



**PROSIDING SEMINAR NASIONAL  
PENERAPAN DIGITAL MARKETING  
DAN TEKNOLOGI INFORMASI  
DALAM BISNIS RITEL**

**Hari, Tanggal;  
Di Aula Lantai 4 Kampus  
Universitas Informatika dan Bisnis Indonesia  
Jl. Purnawarman 34-36 B Bandung**



Diselenggarakan Oleh  
Lembaga Penelitian dan Pengabdian Kepada Masyarakat  
**Universitas Informatika dan Bisnis Indonesia**  
2016

**PROSIDING SEMINAR NASIONAL  
PENERAPAN DIGITAL MARKETING  
DAN TEKNOLOGI INFORMASI  
DALAM BISNIS RITEL**



Lembaga Penelitian dan Pengabdian Kepada Masyarakat  
Universitas Informatika dan Bisnis Indonesia



## DAFTAR ISI

1. Pengaruh Kepercayaan Konsumen dan Kualitas Informasi Terhadap Keputusan Pembelian Menggunakan Tokopedia.Com  
**Iqbal Muttaqien, Sabilla Saberina**
2. Analisis Tema Lukisan Mooi Indie Melalui Proses Kreasi Perupa Wanita Indonesia (Studi Kasus Pada Pelukis Wanita Kontemporer Mariam Sofrina)  
**Banon Gilang**
3. Aplikasi Survey Kredit Berbasis Web (Studi Kasus: PT BPR Citradana Rahayu)  
**Vani Maharani Nasution, R. Yadi Rakhman**
4. Perancangan Pelatihan Peer Helping (Suatu Rancangan Pelatihan Untuk Meningkatkan Helping Skill for Understanding Dalam Memberikan Bantuan Pada Staf Mahasiswa di Universitas “X” Bandung)  
**Anggian Heksa Efraim Sinaga**
5. Pengaruh Brand Ambassadors dan Kreatifitas Iklan terhadap Keputusan Pembelian  
**Dikdik Purwadisastra, Angga Permana Putra**
6. Pengaruh Kompensasi Dan Lingkungan Kerja Terhadap Loyalitas Karyawan PT BPR Citradana Rahayu Bandung  
**Ayu Nike Retnowati, Ilin Saputra**
7. Pengaruh Endorse Non Celebrity dan Consumer Trust Terhadap Keputusan Pembelian Konsumen The Body Shop di Bandung Indah Plaza  
**Nurul Lastri, Elan Rusnendar**
8. Aplikasi Pengujian Tanah (CBR – DCP) Untuk Menentukan Tebal Perkerasan Jalan Menggunakan Standar Bina Marga  
**Marwondo, Vani Maharani Nasution**
9. Sistem Pendukung Keputusan Pemilihan Supplier Daging Terbaik Menggunakan Metode AHP (Studi Kasus pada PT. Pandiga Educreation Sport)  
**Graha Prakarsa**
10. *E-Commerce* Lelang pada Divisi AMU di PT WOM Finance, Tbk  
**Graha Prakarsa**
11. Aplikasi Pengolahan Data Peminjaman dan Pengembalian Buku Komik Studi Kasus Pada Daruma Komik  
**Rosalin Samihardjo, Rd. Yadi Rakhman Alamsyah, Feybe Melia Longdong**

12. Aplikasi Penjualan Hasil Tani dan Investasi Gabungan Kelompok Tani Berbasis Web  
**Akbar Pasha, Reni Nursyanti**
13. Pengaruh Profesionalisme Auditor Terhadap Kualitas Audit  
**Rebecca Kizia**
14. Perancangan Sistem Pengolahan Data Dengan Pemodelan Berorientasi Objek di SMK Muhammadiyah Majalaya  
**Aminudin, Rosita**
15. Pengembangan Game Edukasi 2D Rambu Lalu Lintas (Studi Kasus: TK Bayangkari)  
**Akbar Pasha, Marwondo, Vani Maharani Nasution**
16. Pengaruh Kompensasi Finansial dan Motivasi Terhadap Employee Engagement Pada PT Sanbe Farma Unit 2 Cimahi Divisi Produksi  
**Indarta Priyana, Amelia Mulya Supendi**
17. Perilaku Konsumen Dalam Membeli Produk Bakery di Kota Bandung  
**Ratih Hadiantini**
18. Pengaruh Promosi Dan Kualitas Produk Terhadap Keputusan Pembelian di Abracajava Coffee  
**Vika Aprianti**

**APLIKASI PENGUJIAN TANAH (CBR – DCP) UNTUK MENENTUKAN TEBAL  
PERKERASAN JALAN MENGGUNAKAN STANDAR BINA MARGA**Marwondo<sup>1</sup>, Vani Maharani Nasution<sup>2</sup>

Fakultas Teknologi dan Informatika, UNIBI

[marwondo@unibi.ac.id](mailto:marwondo@unibi.ac.id), [vani.maharani@unibi.ac.id](mailto:vani.maharani@unibi.ac.id)**Abstrak**

Jaringan jalan raya merupakan prasarana darat yang memegang peranan sangat penting dalam sektor perhubungan, terutama untuk kesinambungan distribusi barang dan jasa. Keberadaan jalan raya sangat diperlukan untuk menunjang laju pertumbuhan ekonomi seiring dengan meningkatnya kebutuhan sarana transportasi yang dapat menjangkau daerah-daerah terpencil yang merupakan sentra produksi pertanian, perkebunan, ataupun produk-produk dan jasa-jasa lainnya. Dalam pembuatan jalan raya yang tepat guna, diperlukan perencanaan yang matang untuk mengefisienkan perkerasan jalan, salah satunya adalah dengan melakukan perhitungan dan perencanaan yang terperinci untuk meminimalisir kegagalan perencanaan akibat perhitungan yang salah ataupun tidak sesuai dengan kondisi di lapangan. Dalam mengatasi kendala kegagalan dalam perencanaan jalan, perlu dilakukan upaya agar proses perhitungan dapat dilakukan secara lebih sederhana dan mudah, yaitu dengan menggunakan teknologi komputer sebagai suatu alat bantu. Hal ini diperlukan sebagai upaya untuk meningkatkan kemampuan dan kapasitas kerja sehingga diperoleh hasil yang akurat, berkapasitas tinggi dan dapat dilakukan dalam waktu cepat. Salah satunya adalah dengan pembuatan perangkat lunak dan interface *User friendly* yang mudah dalam penggunaannya, tetapi dapat melakukan perhitungan dan perencanaan secara cepat dan akurat sesuai dengan standar yang berlaku, seperti standar perencanaan jalan Standar Bina Marga.

