**JURNAL**

**APLIKASI PENJADWALAN KEJURNAS PENCAKSILAT DENGAN MENGGUNAKAN ALGORITMA PENELUSURAN POHON**



Disusun oleh :

**Septian Dwi Wijayanto**

**123060039**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INDUSTRI**

**UPN “VETERAN” YOGYAKARTA**

**2013**

**APLIKASI PENJADWALAN KEJURNAS PENCAKSILAT DENGAN MENGGUNAKAN ALGORITMA PENELUSURAN POHON**

Septian Dwi Wijayanto (Mahasiswa)

Heru Cahya Rustamaji, S.Si., M.T, Paryati., S.T., M.Kom. (Dosen)

Universitas Pembangunan Nasional “ Veteran” Yogyakarta

Fakultas Teknologi Industri Jurusan Teknik Informatika

JL.Babarsari No.2 Yogyakarta Teknik Informatika 55281

e-mail : arcapada182@yahoo.com

Abstrak

Pencak silat sebagai warisan budaya Indonesia berkembang sesuai dengan perkembangan zaman. Pada era teknologi sekarang, pencak silat dapat beradaptasi dengan perkembangan teknologi yang sangat pesat. Dalam hal ini pemanfaatan teknologi untuk bidang olah raga pencak silat salah satunya adalah pemanfaatan pembuatan jadwal kejuarnas. Pada suatu kejurnas pengolahan data merupakan elemen penting dalam kelancaran kegiatan, dari pendaftaran, pengundian, maupun pembuatan jadwal. Sebuah aplikasi dibutuhkan untuk pengolahan data kejurnas dari pendataan peserta, pendataan manajer, dan pendataan *official*. Data dari peserta dapat langsung dijadikan sebagai data dalam melakukan undian untuk pertandingan kategori tanding dan kategori seni. Kemudian sistem penjadwalan kejurnas menghasilkan jadwal dari babak penyisihan sampai babak final dengan metode algoritmma penelusuran pohon.

Pada skripsi ini telah dapat dibuat perangkat lunak penjadwalan kejurnas berbasis *object oriented* dengan menggunakan algoritma penelusuran pohon. Metode yang digunakan dalam perancangan dan pembuatan perangkat lunak ini adalah metode *grapple*. Perangkat lunak ini dibuat dengan menggunakan bahas pemrograman java dan sistem manajemen database MySQL.

. Program penelusuran pohon ini dapat menghasilkan penjadwalan kejurnas pencak silat dengan jadwal terurut dan tidak ada bentrok pertandingan pada waktu yang sama.

.

1. **Pendahuluan**

Pencak silat adalah salah satu budaya bangsa Indonesia yang mempunyai unsur seni, mental dan spiritual. Perkembangan pencak silat sekarang sangat pesat hingga mencapai mancanegara. Pencak silat di Indonesia sangat beragam jenis perguruannya, sehingga di Indonesia berdiri organisasi yang menaungi berbagai macam perguruan bernama IPSI (Ikatan Pencak Silat Seluruh Indonesia). Dalam program IPSI banyak diadakan kejuaraan - kejuaraan, baik itu lingkup daerah, nasional maupun internasional. Salah satu ajang yang bergensi di Indonesia adalah ajang Kejurnas Pencak Silat diadakan rutin di Universitas Pembangunan Nasional “veteran” Yogyakarta setiap 2 tahun sekali, dimana peserta adalah umum dan nasional dari sabang sampai merauke. Kelas yang dipertandingan adalah katagori tanding yang terdiri dari kelas A – J untuk putra dan A - F untuk kelas putri, dan katagori seni yang terdiri dari seni tuggal, seni ganda , dan seni beregu baik putra maupun putri.

Dalam setiap ajang kejuaraan tentunya ada pengolahan data dari pendaftaran, pengelompokan kelas dan pengundian jadwal pertandingan. Pengolahan data saat ini kebanyakan masih menggunakan konsep manual, yaitu masih memasukkan data dari pendaftaran, *technical meeting* (pengundian) dan pembuatan jadwal pertandingan dengan cara menulis dikertas. Cara ini membutuhkan waktu lama untuk menyelesaikannya, karena *technical meeting* tidak akan jalan apabila input daftar peserta dari pendaftaran belum selesai dan jadwal belum bisa dibuat apabila belum ada pengundian.

