TESLA Vol. 8 No. 2, 75 – 95 (Oktpber 2006)
Jurnal Teknik Elektro

PERANCANGAN DAN IMPLEMENTASI PAPAN JADWAL PERKULIAHAN BERDASARKAN SISTEM PENJADWALAN OTOMATIS

Johan K. W.[1)], Adrianto H.[1)] dan Marsolim[2)]

Abstract
An automated lecture-schedule-board for students is provided. The board designed allows lecturers to enter changes on their teaching schedule and display the changes live automatically. The system is implemented using a microcontroller chip and an LED Matrix board on a Printed Circuit Board (PCB).

Keywords*: lecturer, schedule, display board.*

PENDAHULUAN

Latar Belakang

Penjadwalan (*Scheduling*) secara individual adalah salah satu cara yang dilakukan oleh manusia agar bisa memanfaatkan waktunya dengan seefisien mungkin. Pembuatan jadwal membutuhkan tingkat ketelitian yang tinggi dari orang yang akan membuat jadwal tersebut karena banyaknya batasan (*Constraint*) yang perlu diperhatikan. Masalah pembuatan jadwal ini dikategorikan sebagai *Constraint Satisfaction Problem* (CSP) dalam bidang ilmu Kecerdasan Buatan (*Artificial Intelligence* atau *AI*).

Masalah-masalah yang sering terjadi dalam pembuatan jadwal secara manual adalah kesalahan jadwal karena kurang telitinya pembuat jadwal tersebut. Terutama jika jumlah *constraint* yang ada sangat banyak, sehingga tidak memungkinkan bagi pembuat jadwal untuk memperhatikan semuanya. Tujuan dari pembuatan sistem penjadwalan otomatis adalah untuk menggantikan pembuatan jadwal secara manual, sehingga faktor kesalahan karena ketelitian dapat ditekan.

Dalam lingkungan akademis seperti di kampus, *Resource Scheduling* adalah sistem yang cocok untuk diterapkan karena keterbatasan sumberdaya seperti ruangan, dosen, dan peralatan ajar. Alat berupa *display* dapat ditambahkan untuk menampilkan hasil berupa jadwal yang dibuat oleh sistem penjadwalan otomatis. Alat *display* ini dapat disebut sebagai papan jadwal elektronis. Dalam penelitian ini, dilakukan survei di beberapa tempat yaitu:

1. Internet, untuk mencari informasi tentang sistem penjadwalan otomatis yang telah dipasarkan secara komersial yaitu sistem yang disebut sebagai *ResSched* yang dikembangkan oleh *Madrigal Soft Tools Incorporated*.
2. Bandara Soekarno Hatta, untuk mendapatkan informasi tentang bermacam-macam *display* yang digunakan untuk menampilkan jadwal penerbangan di Bandara Soekarno Hatta seperti papan *Light Emitting Diode –matrix* (*LED-matrix*), televisi dan *Liquid Crystal Display* (LCD).

Lingkungan kampus Univeritas Tarumanagara (UNTAR) menerapkan pergerakan peralatan ajar (fasilitas) untuk

[1)] Staf Pengajar Jurusan Teknik Elektro Fakultas Teknik Universitas Tarumanagara.
[2)] Alumni Teknik Elektro Fakultas Teknik Universitas Tarumanagara

memenuhi kebutuhan kelas sehingga fasilitas berpindah ruang terus menerus. Sistem yang lebih efisien daripada pergerakan fasilitas adalah sistem pergerakan pemakai fasilitas. Dengan meletakkan fasilitas pada ruangan-ruangan tertentu yang tetap, maka hanya diperlukan menggerakkan pemakai (dosen dan mahasiswa) ke ruangan yang menyediakan peralatan yang dibutuhkan. Namun sistem ini tidak bisa diterapkan pada sistem penjadwalan manual karena akan sering terjadi perubahan jadwal sebagai akibat dari perubahan ruangan kuliah yang terjadi secara dinamis sesuai kebutuhan pemakai. Oleh karena itu, dalam penelitian ini akan dirancang sebuah sistem penjadwalan otomatis yang diintegrasikan dengan papan jadwal elektronis (menggunakan papan *LED-matrix*) dan menerapkan sistem pergerakan fasilitas.