**Kata Kunci :** Perkerasan Jalan, Perencanaan Jalan, Standar Bina Marga**Abstract**

*Highway network is a terrestrial infrastructure which plays an important role in the communications sector, particularly for the sustainability of the distribution of goods and services. The existence of the highway is needed to support economic growth by increasing transportation needs that can reach remote areas as the center of agricultural production, plantation, or the products and other services. In making appropriate highway, it takes careful planning to streamline the pavement, one of which is to perform calculations and detailed planning to minimize failures due to incorrect calculations of planning or not in accordance with the conditions in the field. Failure to overcome obstacles in the path planning, efforts are required in order to process calculations can be done more simply and easily, using computer technology as a tool. This is necessary in order to enhance the ability and work capacity in order to obtain accurate results, high capacity and can be done in quick time. One way is to manufacture the software and User friendly interface that is easy in use, but can perform calculations and planning quickly and accurately in accordance with the applicable standards, such as road planning standards of Bina Marga Standards.*

**Keywords:** *Pavement Roads, Road Planning, Highways Standard*

## 1. PENDAHULUAN

### 1.1 Latar Belakang

Jaringan jalan raya merupakan prasarana darat yang memegang peranan sangat penting dalam sektor perhubungan, terutama untuk kesinambungan distribusi barang dan jasa. Keberadaan jalan raya sangat diperlukan untuk menunjang laju pertumbuhan ekonomi seiring dengan meningkatnya kebutuhan sarana transportasi yang dapat menjangkau daerah-daerah terpencil yang merupakan sentra produksi pertanian, perkebunan, ataupun produk-produk dan jasa-jasa lainnya. Dalam pembuatan jalan raya yang tepat guna, diperlukan perencanaan yang matang untuk mengefisienkan perkerasan jalan, salah satunya adalah dengan melakukan perhitungan dan perencanaan yang terperinci untuk meminimalisir kegagalan perencanaan akibat perhitungan yang salah ataupun tidak sesuai dengan kondisi di lapangan.

Banyaknya metode untuk perhitungan secara ilmu sipil, khususnya dalam perencanaan perkerasan jalan untuk menentukan tebal dan bahan lapisan perkerasan yang sesuai, mulai dari pengambilan nilai CBR (California Bearing Ratio) lapangan dan DDT (Daya Dukung Tanah) menggunakan alat DCP (*Direct Cone Penetration*) hingga analisa bahan dan perhitungan tebal perkerasan jalan yang sesuai dengan standar Bina Marga. Tahapan tersebut banyak menggunakan variabel dan prosedur perhitungan yang cukup panjang, sehingga akan memerlukan waktu antara 1 sampai 3 bulan, dan biaya yang cukup besar untuk menyelesaikannya apabila dilakukan dengan cara manual. Terlebih lagi, bila perencanaan tersebut harus dilakukan dengan merubah-ubah nilai variabel dalam proses perencanaannya, dikarenakan terbatasnya ketersediaan bahan ataupun permasalahan lainnya, yang akan makin menambah waktu dan biaya perencanaan.

Dalam mengatasi kendala ini perlu dilakukan upaya agar proses perhitungan

dapat dilakukan secara lebih sederhana dan mudah, yaitu dengan menggunakan teknologi komputer sebagai suatu alat bantu. Hal ini diperlukan sebagai upaya untuk meningkatkan kemampuan dan kapasitas kerja sehingga diperoleh hasil yang akurat, berkapasitas tinggi dan dapat dilakukan dalam waktu cepat. Salah satunya adalah dengan pembuatan perangkat lunak dan *friendly User Interface* yang mudah dalam penggunaannya, tetapi dapat melakukan perhitungan dan perencanaan secara cepat dan akurat sesuai dengan standar yang berlaku, seperti standar perencanaan jalan Standar Bina Marga.

Berdasarkan latar belakang di atas, maka penulis mengambil judul Penelitian **“Aplikasi Pengujian Tanah (CBR-DCP) Untuk Menentukan Tebal Perkerasan Jalan Menggunakan Standar Bina Marga”**, dimana Standar Bina Marga untuk perhitungan Perkerasan Lentur akan menggunakan metode Analisa Komponen, dan untuk Perkerasan Kaku akan menggunakan metode NAASRA (National Association of Australian State Road Authorities).

### 1.2 Identifikasi Masalah

Berdasarkan latar belakang masalah yang telah dikemukakan di atas, maka dapat diidentifikasi masalah sebagai berikut:

1. Bagaimana tahapan membangun aplikasi yang direncanakan agar dapat menunjang pekerjaan konsultan tanah, konsultan bahan, dan konsultan jalan yang sesuai dengan standar Bina Marga?
2. Bagaimana bentuk User Interface perangkat lunak yang direncanakan agar lebih mudah dipahami dan digunakan oleh pengguna?
3. Bagaimana hasil percobaan alat dan perangkat lunak tersebut ketika dipakai pada kondisi lapangan asli?
4. Bagaimana hasil perhitungan perangkat lunak tersebut saat dibandingkan

dengan perhitungan manual dalam rangka pengujian aplikasi?

### 1.3 Rumusan Masalah

Berdasarkan latar belakang masalah di atas, maka penulis dapat merumuskan masalah bagaimana melakukan perhitungan dan perencanaan lapisan perkerasan jalan Standar Bina Marga dengan menggunakan program Visual Basic 6

### 1.4 Batasan Masalah

Dalam perhitungan lapis perkerasan jalan yang sangat populer di Indonesia adalah dengan menggunakan Standar Bina Marga. Oleh sebab itu penulis mencoba membuat perhitungan dan perencanaan lapisan perkerasan jalan Standar Bina Marga dengan menggunakan program Visual Basic

Selain batasan di atas, dalam penulisan Penelitian ini diperlukan pembatasan-pembatasan masalah sehubungan dengan keterbatasan dan kemampuan penyusun. Pembatasan masalah tersebut adalah sebagai berikut:

1. Script pemrograman dan Visual Basic 6 tidak akan dibahas secara terperinci, penulis hanya menyajikan Use Case Diagram pemrogramannya saja.
2. Variabel input data tanah diperoleh dari alat elektronika yang penulis desain dan dibuat sendiri, meskipun komponen-komponen yang dipakai merupakan hak cipta dari produsen-produsen pembuat komponen-komponen itu sendiri.
3. Variabel input data-data perencanaan seperti data Lalu Lintas Harian Rencana (LHR), Faktor Regional, dan Tipe atau Jenis Jalan, merupakan data hasil dari data lapangan atau merupakan asumsi dari pemakai perangkat lunak.
4. Perhitungan tebal perkerasan lentur dan perkerasan kaku menggunakan Standar Bina Marga untuk pembuatan jalan baru, sedangkan untuk perbaikan dan rehabilitasi jalan hanya dilakukan untuk jalan hotmix dan macadam.