Untuk itu dibuat aplikasi dimana bisa memasukkan data secara komputerisasi, sehingga data akan cepat masuk dan cepat menghasilkan data base peserta, nantinya data tersebut langsung digunakan untuk pengundian dan langsung diproses menjadi jadwal pertandingan dari babak penyisihan sampai babak final dalam sekali waktu dan cepat dalam hitungan detik. Penggunaaan algoritma penelusuran pohon lebih dapat diterapkan karena sistem kompetisi kejurnas pencak silat menggunakan sistem gugur.

**Dasar Teori**

**2.1. Sistem**

Sistem merupakan kumpulan dari elemen-elemen atau komponen-komponen yang berinteraksi untuk mencapai tujuan tertentu. Komponen atau elemen tersebut tidak dapat lepas sendiri-sendiri. Subsistem-subsistem tersebut saling berinteraksi dan saling berhubungan membentuk satu kesatuan sehingga tujuan atau sasaran sistem dapat tercapai (Hartono, 2002).

Suatu sistem memiliki karakteristik atau sifat tertentu, yaitu mempunyai komponen-komponen (*components*) yang dapat berupa suatu subsistem atau bagian-bagian dari sistem, batasan sistem (*boundary*) yang merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya, lingkungan luar sistem (*environment*) yaitu apapun di luar batas dari sistem yang mempengaruhi operasi sistem, penghubung (*interface*) yang merupakan penghubung antara satu subsistem dengan subsistem lainnya, masukan (*input*) yaitu energy yang dimasukkan ke dalam sistem, keluaran (*output*) merupakan hasil dari energy yang diolah, pengolah (*process*) yang akan mengubah masukan menjadi keluaran, dan sasaran (*goal*) yang menentukan masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatkan berhasil bila mengenai sasaran atau tujuannya. Gambaran karakteristik dari sistem dapat dilihat pada gambar 2.1.



**Gambar 2.1** Karakteristik Sistem

**2.2. *Software***

*Software* atau perangkat lunak adalah kumpulan perintah-perintah yang dieksekusi dalam menjalankan mesin, biasanya disebut juga dengan program. Tanpa *software* proses komputerisasi tidak mungkin berjalan. Dua kelompok *software* adalah *software* *Operating System* (OS) dan *Application* *Software* (AS).

OS dibuat untuk mengontrol program seperti sistem operasi dan *Database Management System* (DBMS). AS adalah program yang dapat memproses data bagi user seperti *Inventory*, *Payroll*, *Spreadsheet*, *Word Processor*, dan lain-lain. Suatu kesalahpahaman yang sering terjadi, *software* disamakan dengan data, padahal bukan. *Software* memerintahkan hardware untuk memproses data. *Software* dijalankan, sedangkan data diproses.

*Software* digunakan untuk menghasilkan (*producing*), mengelola (*managing*), memperoleh (*acquiring*), mengubah (*modifying*), memperagakan (*displaying*), atau mengirimkan (*transmitting*) informasi. Informasi ini dapat sesederhana satu *bit* sampai kepada persentasi ke dalam bentuk multimedia.

*Software* memiliki dua peranan yaitu sebagai produk dan juga sekaligus sebagai alat atau wahana untuk menghasilkan produk. Sebagai sebuah produk, dia memberikan komputasi yang dikandung oleh *hardware* (perangkat keras), baik itu berupa sebuah komputer maupun sebuah jaringan komputer. Sebagai sebuah alat untuk menghasilkan produk, *software* membantu untuk mengendalikan komputer (misal *operating system*), membentuk jaringan komunikasi, dan alat (*tools*) untuk menghasilkan program (*development tools*). *Software* terdiri atas program, data, dan dokumentasi.

**2.3 Pengertian *Database***

Kumpulan dari item data yang saling berhubungan satu dengan yang lainnya yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, tersimpan di hardware komputer dan dengan software untuk melakukan manipulasi untuk kegunaan tertentu (Hartono, 1995). Dalam satu file atau table terdapat *record-record* yang sejenis, sama besar, sama bentuk, yang merupakan satu kumpulan entitas yang seragam. Satu record (umumnya digambarkan sebagai baris data) terdiri dari field yang saling berhubungan menunjukan bahwa field tersebut dalam satu pengertian yang lengkap dan disimpan dalam satu record.

Adapun Struktur Database adalah:

1. Database
2. File/Table
3. Record
4. Elemen data / Field

Dari pengertian di atas dapat disimpulkan bahwa *database* mempunyai beberapa kriteria penting, yaitu :

1. Bersifat data oriented dan bukan program oriented.
2. Dapat digunakan oleh beberapa program aplikasi tanpa perlu mengubah *database*nya.
3. Dapat dikembangkan dengan mudah, baik volume maupun strukturnya.
4. Dapat memenuhi kebutuhan sistem-sistem baru secara mudah
5. Dapat digunakan dengan cara-cara yang berbeda.