Tujuan Rancangan

Perancangan sistem penjadwalan otomatis ini bertujuan antara lain :

- Agar proses penjadwalan dapat dilakukan secara otomatis
- Untuk menjamin ketelitian dalam pembuatan jadwal
- Untuk meningkatkan efisiensi penggunaan peralatan ajar dan ruangan yang tersedia

Batasan Rancangan

Batasan rancangan pada perancangan sistem penjadwalan otomatis ini adalah:

- Sistem Penjadwalan Otomatis dibatasi hanya melakukan otomatisasi pembuatan jadwal dalam lingkungan Jurusan Elektro, Fakultas Teknik di UNTAR.
- Perangkat keras *display* yang akan dibuat dibatasi hanya terdiri dari dua baris.
- Baris pertama menampilkan tanggal dan waktu dan hanya menggunakan *alphanumeric display*.
- Baris kedua akan menampilkan jadwal yang dibuat secara bergantian dan menggunakan tampilan *LED-matrix*.
- Penampilan *display* secara lengkap akan digambarkan melalui simulasi perangkat lunak.
- Rancangan tidak termasuk unit *Personal Computer* (PC) yang digunakan.

Bagian dari Subblok-subblok yang dirancang, terdiri dari modul sistem, penjadwalan otomatis, modul *display*, modul *software interface*, dan modul *database*, Sedangkan bagian dari subblok-subblok yang tidak dirancang adalah modul komputer PC dan modul RS-232.

Spesifikasi Rancangan

Sistem penjadwalan otomatis yang dirancang menggunakan pendekatan analisis CSP dengan fungsi utamanya adalah membuat jadwal perkuliahan. Jadwal tersebut kemudian ditampilkan pada *display* berupa papan jadwal yang terhubung dengan komputer.

Sistem penjadwalan otomatis memiliki spesifikasi antara lain :

- Menggunakan *display* untuk menampilkan jadwal.
- Menggunakan komunikasi serial RS-232.
- Menggunakan sebelas (11) buah mikrokontroler.
- Menggunakan satu buah catu daya dengan keluaran +5 *Volt* dengan arus keluaran 3 Ampere.

LANDASAN TEORITIK

Deskripsi Konsep

Perancangan papan jadwal perkuliahan berdasarkan sistem penjadwalan otomatis berguna untuk menggantikan sistem penjadwalan manual. Dengan demikian, sistem pengaturan sumber daya yang lebih baik dapat diterapkan sehingga pemanfaatan sumber daya yang lebih efektif dan efisien dapat tercapai.

Perancangan papan jadwal perkuliahan berdasarkan sistem penjadwalan otomatis ini meliputi beberapa bagian yaitu bagian

perancangan sistem penjadwalan otomatis, bagian perancangan *software interface*, bagian perancangan *display* dan bagian perancangan *database*.

Bagian perancangan sistem penjadwalan otomatis merupakan perancangan sistem yang akan melakukan pembuatan jadwal perkuliahan secara otomatis. Sistem ini dibuat dengan menggunakan penerapan CSP yang diintegrasi dengan tehnik searching dalam AI. *Constraint* (batasan) yang diterapkan dalam pembuatan jadwal sesuai data sumberdaya yang tersedia.

Bagian perancangan *software interface* merupakan perancangan antarmuka bagi pengguna sistem ini. *Software interface* ini berfungsi sebagai perantara untuk melakukan perubahan terhadap jadwal kuliah yang sudah dibuat dan memasukkan data ketersediaan sumberdaya kedalam sistem *database*. Selain itu juga untuk mengirim hasil jadwal yang dibuat ke unit *display* yang terhubung dengan komputer yang menjalankan *software* ini. *Software interface* inilah yang akan berinteraksi dengan pengguna sistem ini.