5. Tebal perkerasan kaku yang dihitung adalah tebal perkerasan kaku tanpa tulangan.
6. Sistem Perangkat Keras yang ditambahkan untuk menjadi alat bantu pada aplikasi yang akan dibangun tidak akan dibahas secara mendetail.

## 2. LANDASAN TEORI

### 2.1 Software Engineering

Istilah Rekayasa Perangkat Lunak (Software Engineering) secara umum disepakati sebagai terjemahan dari istilah Software Engineering. Istilah Software Engineering mulai dipopulerkan pada tahun 1968 pada Software Engineering Conference yang diselenggarakan oleh NATO. Sebagian orang mengartikan Software Engineering hanya sebatas pada bagaimana membuat program komputer. Padahal ada perbedaan yang mendasar antara perangkat lunak (software) dan program komputer.

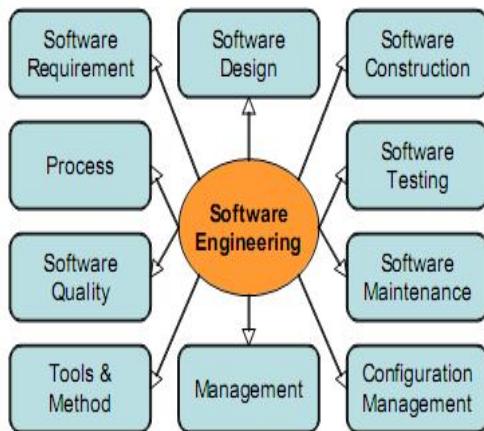
Perangkat lunak adalah seluruh perintah yang digunakan untuk memproses informasi. Perangkat lunak dapat berupa program atau prosedur. Program adalah kumpulan perintah yang dimengerti oleh komputer, sedangkan prosedur adalah perintah yang dibutuhkan oleh pengguna dalam memproses informasi (O'Brien, 1999).

Pengertian Software Engineering sendiri adalah suatu disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal yaitu analisa kebutuhan pengguna, menentukan spesifikasi dari kebutuhan pengguna, desain, pengkodean, pengujian sampai pemeliharaan sistem setelah digunakan. Dari pengertian ini jelaslah bahwa Software Engineering tidak hanya berhubungan dengan cara pembuatan program komputer. Pernyataan "semua aspek produksi" pada pengertian di atas, mempunyai arti semnua hal yang berhubungan dengan proses produksi seperti manajemen proyek, penentuan personil, anggaran biaya, metode, jadwal, kualitas sampai dengan pelatihan pengguna

merupakan bagian dari Software Engineering.

## 2.2 Ruang Lingkup Software Engineering

Sesuai dengan definisi yang telah disampaikan sebelumnya, maka ruang lingkup *Software Engineering* dapat digambarkan sebagai berikut:



Gambar 2.1

Ruang lingkup *Software Engineering* (Abran et.al., 2004).

Berdasarkan gambar di atas dapat dilihat bahwa ruang lingkup *Software Engineering* beberapa langkah yaitu:

1. *Software Requirements* berhubungan dengan spesifikasi kebutuhan dan persyaratan perangkat lunak
2. *Software Design* mencakup proses penampilan arsitektur, komponen, antar muka, dan karakteristik lain dari perangkat lunak
3. *Software Construction* berhubungan dengan detail pengembangan perangkat lunak, termasuk algoritma, pengkodean, pengujian dan pencarian kesalahan
4. *Software Testing* meliputi pengujian pada keseluruhan perilaku perangkat lunak
5. *Software Maintenance* mencakup upaya-upaya perawatan ketika perangkat lunak telah dioperasikan
6. *Software Configuration* management berhubungan dengan usaha perubahan

konfigurasi perangkat lunak untuk memenuhi kebutuhan tertentu

7. *Software Engineering Management* berkaitan dengan pengelolaan dan pengukuran *Software Engineering*, termasuk perencanaan proyek perangkat lunak
8. *Software Engineering Tools And Methods* mencakup kajian teoritis tentang alat bantu dan metode *Software Engineering*
9. *Software Engineering Process* berhubungan dengan definisi, implementasi pengukuran, pengelolaan, perubahan dan perbaikan proses *Software Engineering*
10. *Software Quality* menitik beratkan pada kualitas dan daur hidup perangkat lunak

## 2.3 Tahapan Software Engineering

Meskipun dalam pendekatan berbeda-beda, namun model-model pendekatan memiliki kesamaan, yaitu menggunakan pola tahapan *analysis-design-coding (construction)-testing-maintenance*.

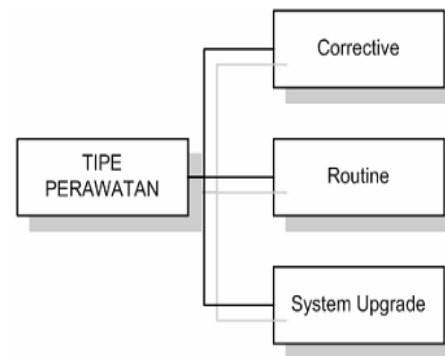
1. **Analisis sistem** adalah sebuah teknik pemecahan masalah yang menguraikan sebuah sistem menjadi komponen-komponennya dengan tujuan mempelajari seberapa bagus komponen-komponen tersebut bekerja dan berinteraksi untuk meraih tujuan mereka. Analisis mungkin adalah bagian terpenting dari proses rekayasa perangkat lunak. Karena semua proses lanjutan akan sangat bergantung pada baik tidaknya hasil analisis. Ada satu bagian penting yang biasanya dilakukan dalam tahapan analisis yaitu pemodelan proses bisnis.
2. **Model proses** adalah model yang memfokuskan pada seluruh proses di dalam sistem yang mentransformasikan data menjadi informasi (Harris, 2003). Model proses juga menunjukkan aliran data yang masuk dan keluar pada suatu proses. Biasanya model ini digambarkan dalam bentuk Diagram Arus Data (*Data Flow*