Prinsip utama database adalah pengaturan data dengan tujuan utama fleksibelitas dan kecepatan pada saat pengambilan data kembali. Adapun ciri-ciri *database* diantaranya adalah sebagai berikut :

1. Efisiensi meliputi kecepatan, ukuran, dan ketepatan
2. Data dalam jumlah besar.
3. Berbagi Pakai (dipakai bersama sama / *sharebility*).
4. Mengurangi bahkan menghilangkan terjadinya duplikasi dan ketidakkonsistenan data.

**2.4 Pengertian UML**

*Unified Modeling Language* (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OO). Definisi ini merupakan definisi yang sederhana. Pada kenyataannya, pendapat orang-orang tentang UML berbeda satu sama lain. Hal ini dikarenakan oleh sejarahnya sendiri dan oleh perbedaan persepsi tentang apa yang membuat sebuah proses rancang – bangun perangkat lunak efektif (Fowler, 2004:p1).

*Unified Modeling Language* (UML) merupakan strandar yang relatif terbuka yang dikontrol oleh Object Management Group (OMG), sebuah konsorsium terbuka yang terdiri dari banyak perusahaan. OMG dibentuk untuk membuat standar – standar yang mendukung interoperabilitas, khusunya interoperabilitas sistem berorientasi objek. OMG mungkin lebih dikenal dengan standar – standar COBRA (Common Object Request Broker Architecture) (Fowler, 2004:p1-2).

UML lahir dari penggabungan banyak bahasa permodelan grafis berorientasi objek yang berkembang pesat pada akhir 1980-an dan awal 1990-an. Sejak kehadirannya pada tahun 1997, UML menghancurkan menara Babel tersebut menjadi sejarah (Fowler, 2004:p2).

Beberapa diagram UML yang digunakan dalam skripsi ini, yaitu:

* Diagram *usecase*
* Diagram *Activity*
* Diagram *sequence*
* Diagram *class*

**2.5 Metode GRAPPLE (*Guidelines for Rapid APPLication Engineering*)**

Metode GRAPPLE atau biasa disebut metode RAD (*Rapid Application Development*) terdiri dari 5 tahapan yaitu :

1. Tahap perencanaan kebutuhan (*requirements gathering*)

2. Tahap analisis (*analysis*)

3. Tahap desain (*design*)

4. Tahap pengembangan (*development*)

5. Tahap penyebaran (*deployment*)

**2.6 Pencak Silat**

Pencak Silat adalah kata majemuk. Pencak dan Silat mempunyai pengertian yang sama dan merupakan bagian dari kebudayaan masyarakat pribumi Asia Tenggara, yakni kelompok masyarakat etnis yang merupakan penduduk asli negara-negara di kawasan Asia Tenggara (Brunei Darussalam, Filipina, Indonesia, Kamboja, Laos, Malaysia, Myanmar, Singapura, Thailand dan Vietnam). Kata Pencak biasa digunakan oleh masyarakat pulau Jawa, Madura dan Bali, sedangkan kata Silat biasa digunakan oleh masyarakat di wilayah Indonesia lainnya maupun di Malaysia, Singapura, Brunei Darussalam serta di Thailand (bagian Selatan) dan Filipina. Penggabungan kata Pencak dan Silat menjadi kata majemuk untuk pertama kalinya dilakukan pada waktu dibentuk suatu organisasi persatuan dari perguruan Pencak dan perguruan Silat di Indonesia yang diberi nama Ikatan Pencak Silat Indonesia, disingkat IPSI pada tahun 1948 di Surakarta. Sejak saat itu Pencak Silat menjadi istilah resmi di Indonesia. Perguruan-perguruan yang mengajarkan Pencak dan Silat asal Indonesia di berbagai negara kemudian juga menggunakan istilah Pencak Silat. Di dunia internasional Pencak Silat menjadi istilah resmi sejak dibentuknya Organisasi Federatif Internasional yang diberi nama Persekutuan Pencak Silat Antarabangsa, disingkat PERSILAT, di Jakarta pada tahun 1980.