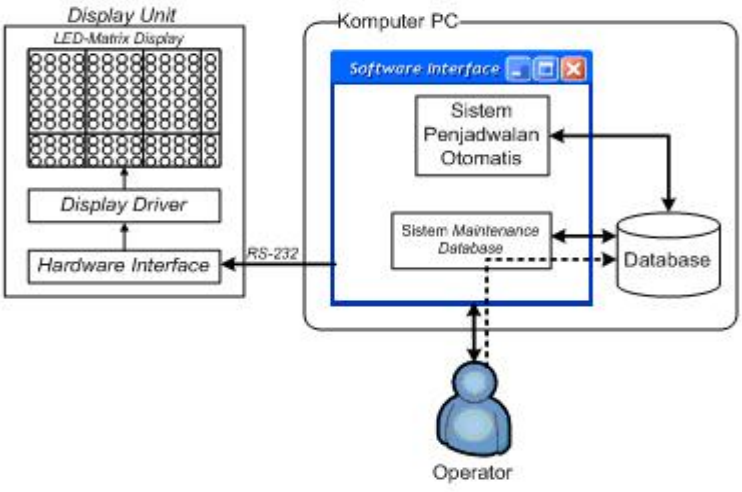
Melalui *software interface* ini, pengguna (dalam konteks ini mengacu pada operator) dapat melakukan berbagai aktivitas perawatan (*maintenance*) data. Aktivitas seperti melakukan perubahan data dan pengecekan terhadap validitas data tidak dapat langsung dilakukan pada *database*, tetapi melalui *software interface*. Hal ini dikarenakan *database* hanya bisa diubah melalui *query* yang dimasukkan ke server database. *Software interface* menyediakan tampilan yang lebih mudah dimengerti untuk melakukan aktivitas perawatan. Dengan demikian, seorang operator tidak perlu memiliki pengetahuan secara khusus tentang sistem *database* yang digunakan.

Bagian *display* dirancang untuk menampilkan jadwal yang dibuat oleh sistem penjadwalan otomatis. *Display* akan menampilkan jadwal yang dikirim oleh *software interface*. Setiap kali terjadi perubahan jadwal, jadwal akan disimpan dalam *database* dan *software interface* hanya akan mengirimkan jadwal dalam database kepada *display*. Jadwal yang ditampilkan *display* selalu merupakan jadwal yang terbaru.

Bagian perancangan *database* merupakan perancangan untuk menyimpan data sumberdaya dan juga data jadwal yang telah dibuat. Bagian ini merupakan bagian yang kritis dan perlu perhatian khusus karena data yang dipakai untuk membuat jadwal tersimpan disini. Untuk mengetahui diagram blok dari perancangan alat ini, dapat dilihat pada Gambar 1.

Diagram Blok

Diagram blok keseluruhan rancangan alat dapat dilihat pada Gambar 1. Secara garis besar sistem terbagi atas dua blok utama yaitu blok komputer PC dan blok *Display Unit*. Komputer PC menjalankan sistem penjadwalan otomatis dan *database maintenance* yang dioperasikan oleh operator. *Display* unit hanyalah merupakan perpanjangan fungsi dari sistem ini yang berfungsi sebagai papan jadwal elektronik.



Gambar 1. Diagram blok perancangan papan jadwal perkuliahan berdasarkan sistem penjadwalan otomatis.

Pada sistem ini dapat dibedakan dua *user level* yang berbeda. Pertama adalah *high level user* atau disebut sebagai operator. Operator adalah user yang mengatur dan melakukan *database maintenance* dalam sistem. *User* yang lain adalah *low level user* yang hanya memiliki kebutuhan terhadap hasil berupa jadwal yang dibuat oleh sistem. *User* seperti ini antara lain adalah mahasiswa dan dosen. *Low level user* tidak bisa melakukan perubahan jadwal secara langsung tanpa melalui operator.