- Diagram/DFD*). DFD menyajikan gambaran apa yang manusia, proses dan prosedur lakukan untuk mentransformasi data menjadi informasi.
3. **Desain perangkat lunak** adalah tugas, tahapan atau aktivitas yang difokuskan pada spesifikasi detil dari solusi berbasis *computer* (Whitten et al, 2004). Desain perangkat lunak sering juga disebut sebagai *physical design*. Jika tahapan analisis sistem menekankan pada masalah bisnis (*business rule*), maka sebaliknya desain perangkat lunak fokus pada sisi teknis dan implementasi sebuah perangkat lunak (Whitten et al, 2004). Output utama dari tahapan desain perangkat lunak adalah spesifikasi desain. Spesifikasi ini meliputi spesifikasi desain umum yang akan disampaikan kepada stakeholder sistem dan spesifikasi desain rinci yang akan digunakan pada tahap implementasi. Spesifikasi desain umum hanya berisi gambaran umum agar *stakeholder* sistem mengerti akan seperti apa perangkat lunak yang akan dibangun. Biasanya diagram USD tentang perangkat lunak yang baru merupakan point penting dibagian ini. Spesifikasi desain rinci atau kadang disebut desain arsitektur rinci perangkat lunak diperlukan untuk merancang sistem sehingga memiliki konstruksi yang baik, proses pengolahan data yang tepat dan akurat, bernilai, memiliki aspek *user friendly* dan memiliki dasar-dasar untuk pengembangan selanjutnya. Desain arsitektur ini terdiri dari desain *database*, desain proses, desain *user interface* yang mencakup desain *input*, *output form* dan *report*, desain *hardware*, *software* dan jaringan. Desain proses merupakan kelanjutan dari pemodelan proses yang dilakukan pada tahapan analisis.
  4. **Konstruksi** adalah tahapan menerjemahkan hasil desain logis dan

fisik ke dalam kode-kode program komputer.

5. **Pengujian** sistem melibatkan semua kelompok pengguna yang telah direncanakan pada tahap sebelumnya. Pengujian tingkat penerimaan terhadap perangkat lunak akan berakhir ketika dirasa semua kelompok pengguna menyatakan bisa menerima perangkat lunak tersebut berdasarkan kriteria-kriteria yang telah ditetapkan.
6. **Perawatan dan Konfigurasi**. Ketika sebuah perangkat lunak telah dianggap layak untuk dijalankan, maka tahapan baru menjadi muncul yaitu perawatan perangkat lunak. Ada beberapa tipe perawatan yang biasa dikenal dalam dunia perangkat lunak seperti terlihat pada diagram di gambar di bawah ini :

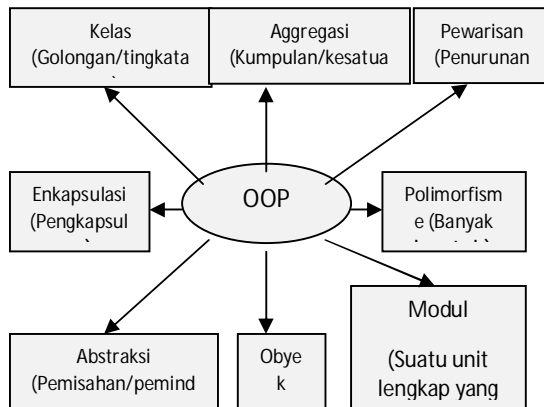


Gambar 2.2 Tipe-tipe perawatan.

- a. Tipe perawatan *corrective* dilakukan jika terjadi kesalahan atau biasa dikenal sebagai *bugs*. Perawatan bisa dilakukan dengan memperbaiki kode program, menambah bagian yang dirasa perlu atau malah menghilangkan bagian-bagian tertentu.
- b. Tipe perawatan *routine* biasa juga disebut *preventive maintenance* dilakukan secara rutin untuk melihat kinerja perangkat lunak ada atau tidak ada kesalahan.
- c. Tipe perawatan *upgrade system* dilakukan jika ada perubahan dari komponen-komponen yang terlibat dalam perangkat lunak tersebut.

Sebagai contoh perubahan *platform* sistem operasi dari versi lama ke versi baru menyebabkan perangkat lunak harus di-*upgrade*.

### 2.3 Object Oriented Programming



Gambar 2.3 Konsep Dasar OOP.

Pemrograman berorientasi obyek (*Object Oriented Programming/OOP*) merupakan paradigma pemrograman yang berorientasikan kepada obyek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam kelas-kelas atau obyek-obyek. Bandingkan dengan logika pemrograman terstruktur, setiap obyek dapat menerima pesan, memproses data, dan mengirim pesan ke obyek lainnya.

Model data berorientasi obyek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik piranti lunak skala besar. Lebih jauh lagi, pendukung OOP mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat.

Dalam kehidupan sehari-hari obyek dinyatakan dengan apa yang kita lihat dan kita rasakan dengan panca indra kita yaitu penglihatan, pendengaran, penciuman dan perasaan (diraba) sudah tentu saja sebuah obyek dalam keseharian tersebut sangatlah berbeda dari obyek satu dengan obyek lainnya yang dirasakan oleh panca indra

kita. Dan obyek-obyek tersebut dapat untuk masa yang akan datang digabungkan dengan obyek lainnya dengan sesuai kebutuhan yang kita inginkan dan dapat dipergunakan sebagai obyek yang sangat berguna untuk sekarang serta dapat dikembangkan untuk masa yang akan datang.

Pemrograman berorientasi obyek diciptakan dan dikembangkan untuk mempermudah dalam pengembangan program dengan cara mengikuti model atau obyek yang telah ada dalam kehidupan nyata. Dalam paradigma ini, sesuai dengan model kehidupan nyata, segala bagian dari suatu permasalahan adalah obyek. Obyek-obyek tersebut juga dapat merupakan penggabungan-penggabungan dari beberapa obyek yang lebih kecil. Sebagai contohnya saya gunakan pada sebuah obyek Sepeda Motor (yang merupakan bagian dari obyek di kehidupan nyata) yang merupakan gabungan dari beberapa bagian obyek yang fungsi, bentuknya yang sesuai dengan kebutuhan pada sebuah sepeda motor, seperti : roda, jok, mesin, rangka motor, dan lain sebagainya.

Begitu juga pada sebuah program. Obyek besar dapat dibentuk dengan menggabungkan beberapa obyek-obyek dalam bahasa pemrograman. Obyek-obyek tersebut berkomunikasi dengan saling mengirim pesan kepada obyek lain yang saling membutuhkan satu dengan yang lainnya.

Khusus pada pemrograman dalam bahasa Java yang merupakan bahasa pemrograman yang berorientasi dengan konsep OOP yang dikembangkan oleh bahasa pemrograman terdahulunya yaitu pemrograman bahasa C. Sebagaimana bahasa yang memiliki konsep OOP memiliki karakteristik yang sangat terinci dengan sifat dari bahasa pemrograman berkonsep OOP pada pemrograman Java yang mendasar.