* 1. **Sistem Pertandingan Pencak Silat**
1. Kategori Tanding

 Sistem pertandingan pencak silat untuk kategori tanding menggunakan sistem gugur, yaitu sistem dimana peserta yang telah kalah gugur haknya untuk mengikuti pertandingan selanjutnya. Artinya ialah bahwa peserta yang telah kalah tidak diperkenankan lagi untuk bertanding pada babak berikutnya. Yang paling penting harus diperhatikan dalam sistem gugur ini adalah cara membuat skema / bagan pertandingan. Karena apabila skema tersebut tidak benar maka akan ada peserta yang akan diuntungkan dan ada peserta yang dirugikan. Sebelum skema / bagan dibuat, terlebih dahulu ditentukan bagian kelompok yang terdiri dari kelompok atas dan kelompok bawah dari seluruh peserta, dengan ketentuan sebagai berikut :

1. Jika pesertanya genap, maka separuh bagian atas harus sama dengan separuh bagian bawah.
2. Jika pesertanya ganjil, maka separuh bagian atas harus lebih banyak dari bagian bawah.



**Gambar 2.2** Contoh bagan pertandingan kategori tanding pencak silat

Untuk menghitung jumlah partai pertandingan dalam system gugur dioergunakan rumus : Dimana N adalah jumlah peserta dan K adalah kelas yang dipertandingkan.

N - K

1. Kategori Seni

 Sistem pertandingan kategori seni digunakan system pool, yaitu sistem dimana setiap peserta seni dikumpulkan dalam satu kelompok atau biasa disebut pool. Dalam setiap pool dibatasi 7 peserta, apabila lebih dari 7 peserta dalam setiap kelompok seni maka akan dibagi menjadi 2 pool.

**2.8 Pohon**

 Pohon merupakan salah satu penerapan dari teori graf. Konsep pohon sangat penting posisinya baik dalam dunia informasi ataupun dalam bidang lainya. Penerapan pohon banyak diterapkan pada kehidupan sehari hari misalnya silsilah keluarga, struktur organisasi, bagan sistem gugur dalam pertandingan dan lain – lain.

Pohon merupakan *graf* yang memiliki sifat khusus, sehingga dapat memiliki definisi yg mempunyai sifat :

1. Misalkan G merupakan suatu graf dengan n buah simpul dan tepat n – 1 buah sisi. Jika G tidak mempunyai sirkuit maka G merupakan pohon.
2. Suatu pohon dengan n buah simpul mempunyai n – 1 buah sisi.
3. Setiap pasang simpul di dalam suatu pohon terhubung dengan lintasan tunggal.
4. Misalkan G adalah graf sederhana dengan jumlah simpul n, jika G tidak mengandung sirkuit maka penambahan satu sisi pada frat hanya akan membuat satu sirkuit.

 Pada kebanyakan aplikasi pohon, simpul tertentu diperlukan sebagai akar (root). Sekali sebuah simpul ditetapkan sebagai sebuah akar, maka simpul – simpul lainnya dapat dicapai dari akar dengan memeberi arah pada sisi – sisi pohon yang mengikutinya.

* 1. **Algoritma Penelusuran Pohon**

 Algoritma merupakan urutan langkah – langkah logis penyelesaian masalah yang disusun secara sistematis. Algoritma tersebut mencari node dari pohon, terlepas apakah pohon tersebut *eksplisit* atau *implisit* (dibangkitkan saat pengerjaan). Prinsip dasarnya adalah sebuah node diambil dari sebuah struktur data, suksesornya diperiksa dan ditambahkan pada struktur data. Dengan memanipulasi struktur data, pohon dieksplorasi dalam urutan yang berbeda-beda, dieksplore dari satu tingkat ke tingkat berikutnya (pencarian *Breadth-first*) atau mengunjungi node pucuk terlebih dahulu kemudian lacak balik/*backtracking* (pencarian *Depth-first*)

**3. Analisis dan Perancangan**

**3.1 Diagram *Usecase***

Proses pengembangan perangkat lunak berlangsung melalui serangkaian aktivitas yang dapat dimodelkan pada diagram *use case*. Dalam sistem ini terdapat diagram *use case* yang mempunyai dua *user* atau disebut sebagai aktor. Aktor pertama adalah *user* admin, dimana admin dapat melakukan login, melihat report data *core analysis*, dan juga mempunyai hak untuk melakukan manajemen *user* meliputi membuat akun *user* baru, mengedit dan menghapus akun *user* yang sudah ada. Aktor yang kedua adalah *user* penginput, yang dapat melakukan login, melakukan proses penginputan dan pengeditan data, melihat report data, namun tidak mempunyai akses ke manajemen user.



**Gambar 3.1** Diagram *Use Case*

**3.2 Diagram Kelas**

Diagram kelas menggambarkan struktur kelas dari aplikasi. Diagram ini memperlihatkan hubungan antarkelas dan penjelasan detail dari tiap-tiap kelas yang ada pada aplikasi.

