+DateTimeToStr(now));
    StatusBar1.SimpleText := 'Connection established to '+server+'. Logged as '+user+' at
'+DateTimeToStr(now);
    LoadFromXMLFile(Edit1.Text,1,StringGrid1,SheetName);
    NRows := 0;
    for ARow := 2 to StringGrid1.RowCount-1 do
    begin
        StrSQL := 'INSERT INTO '+Edit2.Text +' (';
        for ACol := 1 to StringGrid1.ColCount-1 do
            StrSQL := StrSQL + LowerCase(StringGrid1.Cells[ACol,1])+',';
        StrSQL := copy(StrSQL,1,length(StrSQL)-1)+' VALUES (';

        for ACol := 1 to StringGrid1.ColCount-1 do
            StrSQL := StrSQL + QuotedStr(StringGrid1.Cells[ACol,ARow])+',';
        StrSQL := copy(StrSQL,1,length(StrSQL)-1)+'';
        Memo1.Lines.Add('QUERY : '+StrSQL);
        ADOQuery1.Close;
        ADOQuery1.SQL.Clear;
        ADOQuery1.SQL.Add(StrSQL);
        ADOQuery1.ExecSQL;
        NRows := NRows + 1;
    end;
    StatusBar1.SimpleText := 'Operation Complete. '+IntToStr(NRows)+' rows inserted';
end;
// Except
//ShowMessage('Periksa Lagi Inputan Anda Masih Ada yang Salah')
//end;
end;

procedure TKonvert.FormCreate(Sender: TObject);
begin
    Memo1.Lines.Clear;
    ADOQuery1.Connection := ADOConnection1;
    Application.OnException := DisplayException;
end;

```

11. Source Code Export Data MySQL to Excel

```

procedure TForm12.BitBtn4Click(Sender: TObject);
var i,x:integer;

Sfile:string;
begin
    DataModule2.QOut.Close;
    DataModule2.QOut.SQL.Clear;
    DataModule2.QOut.SQL.add('select * from t_pengeluaran_barang');

```



```
DataModule2.QOut.Open;  
DataSource1.DataSet:=DataModule2.QOut;
```

```
// buka excel  
XlApp:=CreateOleObject('Excel.Application');  
// tambahkan workbook  
XlBook := XlApp.WorkBooks.Add;  
// tambahkan worksheet  
XlSheet := XlBook.worksheets.add;  
//cetak header field dari dbgrid  
for i:=0 to dbgrid1.FieldCount-1 do  
begin  
XlSheet.cells[2,i+1].value:=dbgrid1.columns[i].Title.Caption;  
end;  
// transfer data ke excel  
DataModule2.QOut.First;  
x:=1;  
while not DataModule2.QOut.Eof do  
begin  
for i:=0 to DataModule2.QOut.FieldCount-1 do  
begin  
XlSheet.cells[2+x,i+1].value:=DataModule2.QOut.Fields[i].Text;  
end;  
DataModule2.QOut.Next;  
inc(x);  
end;
```

```
//menampilkan aplikasi //XlApp.visible:=true;
```

```
//script dibawah ini untuk dialog disimpan atau ditampilkan
```

```
if MessageDlg('Apakah hasil export ditampilkan..?', mtConfirmation, [mbYes, mbNo], 0)  
= mrYes then  
begin  
MessageDlg('Hasil Ditampilkan', mtInformation,  
[mbOk], 0);  
XlApp.visible:=true;  
end  
else  
//simpan ke file  
begin  
Sfile:= InputBox('Nama File', 'hasil export', 'c:\hasil.xls');  
XlApp.ActiveWorkbook.SaveAs(sfile);  
XlApp.visible:=true;  
end  
end;
```

12. Source code tampilkan grafik

```
procedure TForm11.BitBtn1Click(Sender: TObject);
var a:Integer;
begin
If Januari_In=" then
Januari_In:='0';
If Januari_Out=" then
Januari_Out:='0';
If february_In=" then
february_In:='0';
If february_Out=" then
february_Out:='0';
If Maret_In=" then
Maret_In:='0';
If Maret_Out=" then
Maret_Out:='0';
If April_In=" then
April_In:='0';
If April_Out=" then
April_Out:='0';
If Mei_In=" then
Mei_In:='0';
If Mei_Out=" then
Mei_Out:='0';
If Juni_In=" then
Juni_In:='0';
If Juni_Out=" then
Juni_Out:='0';
If Juli_In=" then
Juli_In:='0';
If Juli_Out=" then
Juli_Out:='0';
If Agustus_In=" then
Agustus_In:='0';
If Agustus_Out=" then
Agustus_Out:='0';
If September_In=" then
September_In:='0';
If September_Out=" then
September_Out:='0';
If Oktober_In=" then
Oktober_In:='0';
If Oktober_Out=" then
Oktober_Out:='0';
If November_In=" then
November_In:='0';
If November_Out=" then
November_Out:='0';
If Desember_In=" then
Desember_In:='0';
```