Komputer PC

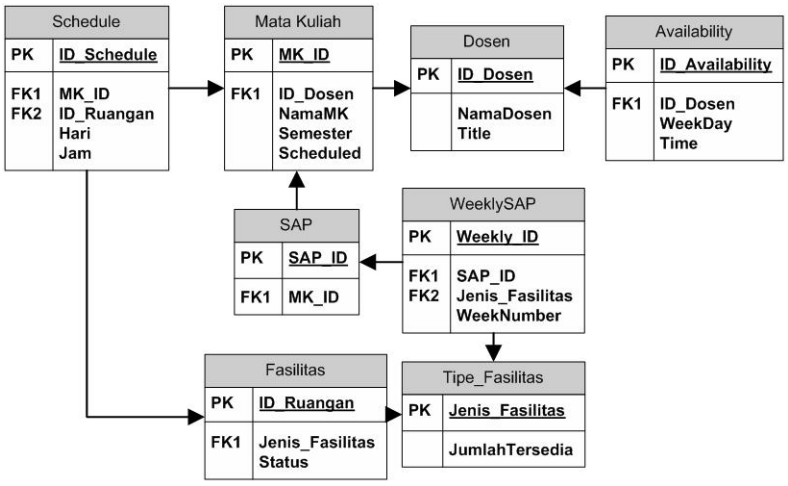
Komputer PC merupakan otak dari keseluruhan sistem. Pada komputer PC, interaksi antara operator dengan *software interface* terjadi. Pembuatan jadwal menggunakan sistem penjadwalan otomatis dan *database maintenance* juga dilakukan di komputer PC melalui *software interface*. Fungsi lain yang tidak kalah penting adalah sebagai tempat penyimpanan data dalam sebuah sistem *database*.

Dalam komputer PC terdapat dua modul yang sangat penting yaitu *software interface* dan *database*. Modul *software interface* berfungsi sebagai perantara yang akan menjembatani antara operator dengan modul lainnya. Modul *data base* ini memiliki fungsi utama sebagai tempat menampung informasi.

Fungsi utama modul *Database* adalah sebagai penyimpanan data berupa tabel. Semua data permanen yang digunakan oleh sistem ini disimpan dalam modul ini. Modul ini sangat penting karena sistem ini sangat tergantung pada *data* yang tersimpan dalam modul ini. Integritas *data* menjadi hal yang sangat penting karena tingkat ketergantungan sistem terhadap *data* sangat tinggi. Modul ini menjadi penentu tingkat *scalability* sistem penjadwalan otomatis. Penambahan *data* yang ada dalam *database* dapat menambah variabel dan *constraint* dalam sistem penjadwalan otomatis sesuai dengan *data* yang ditambahkan. Hal inilah yang menyebabkan ketergantungan sistem terhadap data sangat tinggi. Selain itu *relational model* dari *database* juga harus diperhatikan. Rancangan *database* pada sistem dapat dilihat pada Gambar 2.

Software interface memiliki tiga fungsi utama. Yang pertama adalah sebagai *user interface* bagi operator dalam menggunakan sistem ini. Fungsi kedua adalah sebagai pembuat jadwal otomatis dengan menggunakan sistem penjadwalan otomatis. Karena sistem ini menggunakan *database* sebagai penyimpanan *data*, maka diperlukan fungsi ke tiga yaitu sebagai *database maintenance system*. Sebagai *user interface*, modul ini menyediakan berbagai fungsi administrasi yang disederhanakan. Tujuan penyederhanaan ini adalah untuk memudahkan operator dalam mengoperasikan sistem ini. Operator tidak perlu memiliki pengetahuan khusus tentang *database* untuk menjalankan sistem ini.