**Tabel** 3.1 Tabel Kelas Aplikasi

|  |  |
| --- | --- |
| **Nama Kelas** | **Keterangan** |
| InputPesertaSeni | Kelas untuk menampilkan form pendaftaran peserta seni |
| InputPesertaTanding | Kelas untuk menampilkan form pendaftaran peserta tanding |
| InputManajer | Kelas untuk menampilkan form pendaftaran manajer |
| InputOfficial | Kelas untuk menampilkan form pendaftaran official |
| RekapPesertaPerSeni | Kelas untuk menampilkan form data peserta seni |
| RekapPesertaPerKelas | Kelas untuk menampilkan form data peserta per kelas |
| RekapPesertaPerKontingen | Kelas untuk menampilkan form data peserta per kontingen |
| Bagan | Kelas untuk menampilkan form input bagan |
| RekapPesertaTanding | Kelas untuk menampilkan form data peserta tanding |
| UndianTanding | Kelas untuk menampilkan form undian tanding |
| UndianSeni | Kelas utuk menampilkan form undian seni |
| RekapUndianSeni | Kelas untuk menampilkan form data undian seni |
| RekapUndianTanding | Kelas untuk menampilkan form data undian tanding |
| GenerateJadwalTanding | Kelas untuk menampilkan form generate jadwal tanding |
| RekapJadwalTanding | Kelas untuk menampilkan form jadwal tanding |
| GenerateJadwalSeni | Kelas untuk menampilkan form generate jadwal seni |
| RekapjadwalSeni | Kelas untuk melihat form jadwal seni |
| KoneksiDatabase | Kelas sebagai koneksi database aplikasi |
| Kejurnasview | Kelas untuk menampilan utama aplikasi |
| MTManajer | Kelas tabel model untuk menampilkan data manajer |
| MTOficial | Kelas tabel model untuk menampilkan data oficial |
| MTPartaiTanding | Kelas tabel model untuk menampilkan data partai tanding |
| MTPesertaSeni | Kelas tabel model untuk menampilkan data peserta seni |
| MTPesertaTanding | Kelas tabel model untuk menampilkan data peserta tanding |
| MTRekapKelas | Kelas tabel model untuk menampilkan data kelas |
| MTRekapKontingen | Kelas tabel model untuk menampilkan data kontingen |
| MTRekapSeni | Kelas tabel model untuk menampilkan data peserta seni |
| MTRekapUndianSeni | Kelas tabel model untuk menampilkan data undian seni |
| MTRekapUndianTanding | Kelas tabel model untuk menampilkan data undian tanding |
| MTJadwalTanding | Kelas tabel model untuk menampilkan data jadwal tanding |
| MTJadwalSeni | Kelas tabel model untuk menampilkan data jadwal seni |
| MTUndianTanding | Kelas tabel model untuk menampilkan data undian tanding |
| MTUndianSeni | Kelas tabel model untuk menampilkan tabel undian seni |
| MTbagan | Kelas tabel model untuk menampilkan tabel bagan |
| MTBManajer | Kelas untuk menyimpan data tabel manajer |
| MTBOficial | Kelas untuk menyimpan data tabel oficial |
| MTBPartaiTanding | Kelas untuk menyimpan data tabel partai tanding |
| MTBPesertaSeni | Kelas untuk menyimpan data tabel peserta seni |
| MTBPesertaTanding | Kelas untuk menyimpan data tabel peserta tanding |
| MTBRekapKelas | Kelas untuk menyimpan data tabel rekap kelas |
| MTBRekapKontingen | Kelas untuk menyimpan data tabel rekap kontingen |
| MTBRekapSeni | Kelas untuk menyimpan data tabel rekap seni |
| MTBRekapUndianSeni | Kelas untuk menyimpan data tabel rekap undian seni |
| MTBRekapUndianTanding | Kelas untuk menyimpan data tabel rekap undian tanding |
| MTBJadwalTanding | Kelas untuk menyimpan data tabel jadwal tanding |
| MTBJadwalSeni | Kelas untuk menyimpan data tabel jadwal seni |
| MTBUndianTanding | Kelas untuk menyimpan data tabel undian tanding |
| MTBUndianSeni | Kelas untuk menyimpan data tabel undian seni |
| DbCodeConstan | Kelas untuk pendeklarasi code peserta seni |
| GeneratingJadwalTandingFacade | Kelas untuk menghitung jumlah partai tanding dan mengatur jadwal pada kategori tanding  |
| GeneratingJadwalSeniFacade | Kelas untuk menghitung jumlah partai tanding dan mengatur jadwal pada kategori seni |
| GeneralLogic | Kelas untuk mengatur jumlah partai dalam satu shift pertandingan  |
| DatabaseLogic | Kelas untuk memanggil database kelas dan peserta tanding |
| GenerateSeniInputDto | Kelas untuk memanggil dan memberikan waktu dimulainya pertandingan, jumlah hari pertandingan ,dan banyaknya pertandingan pada kategori seni  |
| GenerateTandingInputDto | Kelas untuk memanggil dan memberikan waktu dimulainya pertandingan, jumlah hari pertandingan ,dan banyaknya pertandingan pada kategori tanding |
| HasilEstimasiTandingDto | Kelas untuk memanggil dan memberikan jumlah hari pertandingan ,jumlah kelas yang ditandingkan, waktu berakhirnya pertandingan, dan banyaknya pertandingan pada kategori tanding |
| HasilEstimasiSeniDto | Kelas untuk memanggil dan memberikan jumlah hari pertandingan, waktu berakhirnya pertandingan, dan banyaknya pertandingan pada kategori seni |
| JadwalSeniBean | Kelas untuk memanggil dan memberikan tanggal, waktu, seni, jenis kelamin, id kontingen, dan nama kontingen untuk kelas generate jadwal seni |
| MatchInfoBean | Kelas untuk memanggil dan memberikan id kelas, keseluruhan no pertandingan, no partai, id player 1, id player 2, tanggal pertandingan, shift pertandingan, waktu pertandingan, no undian 1, dan no undian 2 untuk kelas generate jadwal tanding |
| MatchTimeBean | Kelas untuk memanggil dan memberikan hari pertandingan, shift pertandingan, dan waktu pertandingan |
| DataBarisJadwalTanding | Kelas untuk memanggil dan memberikan tanggal, waktu, id kelas dan no partai |
| DatabaseUtil | Kelas untuk memanggil database pada jadwal tanding dan jadwal seni |
| DateUtil | Kelas untuk memberikan nama hari |
| DbConstantUtil | Kelas untuk memberikan code pada peserta seni |
| KelasPartai | Kelas untuk memanggil dan memberikan kelas dan no partai |
| KnockoutNode | Kelas untuk memanggil dan memberikan no undian, induk, anak kanan, dan anak kiri |
| KnockoutTree | Kelas untuk menampilkan alur algoritma pohon dan menentukan berapa jumlah level maksimum pada generate jadwal tanding |
| Shift  | Kelas untuk menampilkan jumlah shift dalam sehari |