**2.4 Bahasan Teori Tentang Permasalahan yang Dibahas**

**2.4.1 Faktor-Faktor Perencanaan Jalan Raya**

Faktor-faktor yang mempengaruhi fungsi pelayanan konstruksi jalan :

1. Fungsi dan kelas jalan
2. Kinerja Perkerasan
3. Umur Rencana
4. Beban Lalu lintas
5. Sifat dan daya dukung Tanah dasar
6. Kondisi Lingkungan
7. Sifat dan ketersediaan bahan konstruksi jalan
8. Bentuk geometrik jalan

**2.4.2 Kinerja Perkerasan Jalan**

Wujud perkerasan dan fungsi pelayanan umumnya satu kesatuan yang digambarkan dengan “Kenyamanan mengemudi (*riding quality*)”

1. Keamanan, ditentukan berdasarkan gesekan akibat adanya kontak antara ban dan permukaan jalan
2. Wujud Perkerasan
3. Fungsi pelayanan, Kinerja perkerasan dapat dinyatakan dengan :
  - a. Indeks permukaan/serviceability index

**Tabel 2.3 Nilai Indeks Permukaan**

Indeks Permukaan (IP)	Fungsi Pelayanan
4 -5	Sangat baik
3-4	Baik
2 -3	Cukup
1-2	Kurang
0 -1	Sangat Jelek

- b. Indeks kondisi jalan/road condition index

**Tabel 2.4 Nilai Indeks Kondisi Jalan**

RCI	Kondisi permukaan jalan secara visual
8-10	Sangat rata dan teratur
7-8	Sangat baik, umumnya rata
6-7	Baik
5-6	Cukup, sedikit sekali atau tidak ada lubang, tetapi permukaan jalan tidak rata
4-5	Jelek, kadang-kadang ada lubang, permukaan jalan tidak rata
3-4	Rusak, bergelombang, banyak lubang
2-3	Rusak berat, banyak lubang dan seluruh daerah perkerasan hancur

**3. METODE PENELITIAN**

Dalam penyusunan penelitian ini penulis menggunakan metode pendekatan deskriptif, dengan cara menggambarkan alur sistem, dan menginterpretasikan hasil penelitian berdasarkan kondisi yang sebenarnya di lapangan. Untuk mendapatkan data penunjang dalam penyusunan Penelitian ini penulis menggunakan teknik pengumpulan data sebagai berikut:

- a. Observasi: Melakukan pengamatan dan ikut aktif dalam pengujian tanah secara langsung, baik di lapangan ataupun di laboratorium.
- b. Wawancara (Interview): Dalam proses pengumpulan data penulis melakukan wawancara terhadap pihak yang terkait yang berada di lingkungan Politeknik Negeri Bandung, yaitu Dosen Jalan Raya, Dosen Mekanika Tanah, Dosen Uji Bahan, Laboran Laboratorium Uji Tanah, Laboran Laboratorium Uji Bahan.
- c. Media Dokumenter: Penulis mempelajari data-data dan sumber-sumber referensi yang diberikan oleh Dosen Jalan Raya, Dosen Mekanika Tanah, Dosen Uji Bahan, untuk

dijadikan sebagai tolak ukur standar dalam penyusunan komponen-komponen Perencanaan Jalan Raya. Penulis membuat data-data dokumenter seperti Foto dan Video untuk dijadikan sebagai analisa tambahan dalam pertimbangan-pertimbangan penyusunan perangkat lunak.

- d. Studi Pustaka: Pengumpulan data dilakukan dengan mempelajari buku-buku pustaka, jurnal, e-book, artikel, dan data sheet sebagai referensi dalam penulisan laporan dan pembuatan aplikasi pengolahan data ini.

Model pembangun perangkat lunak yang digunakan pada Penelitian ini menggunakan model pendekatan Object Oriented Programming.

### 3.1 Analisis

Analisis adalah tahap pendefinisian kebutuhan sistem perangkat lunak dan pengguna. Analisis ini penting dan harus dilakukan karena merupakan bagian dari tahap pembangunan perangkat lunak. Dalam proses pembangunan perangkat lunak diperlukan tahapan analisis yang meliputi analisis kebutuhan data, kebutuhan antarmuka, *Use Case Diagram*, analisis kebutuhan fungsional dan analisis perangkat lunak berbasis *Object Oriented Programming* (OOP). Analisis kebutuhan data menjadi bagian yang sangat penting karena perangkat lunak yang dibangun membutuhkan data yang asli, relevan, lengkap dan akurat sehingga pada saat pengolahan data, baik yang diterima maupun yang dihasilkan oleh perangkat lunak dapat diproses dengan benar.

### 3.2 Use Case Diagram

*Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem.

Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. *Use case diagram* dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem.

Sebuah *use case* dapat meng-include fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-include akan dipanggil setiap kali *use case* yang meng-include dieksekusi secara normal. Sebuah *use case* dapat di-include oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-extend *use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

### 3.3 Activity Diagram

Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya (internal processing). Oleh karena itu activity diagram tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

Sama seperti state, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. Decision digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (fork dan join) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal.

Activity diagram dapat dibagi menjadi beberapa object swimlane untuk menggambarkan obyek mana yang bertanggung jawab untuk aktivitas tertentu.

### 3.4 Sequence Diagram

Sequence diagram menggambarkan interaksi antar obyek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu. Sequence diagram terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (obyek-obyek yang terkait).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.

Masing-masing obyek, termasuk aktor, memiliki lifeline vertikal. Message digambarkan sebagai garis berpanah dari satu obyek ke obyek lainnya. Pada fase desain berikutnya, message akan dipetakan menjadi operasi/metoda dari class. Activation bar menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah message. Untuk obyek-obyek yang memiliki sifat khusus, standar UML mendefinisikan icon khusus untuk obyek boundary, controller dan persistent entity.

### 3.5 Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah obyek dan merupakan inti dari pengembangan dan desain berorientasi obyek. Class menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Class diagram menggambarkan struktur dan dePenelitian class, package dan obyek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain.

- Class memiliki tiga area pokok :
  - a. Nama (dan stereotype)
  - b. Atribut
  - c. Metoda
- Atribut dan metoda dapat memiliki salah satu sifat berikut :
  - a. Private, tidak dapat dipanggil dari luar class yang bersangkutan
  - b. Protected, hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak yang mewarisinya
  - c. Public, dapat dipanggil oleh siapa saja

Class dapat merupakan implementasi dari sebuah interface, yaitu class abstrak yang hanya memiliki metoda. Interface tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah class. Dengan demikian interface mendukung resolusi metoda pada saat run-time. Sesuai dengan perkembangan class model, class dapat dikelompokkan menjadi package. Kita juga dapat membuat diagram yang terdiri atas package.

Hubungan Antar Class :

1. Asosiasi, yaitu hubungan statis antar class. Umumnya menggambarkan class yang memiliki atribut berupa class lain, atau class yang harus mengetahui eksistensi class lain. Panah navigability menunjukkan arah query antar class.
2. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas..”).