Fungsi *database maintenance sistem* dalam modul ini hanya berupa *maintenance sistem* yang sederhana. *Database maintenance sistem* pada modul ini tidak seperti yang diimplementasikan pada sebuah *Data Base*



Gambar 2. Diagram desain *database*

Management Sistem (DBMS) seperti SQL (*Structured Query Language*). Modul ini hanya mengimplementasikan fungsi-fungsi *maintenance* yang sederhana.

Sistem Penjadwalan Otomatis

Penjadwalan otomatis dalam modul ini dilakukan dengan mengaktifkan sistem penjadwalan otomatis yang terintegrasi dalam modul. Sistem penjadwalan otomatis yang diimplementasikan pada modul ini tidak bisa menjamin untuk mendapatkan solusi. Karena itulah perlu adanya implementasi penjadwalan manual dan perubahan jadwal secara manual supaya dapat menggantikan sistem jika terjadi kegagalan dalam penjadwalan otomatis. Selain implementasi penjadwalan manual, modul ini juga melakukan fungsi pengiriman *data* ke modul *display*. *Data* yang dikirim ke modul *display* merupakan *data* hasil pengolahan yang berasal dari *database*. Modul ini mengirim *data* ke modul *display* melalui jalur komunikasi *serial port* yang tersedia pada komputer PC.

Software interface merupakan *Application Core* dari sistem. Modul ini melakukan penjadwalan otomatis, berinteraksi dengan operator, dan menampilkan jadwal pada *unit display*. Semua aktivitas sistem terpusat pada modul ini. Karena itulah modul ini merupakan bagian paling vital dalam sistem ini.

Sistem penjadwalan secara otomatis diimplementasikan dalam bentuk *class library*. Semua *library* disusun dalam satu *framework* yang diberi nama *JSched*. *Framework* JSched ini disusun khusus untuk penjadwalan mata kuliah, bukan merupakan sebuah *framework* yang bersifat *general purpose*. Dengan implementasi seperti ini maka pengembangan modul ini menjadi lebih sederhana dan mudah untuk di adaptasikan dengan sistem lain.

Teknik pendekatan yang digunakan untuk menyusun *framework* JSched adalah pendekatan CSP. CSP didefinisikan sebagai sebuah masalah untuk mengasosiasikan sekumpulan variabel (V) dengan sekumpulan *value* dalam *domain* (D) dengan syarat harus memenuhi batasan-batasan tertentu (C). Solusi dari CSP dicapai dengan setiap variabel (V) berhasil diasosiasikan dengan *value* dari *domain* (D) dimana setiap asosiasi tidak melanggar *constraint* (C). Jadi constraint menentukan *value* dari domain yang boleh diasosiasikan dengan variabel. Notasi umum untuk CSP adalah P = {V, D, C} dimana P adalah *problem* CSP, V adalah himpunan variabel V = $\{X_1, X_2 \ldots X_n\}$, D adalah *domain* yang merupakan himpunan *value* D_i = {A, B, C, dll}, dan C adalah *constraint* yang merupakan himpunan pasangan variable dan himpunan value yang diperkenankan C = $\{(V_1, D_1) \ldots (V_i, D_i)\}$. C biasanya direpresentasikan dalam bentuk fungsi matematis tergantung pada struktur dari *problem* yang ingin diselesaikan.

Dalam penjadwalan kuliah, yang menjadi variabel adalah matakuliah yang harus disusun dalam sebuah tabel waktu (*domain*) dimana yang menjadi *value* adalah waktu. Sedangkan yang menjadi *constraint* adalah *resource* berupa fasilitas, ruangan dan ketersediaan staf pengajar. Aturan-aturan dasar dalam sistem penjadwalan seperti bentrokan waktu (*Time Collision*) dan juga bentrokan tempat (*Space Collision*) termasuk dalam *constraint*.