**3.3. Diagram Activity**

Diagram activity input data memiliki dua aksi yaitu admin dan aplikasi. Dalam sistem ini *admin* mempunyai hak akses untuk melakukan proses input yang akan diproses oleh aplikasi dan akan disimpan pada database, jika batal maka akan kembali ke form awal. Diagram activity dalam rancangan ini yaitu :

- Diagram *Activity* Input Data

- Diagram *Activity* Edit Data

- Diagram *Activity* Hapus Data

- Diagram *Activity* Lihar Rekap

- Diagram *Activity* Melakukan Undian

- Diagram *Activity* Melihat Jadwal

- Diagram *Activity* Generate Jadwal Tanding

*-* Diagram *Activity* Generate Jadwal Seni

**3.3. Diagram Sequence**

Diagram *sequence* adalah teknik untuk menggambarkan interaksi antar objek di dalam dan di sekitar sistem. Diagram *sequence* dapat dihasilkan dari setiap *use case* yang dianalisis. Diagram sequence yang digunakan dalam perancangan ini yaitu :

- Diagram *Sequence* Proses Pendaftaran Peserta Tanding dan Seni

- Diagram *Sequence* Proses Pendaftaran Manajer dan *Official*

- Diagram *Sequence* Proses Rekap Peserta Tanding, Peserta Seni, Peserta Per Kontingen, Undian Tanding, Undian Seni, Jadwal Tanding, dan Jadwal Seni

- Diagram *Sequence* Proses Undian Tanding dan Undian Seni

- Diagram *Sequence* Proses Generate Jadwal Tanding

- Diagram *Sequence* Proses Generate Jadwal Seni

**4. Implementasi**

**4.1 Perangkat lunak yang digunakan**

Perangkat lunak yang digunakan untuk membuat sistem pada penelitian ini dapat dilihat pada tabel 4.1.