```

If Desember_Out="" then
Desember_Out:='0';
a:=100;
Stok_January:=(a+StrToInt(Januari_In))-StrToInt(Januari_Out);
Stok_February:=(Stok_January+StrToInt(Februari_In))-StrToInt(Februari_Out);
Stok_Maret:=(Stok_February+StrToInt(Maret_In))-StrToInt(Maret_Out);
Stok_April:=(Stok_Maret+StrToInt(April_in))-StrToInt(April_Out);
Stok_Mei:=(Stok_April+StrToInt(Mei_in))-StrToInt(Mei_Out);
Stok_Juni:=(Stok_Mei+StrToInt(juni_in))-StrToInt(juni_Out);
Stok_Juli:=(Stok_Juni+StrToInt(Juli_in))-StrToInt(Juli_Out);
Stok_Agustus:=(Stok_Juli+StrToInt(Agustus_in))-StrToInt(Agustus_Out);
Stok_September:=(Stok_Agustus+StrToInt(September_in))-StrToInt(September_Out);
Stok_Oktober:=(Stok_September+StrToInt(Oktober_in))-StrToInt(Oktober_Out);
Stok_November:=(Stok_Oktober+StrToInt(November_in))-StrToInt(November_Out);
Stok_Desember:=(Stok_November+StrToInt(Desember_in))-StrToInt(Desember_Out);

```

```

DBChart1.Series[0].Clear;
DBChart1.Series[0].Add(StrToFloat(Januari_In),'Januari');
DBChart1.Series[0].Add(StrToFloat(Februari_in),'Februari');
DBChart1.Series[0].Add(StrToFloat(Maret_in),'Maret');
DBChart1.Series[0].Add(StrToFloat(April_in),'April');
DBChart1.Series[0].Add(StrToFloat(Mei_in),'Mei');
DBChart1.Series[0].Add(StrToFloat(juni_in),'Juni');
DBChart1.Series[0].Add(StrToFloat(Juli_in),'Juli');
DBChart1.Series[0].Add(StrToFloat(Agustus_in),'Agustus');
DBChart1.Series[0].Add(StrToFloat(September_in),'September');
DBChart1.Series[0].Add(StrToFloat(Oktober_in),'Oktober');
DBChart1.Series[0].Add(StrToFloat(November_in),'November');
DBChart1.Series[0].Add(StrToFloat(Desember_in),'Desember');

```

```

DBChart1.Series[1].Clear;
DBChart1.Series[1].Add(StrToFloat(Januari_Out),'Januari');
DBChart1.Series[1].Add(StrToFloat(Februari_Out),'Februari');
DBChart1.Series[1].Add(StrToFloat(Maret_Out),'Maret');
DBChart1.Series[1].Add(StrToFloat(April_Out),'April');
DBChart1.Series[1].Add(StrToFloat(Mei_Out),'Mei');
DBChart1.Series[1].Add(StrToFloat(juni_Out),'Juni');
DBChart1.Series[1].Add(StrToFloat(Juli_Out),'Juli');
DBChart1.Series[1].Add(StrToFloat(Agustus_Out),'Agustus');
DBChart1.Series[1].Add(StrToFloat(September_Out),'September');
DBChart1.Series[1].Add(StrToFloat(Oktober_Out),'Oktober');
DBChart1.Series[1].Add(StrToFloat(November_Out),'November');
DBChart1.Series[1].Add(StrToFloat(Desember_Out),'Desember');

```

```

DBChart1.Series[2].Clear;
DBChart1.Series[2].Add(Stok_January,'Januari');
DBChart1.Series[2].Add(Stok_February,'Februari');
DBChart1.Series[2].Add(Stok_Maret,'Maret');
DBChart1.Series[2].Add(Stok_April,'April');
DBChart1.Series[2].Add(Stok_Mei,'Mei');
DBChart1.Series[2].Add(Stok_Juni,'Juni');

```

```
DBChart1.Series[2].Add(Stok_Juli,'Juli');
DBChart1.Series[2].Add(Stok_Agustus,'Agustus');
DBChart1.Series[2].Add(Stok_September,'September');
DBChart1.Series[2].Add(Stok_Oktober,'Oktober');
DBChart1.Series[2].Add(Stok_November,'November');
DBChart1.Series[2].Add(Stok_Desember,'Desember');
end;
```