3. Pewarisan, yaitu hubungan hirarkis antar class. Class dapat diturunkan dari class lain dan mewarisi semua atribut dan metoda class asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari class yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.
4. Hubungan dinamis, yaitu rangkaian pesan (message) yang di-passing dari satu class kepada class lain. Hubungan dinamis dapat digambarkan dengan menggunakan sequence diagram yang telah dijelaskan sebelumnya.

#### 4. HASIL DAN PEMBAHASAN

##### 4.1 Batasan Sistem

Implementasi sistem adalah prosedur yang dilakukan untuk menyelesaikan desain sistem yang ada dalam dokumen sistem yang disetujui dan menguji, menginstal, dan memulai menggunakan sistem yang baru atau sistem yang diperbaiki.

Adapun tujuan-tujuan dari implementasi sistem adalah sebagai berikut :

1. Menyelesaikan desain sistem yang ada dalam dokumentasi desain sistem yang telah disetujui.
2. Menulis, menguji dan mendokumentasikan program-program dan prosedur-prosedur yang diperlukan oleh desain sistem yang disetujui.
3. Memastikan bahwa *user* dapat mengoperasikan sistem yang baru yaitu dengan mempersiapkan pemakaian manual.
4. Memperhitungkan bahwa sistem memenuhi permintaan pemakai yaitu dengan menguji sistem secara keseluruhan.
5. Memastikan bahwa konversi kesistem yang baru berjalan secara benar yaitu dengan merencanakan, mengontrol dan melakukan instalasi sistem yang baru secara benar.

Dari pengertian dan tujuan-tujuan implementasi yang telah dijelaskan diatas maka ada beberapa langkah-langkah

implementasi yang dilakukan dalam menyelesaikan rancangan sistem. Adapun langkah-langkah yang dibutuhkan dalam mengimplementasikan sistem ini adalah sebagai berikut:

1. Menyelesaikan desain sistem  
Penulis membuat *Use Case Diagram*, program, model *input* dan *output* untuk pengolahan Data Tanah hingga Perencanaan Perkerasan Jalan dan Perbaikan Jalan.
2. Menyediakan Perangkat Keras (*Hardware*) dan Perangkat Lunak (*Software*)  
Tahapan ini merupakan tahapan menyediakan segala perangkat lunak seperti sistem operasi komputer, bahasa pemrograman yang digunakan dan perangkat keras yang dibutuhkan, diantaranya adalah CD ROM (*Compact Disc Read Only Memory*) atau Flash Disk (*Removeable Drive*) sebagai media untuk penginstalan sistem operasi ataupun perangkat lunak lainnya, dan juga sebagai peralatan pemasukan data-data (*input device*) yang dibutuhkan agar komputer dapat dioperasikan dengan baik dan pelaksanaan pemrosesan data oleh program yang dibuat dapat terlaksana dengan baik.
3. Menuliskan Program Kedalam Komputer  
Pada saat perangkat lunak (*software*) dan perangkat keras (*hardware*) komputer telah dipersiapkan, maka penulis akan menyalin atau mengetikkan instruksi-instruksi (*listing*) rancangan sistem kekomputer sesuai dengan bahasa pemrograman yang digunakan yaitu Microsoft Visual Basic 6.0.
4. Mendapat Pengarahan  
Pengarahan dari pihak yang akan memakai sistem yang dirancang, yaitu bidang-bidang pekerjaan yang berhubungan dengan perencanaan jalan raya.
5. Menguji Sistem  
Tahapan ini merupakan suatu langkah yang ditujukan untuk mengevaluasi,

- apakah sistem yang telah dibuat sesuai dan memenuhi prosedur-prosedur yang telah ditetapkan.
6. Melakukan Bimbingan Terhadap Langkah-langkah Pengoperasian Sistem Pada Tahapan ini, dilakukan beberapa panduan tentang pengoperasian sistem kepada user yang akan menggunakannya.
  7. Memilih dan Melatih User Tahapan ini akan dikerjakan apabila diperlukan, yaitu memilih dan melatih personel atau karyawan agar dapat menguasai penggunaan sistem yang sudah dirancang.
  8. Perawatan Sistem
  9. Setelah sistem baru dipakai oleh user dalam kegiatan perencanaan jalan, maka sistem akan terus dievaluasi untuk memperkecil kesalahan/kerusakan sistem.

**4.2 Pembuatan Antar Muka**

1. Tampilan *Login*  
Tampilan *login* ini berfungsi untuk mengidentifikasi *user* yang akan memasuki sistem, dimana data *user* akan disimpan sementara pada *registry windows* dalam bentuk data terpenelitian, dan *registry user* akan otomatis dihapus pada saat *user* keluar dari sistem.



Gambar 4.1 Tampilan *Login*

2. Tampilan *Menu Utama*  
Tampilan Menu Utama akan muncul pada saat sistem telah memvalidasi data *user* pada saat *login*. Dimana hak akses

menu akan sesuai dengan tipe *user* yang datanya telah tersimpan dalam *registry*.

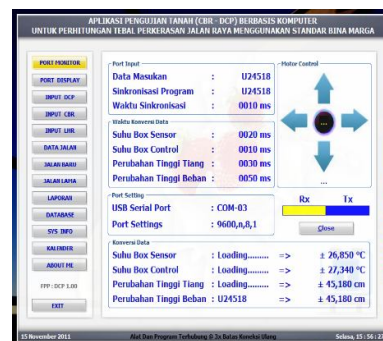


Gambar 4.2 Tampilan Menu Utama



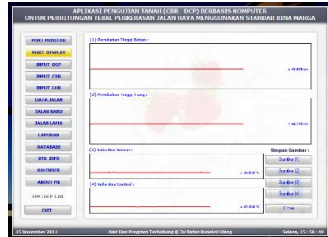
Gambar 4.3 Tampilan Identifikasi *User* pada *Menu Utama*