**Tabel 4.1** Perangkat Lunak yang digunakan

|  |  |  |
| --- | --- | --- |
| **No** | **Perangkat Lunak** | **Keterangan** |
| 1 | Windows 7  | Sistem Operasi |
| 2 | J2SE JDK versi 6 update 20 | Perangkat lunak lingkungan pengembangan bahasa pemrograman Java untuk membat aplikasi |
| 3 | Netbeans IDE 6.7.1 | Perangkat lunak untuk *code* *editor* dan pemodelan UML |
| 4 | MySQL versi 5.1.36 | Perangkat lunak manajemen *database* untuk penyimpanan dan pengolahan data |
| 5 | CorelDRAW X4 | Perangkat lunak untuk mendesain *user interface* aplikasi |

**4.2 Perangkat keras yang digunakan**

Perangkat keras yang digunakan untuk membuat aplikasi pada penelitian ini dapat dilihat pada tabel 4.2 di halaman berikutnya.

**Tabel 4.2** Perangkat keras yang digunakan

|  |  |  |
| --- | --- | --- |
| **No** | **Perangkat Keras** | **Keterangan** |
| 1 | Processor | Intel Pentium Dual-Core T4200 @2.00 GHz |
| 2 | Memori | DDR2 1024 MB |
| 3 | VGA | Mobile Intel(R) 45 Express Chipset Family |
| 4 | Hard disk | 160 GB |

**4.3 Implementasi Program**

**4.3.1 Implementasi Form Utama Aplikasi**

Frame utama aplikasi adalah window utama/induk dari aplikasi pada saat pertama dijalankan.

****

**Gambar 4.1** Form Utama Aplikasi

**4.3.2 Implementasi Form Pendaftaran Peserta Tanding**

Form pendaftaran peserta tanding digunakan untuk menginput, menyimpan, mengedit, dan menghapus data peserta tanding.



**Gambar 4.2** Form Pendaftaran Peserta Tanding

**4.3.3 Implementasi Form Pendaftaran Peserta Seni**

Form pendaftaran peserta seni digunakan untuk menginput, menyimpan, mengedit, dan menghapus data peserta seni.



**Gambar 4.3** Form Pendaftaran Peserta Seni

**4.3.4 Implementasi Form Pendaftaran Manajer**

Form pendaftaran manajer digunakan untuk menginput, menyimpan, mengedit, dan menghapus data manajer.



**Gambar 4.4** Form Pendaftaran Manajer

**4.3.5 Implementasi Form Pendaftaran *Official***

Form pendaftaran *official* digunakan untuk menginput, menyimpan, mengedit, dan menghapus data *official*.



**Gambar 4.5** Form Pendaftaran *Official*

**4.3.6 Implementasi Form Rekap Peserta Per Kontingen**

Form rekap peserta per kontingen digunakan untuk melihat rekap data kontingen yang berpartisipasi dalam kejurnas.



**Gambar 4.6** Form Rekap Peserta Per Kontingen

**4.3.7 Implementasi Form Rekap Peserta Tanding**

Form rekap peserta tanding digunakan untuk melihat data peserta yang bertanding disetiap kelasnya.



**Gambar 4.7** Form Rekap Peserta Tanding

**4.3.8 Implementasi Form Rekap Peserta Seni**

Form rekap peserta senidigunakan untuk melihat data peserta seni yang bertanding sesuai dengan seni yang diikuti.



**Gambar 4.8** Form Rekap Peserta Seni

**4.3.9 Implementasi Form Undian Tanding**

Form undian tanding digunakan untuk melakukan proses pengundian pada kategori tanding.



**Gambar 4.9** Form Undian Tanding

**4.3.10 Implementasi Form Rekap Undian Tanding**

Form rekap undian tanding digunakan untuk rekap data pengundian pada kategori tanding.



**Gambar 4.10** Form Rekap Undian Tanding

**4.3.11 Implementasi Form Undian Seni**

Form undian seni digunakan untuk memproses data pengundian pada kategori seni.



**Gambar 4.11** Form Undian Seni

**4.3.12 Implementasi Form Rekap Undian Seni**

Form rekap undian seni digunakan untuk melihat rekap data pengundian pada kategori seni.



**Gambar 4.12** Form Rekap Undian Seni

**4.3.13 Implementasi Form Generate Jadwal tanding**

Form rekap undian seni digunakan untuk melihat rekap data pengundian pada kategori seni.



**Gambar 4.13** Form Generate Jadwal Tanding

**4.3.14 Implementasi Form Jadwal tanding**

Form rekap undian seni digunakan untuk melihat jadwal pada katergori tanding.