3. Tampilan *Port Monitor*  
Tampilan untuk memonitor *input data* yang masuk dari microcontrol ke port komputer.



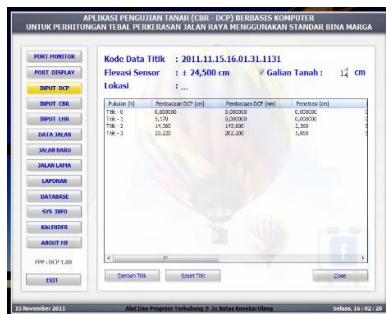
Gambar 4.4 Tampilan *Port Monitor*

4. Tampilan *Port Display*  
 Grafik *history port input* untuk mempermudah dalam pengecekan.



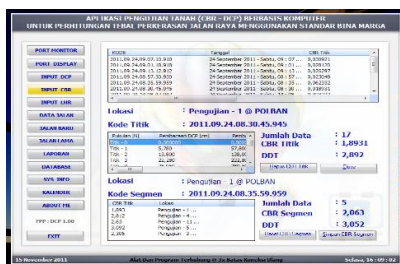
Gambar 4.5 Tampilan *Port Display*

5. Tampilan *Input DCP*  
 Tampilan untuk merekan data elevasi yang dikirim dari alat.



Gambar 4.6 Tampilan *Input DCP*

6. Tampilan *Input CBR*  
 Tampilan untuk kalkulasi data dari CBR titik ke CBR Segmen.



Gambar 4.7 Tampilan *Input CBR*

7. Tampilan *Input LHR*  
 Tampilan untuk *input* data jumlah kendaraan dari hasil *survey* lapangan.



Gambar 4.8 Tampilan *Input LHR*

8. Tampilan (*Input*) Data Jala  
 Tampilan untuk *input* data kondisi lingkungan proyek dari hasil *survey* lapangan.



Gambar 4.9 Tampilan *Input Data Jalan*

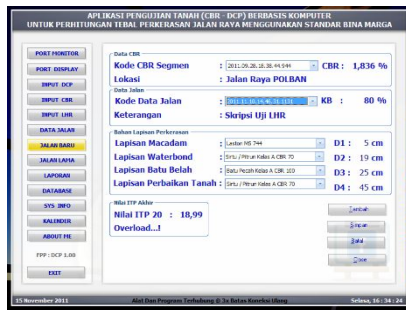
9. Tampilan (Perhitungan Perencanaan) Jalan Baru  
 a. Tampilan untuk menghitung perencanaan tebal perkerasan jalan Hotmix.



Gambar 4.10 Perhitungan Perencanaan Jalan Hotmix

- b. Tampilan untuk menghitung perencanaan tebal perkerasan jalan Macadam.





Gambar 4.11 Perhitungan Perencanaan Jalan Macadam

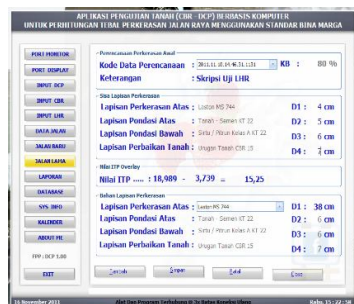
c. Tampilan untuk menghitung perencanaan tebal perkerasan jalan Rigid.



Gambar 4.12 Perhitungan Perencanaan Jalan Rigid

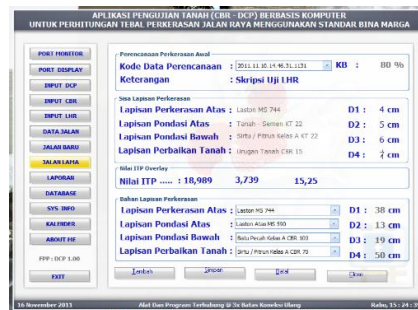
10. Tampilan (Perhitungan Perbaikan) Jalan Lama

a. Tampilan untuk menghitung perbaikan Perkerasan Atas jalan Hotmix.



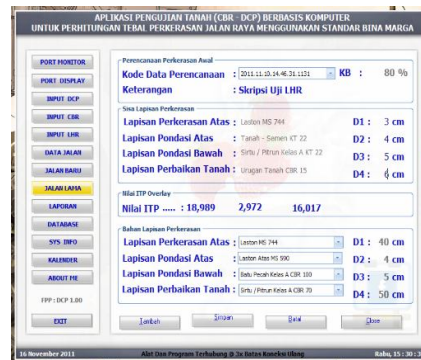
Gambar 4.13 Perhitungan Overlay Jalan Hotmix

b. Tampilan untuk menghitung perbaikan dan pelebaran jalan Hotmix.



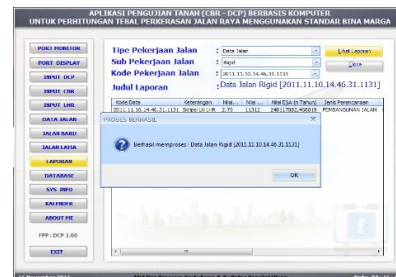
Gambar 4.14 Perhitungan Peningkatan Jalan Hotmix

c. Peningkatan Macadam Tampilan untuk menghitung perbaikan dan pelebaran jalan Macadam.



Gambar 4.15 Perhitungan Peningkatan Jalan Macadam

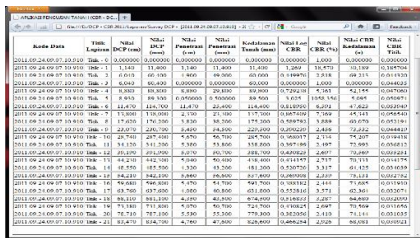
11. Tampilan (Pembuatan) Laporan Tampilan untuk melihat laporan data-data dari mulai perencanaan hingga perhitungan, yang dapat di-generate menjadi tampilan dalam browser.



Gambar 4.16 Jendela Laporan Data Jalan Rigid



Gambar 4.17 Hasil Laporan Data Jalan Rigid

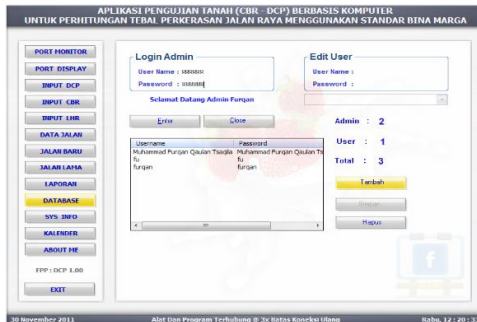


Gambar 4.18 Hasil Laporan Data Survey DCP Pada Satu Titik

12. Tampilan (*Edit*) Database  
Tampilan untuk menghapus data-data yang salah atau tidak diperlukan. Untuk data *user*, selain tombol hapus, terdapat juga tombol tambah, edit, dan reset.