**Gambar 4.14** Form Jadwal Tanding

**4.3.15 Implementasi Form Generate Jadwal Seni**

Form rekap undian seni digunakan untuk melihat membuat jadwal pada kategori seni..



**Gambar 4.15** Form Generate Jadwal Seni

**4.3.16 Implementasi Form Jadwal Seni**

Form rekap undian seni digunakan untuk melihat membuat jadwal pada kategori seni..



**Gambar 4.16** Form Jadwal Seni

**5. Penutup**

**5.1 Kesimpulan**

Berdasarkan penelitianyang telah dilakukan, maka dapat disimpulkan :

1. Telah berhasil dibangun sebuah aplikasi penjadwalan kejurnas pencak silat yang bermanfaat untuk membantu kepanitiaan kejurnas pencak silat dalam olah data dan pembuatan jadwal.
2. Sistem aplikasi ini digunakan oleh satu tipe *user*, yaitu Admin.

**5.2 Saran**

Pada penelitian ini masih banyak terdapat kekurangan yang perlu dikembangkan lebih lanjut. Beberapa saran yang diajukan berkaitan dengan pengembangan aplikasi selanjutnya yaitu pengolahan data dibuat dengan sistem web, sehingga peserta dapat langsung input data dari jauh hari sebelum kejurnas dimulai. Aplikasi untuk pembuatan jadwal dibuat dinamis sehingga jadwal dapat berganti secara otomatis pabila terdapat perubahan pada bagan pertandingan.

**6. Daftar Pustaka**

Arbie, 2004, *Manajemen Database dengan MySQL*, Penerbit ANDI, Yogyakarta

Arianty, Ria, 2007, *Implementasi metode OLAP dalam Aplikasi Bisnis Perusahaan untuk Analisa Data Proyek Pengiriman Barang,* Skripsi Jurusan Teknik Informatika UPN “Veteran” Yogyakarta

Febrian, Jack, Andayani, Farida, 2002, *Kamus Komputer dan Istilah Teknologi Informasi*, CV INFORMATIKA, Bandung

Fowler, Martin, 2004, *A brief Duide to the Standard Object Modeling Language*, Penerbit ANDI, Yogyakarta.

Fowler, Martin, 2005, *UML Distilled Edisi 3 Panduan Singkat Bahasa Pemodelan Objek Standar*, Penerbit ANDI, Yogyakarta.

Haqqi, M. Fauzil, 2010, *Pengenalan Pemrograman Game Java Bagian 1,* PC Media Edisi 04/2010, Jakarta

Huda, Miftakhul, 2010, *Trik Rahasia Pemrograman Database Dengan Java*, PT Elex Media Komputindo, Jakarta

Jogiyanto, H. M., 1997, *Pengenalan Komputer*, Andi Offset, Yogyakarta

Leonardo, Ian, 2003, *Belajar Sendiri Pemrograman Java 2D*, PT Elex Media Komputindo, Jakarta

Munas IPSI, 2007, *Peraturan Pertandingan Pencak Silat*, PB IPSI, Jakarta

Nugroho, Bunafit, 2005, *Database Relational Dengan MySQL*, Andi Offset, Yogyakarta

Rumbaught J, Jacobson I, Booch G, 1999, *The Unified Modeling Language Reference Manual,* Addison Wesley, New Jersey.

Suhendar, A., Gunadi, Hariman, 2002, *Visual Modeling Menggunakan UML dan Rational Rose*, Informatika, Bandung

\_\_\_\_\_\_, 2010, *Database Website*, http://www.tips-techno.co.cc/2010/04/database-website.html (diakses 30 September 2010)

\_\_\_\_\_\_, 2012, *Pengertian Database*, <http://rudyharun.blog.binusian.org/2010/09/13/tugas-minggu-5-database/> (diakses 30 September 2010)

\_\_\_\_\_\_, 2012, *Teori Dasar Basis Data << Toromorphosis Mind’s*, <http://toromorphosist.wordpress.com/2010/02/06/teori-dasar-basis-data/> (diakses 30 September 2012)

\_\_\_\_\_\_, 2011, *RAD (Rapid Application Development*, http://odl-skopje.etf.ukim.edu.mk/UML-Help/html/01day3.html#ana (diakses 13 Januari 2013)

\_\_\_\_\_\_, 2011, *RAD (Rapid Application Development*, http://odl-skopje.etf.ukim.edu.mk/UML-Help/html/01day3.html#ana (diakses 30 april 2013)

Wikipedia, Pengertian Pohon, <http://en.wikipedia.org/wiki/Tree_graph_theory> (diakses 30 april 2013)