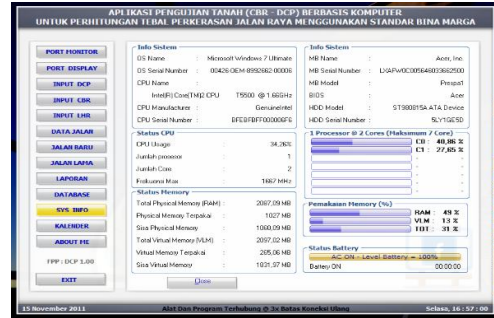


Gambar 4.19 Edit Database



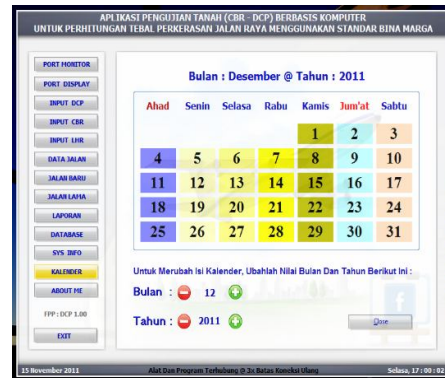
Gambar 4.20 Edit User Login

13. Tampilan Sistem Info  
Tampilan yang menginformasikan perangkat komputer yang sedang dipakai.



Gambar 4.21 Tampilan Informasi Sistem

14. Tampilan Kalender  
Tampilan yang menginformasikan tanggal masa lalu, sekarang, dan masa depan.



Gambar 4.22 Tampilan Kalender

15. Tampilan About Me  
Tampilan yang menampilkan informasi umum dari penyusun perangkat lunak.



Gambar 4.23 Tampilan About Me

## 5. KESIMPULAN

Permasalahan dalam penyusunan Penelitian ini seperti terlampirkan pada identifikasi masalah di bab 1, akhirnya dapat penulis implementasikan dalam bentuk perangkat lunak yang dapat menjawab permasalahan-permasalahan yang menjadi dasar paling utama, yaitu dengan membangun perangkat lunak yang dapat menunjang 3 bidang penting perencanaan jalan menjadi satu bagian yang saling terhubung, sehingga diharapkan efisiensi waktu dan biaya dapat diwujudkan oleh perangkat lunak ini.

Desain perangkat lunak yang berbasis Graphical User Interface pada perangkat lunak ini diharapkan makin mempermudah peng-input-an, pemrosesan, hingga pelaporan data untuk menunjang perencanaan lain dalam pekerjaan jalan, karena pemrosesan perangkat lunak ini di desain secara terstruktur yang salah satunya adalah dengan cara menyusun menu sesuai urutan pekerjaan perencanaan perkerasan jalan. Kemudian efek-efek pada tampilan, diharapkan tidak membuat pengguna menjadi jenuh meskipun harus bekerja lama dihadapan perangkat lunak ini.

Kesederhanaan dan efek-efek pada tampilan untuk mempengaruhi kenyamanan psikologis pemakai perangkat lunak ini, didukung dengan kemampuan perangkat lunak untuk memproses data dengan akurasi dan kecepatan yang optimal menggunakan sistem trial and error, sehingga istilah "Biarkan program yang bekerja, silahkan user bersantai-santai sambil menunggu hasil" telah diterapkan pada perangkat lunak ini. Perhitungan-perhitungan yang rumit dan uji coba data hingga 25.000 kali, dan penambahan faktor koreksi terhadap standar deviasi kesalahan perhitungan dengan ketelitian  $\pm 0.001$  telah diterapkan pada perangkat lunak ini. Sehingga hasil akhir keluaran dari perhitungan perangkat lunak ini mendekati hasil akhir cara perhitungan manual perkerasan jalan menggunakan standar Bina Marga, yang disertai dengan

penambahan koreksi terhadap faktor error deviation perhitungan.

Kelebihan yang paling utama dari perangkat lunak yang dibangun adalah, efisiensi waktu dan biaya. Karena jika perhitungan perkerasan dilakukan secara manual, mulai dari perhitungan daya dukung tanah, analisa bahan, dan perancangan perkerasan akan memerlukan waktu antara 1 hingga 3 bulan. Sedangkan perangkat lunak yang dibangun dapat melakukan perhitungan antara 5-10 menit. Dengan catatan bahwa perbandingan yang dilakukan hanya dalam perhitungan saja, tanpa mempertimbangkan aktivitas survey dilapangan. Hal ini ditinjau dari salah satu efisiensi waktu perhitungan, yaitu perhitungan data tanah secara manual saat perhitungan nilai CBR Titik hingga mendapatkan nilai CBR Segmen oleh mahasiswa Politeknik Negeri Bandung. Pada umumnya jika dalam aktivitas normal, mereka memerlukan waktu 3 hari hingga 1 minggu untuk melakukan perhitungan data tanah hingga menjadi nilai CBR Segmen yang akan dipakai dalam perencanaan jalan raya.

## 6. REFERENSI

- Bahrami. 2000. Object Oriented System Development, Mac Graw-Hill.
- Barker, Richard M & Puckett, Jay A. 1997. Design of Highway Bridges. Wiley Interscience.
- Booch, G., J., Rumbaugh, Jacobson. 1999. The Unified Modeling Language. User Guide, Rational Software Corporation. Addison-Wesley. Massachusetts, USA.
- Davis, M., Alan. 1993. Software Requirement Object, Functions, and States. Fifth Edition. Prentice Hall.
- Departemen Pekerjaan Umum. 1980. Manual Penyelidikan Geoteknik Untuk Perencanaan Pondasi Jembatan. Yayasan Padamu Negeri. Jakarta.
- Departemen Pekerjaan Umum Direktorat Jenderal Bina Marga. 1983. Pedoman

- Penentuan Tebal Perkerasan Lentur Jalan Raya. Jakarta.
- Departemen Pekerjaan Umum Direktorat Jenderal Bina Marga. 1985. Petunjuk Perencanaan Perkerasan Kaku (Beton Semen). Jakarta.
- Departemen Pekerjaan Umum. 1987. Petunjuk Perencanaan Tebal Perkerasan Lentur Jalan Raya Dengan Metode Analisa Komponen. Jakarta.
- Hendarsin, Shirley, 2000. Petunjuk Praktis Perencanaan Teknik Jalan Raya. Politeknik Negeri Bandung-Jurusan Teknik Sipil. Bandung.
- Indrasurya, B.Mochtar & Sudjanarko, S., 1993. Jalan Raya II, Institut Teknologi Sepuluh Nopember Surabaya-Jurusan Teknik Sipil. Surabaya.
- ITS, Bahan kuliah Rekayasa Jalan Raya, 2005. Institut Teknologi Sepuluh Nopember Surabaya-Jurusan Teknik Sipil. Surabaya.
- Ken Lunn. 2001. Information Modeling and Development. Prentice Hall.
- Mulyanto, Aunur R. 2008. Rekayasa Perangkat Lunak Jilid 1 untuk SMK. Direktorat Pembinaan Sekolah Menengah Kejuruan, Direktorat Jenderal Manajemen Pendidikan Dasar dan Menengah, Departemen Pendidikan Nasional. Jakarta.
- Pressman, Roger, S. 2000. Software Engineering A Practitioner's Approach. Fifth Edition. MacGraw-Hill.
- Samorville, Ian. 1996. Software Engineering. Third Edition. Prentice Hall.
- Sukirman, Silvia, 1999. Perkerasan Lentur Jalan Raya. Penerbit Nova. Bandung.