Time collision terjadi jika seorang dosen mengajar dua matakuliah yang berbeda pada ruangan yang berbeda tapi dijadwalkan pada waktu yang sama. *Constraint* ini dapat diatasi dengan menggunakan struktur *time slot*, dimana waktu kuliah dalam satu hari dibagi-bagi menjadi bagian-bagian kecil yang memiliki durasi tertentu (misalnya 1 jam). *Time slot* inilah yang akan dialokasikan untuk jadwal matakuliah. Dengan demikian, tidak ada matakuliah yang bisa berbagi *time slot* yang artinya tidak akan terjadi *time collision*.

Space collision, berbeda dengan *time collision*, terjadi ketika dua orang dosen yang mengajar dua matakuliah yang berbeda dijadwalkan pada ruangan yang sama. Untuk mengatasinya, maka tiap ruangan akan

mengimplementasikan *time slot*. Dengan demikian, setiap ruangan akan memiliki alokasi waktu tersendiri. Namun implementasi ini memicu permasalahan lain. Karena alokasi waktu setiap ruangan bersifat independen, maka dapat menyebabkan terjadinya *time collision* yang lebih terikat pada ketersediaan staf pengajar. Masalah *time collision* dapat diatasi dengan mengimplementasikan *time slot* pada data ketersediaan staf pengajar. Jadi sistem dapat melacak terjadinya *time collision*.

Perubahan struktur data yang digunakan menimbukan perubahan pada definisi masalah. Penyatuan waktu dan ruangan (*time and space*) dalam satu struktur *data* menyebabkan perubahan *constraint* yang ada. *Constraint* yang menentukan ruangan yang dipakai kini hanya bergantung pada fasilitas. Dengan demikian *problem* ini menjadi lebih sederhana namun bersifat *domain specific*.

Solusi dari *problem* penjadwalan dapat dicapai dengan menggunakan algoritma *searching* dalam AI. Semua jenis algoritma searching dapat digunakan untuk mencari solusi *problem* CSP. Mulai dari *stochastic search* (menggunakan algoritma *generate and test*) sampai implementasi *heuristic*. Dari semua algoritma ini tidak ada satupun algoritma yang bisa menjamin kesuksesan menemukan solusi. Yang menjadi pertimbangan pemilihan algoritma-algoritma di atas adalah pertimbangan *time and space* (waktu proses dan *space* memori yang dibutuhkan) *tradeoff*.

Sistem *Maintenance Database*

Sistem *maintenance database* merupakan salah satu bagian utama aplikasi ini. Keberadaan modul sistem memungkinkan operator melakukan berbagai perubahan terhadap *database* dan kegiatan lain yang bersangkutan dengan *database*. Modul ini menjadikan sistem penjadwalan otomatis dalam rancangan ini bersifat *scalable* terhadap *problem size*. Artinya dapat dilakukan penambahan variabel dan *constraint* sesuai dengan *data* yang ada dalam *database*. Modul ini diimplementasikan dalam bentuk kumpulan blok *user interface* yang khusus mengimplementasikan akses *administrator* ke *database*. Perubahan *database* ruangan, matakuliah, dosen, dan fasilitas harus dilakukan dalam ruang lingkup modul ini. Sedangkan perubahan pada *database* jadwal dapat dilakukan di luar ruang lingkup modul ini karena adanya implementasi penjadwalan dan perubahan jadwal manual. Fungsi-fungsi perubahan *database* yang diimplementasikan dalam modul antara lain :

1. Fungsi untuk mengubah *data* yang tersimpan dalam *database* yang menggunakan *query* update.
2. Fungsi untuk menambah *data* kedalam *database* menggunakan *query* insert.
3. Fungsi untuk menghapus *data* yang tersimpan dalam *database* yang menggunakan *query* delete.

Ketiga fungsi memberikan kemampuan *maintenance* terhadap *data* yang tersimpan dalam *database*.

Display Unit

Penambahan *unit display* ditujukan untuk meningkatkan efisiensi sistem. Kemampuan penjadwalan dinamis akan menyebabkan terjadinya perubahan jadwal secara dinamis. Perubahan jadwal dinamis membutuhkan alat untuk menampilkan jadwal yang memiliki kemampuan tampilan dinamis. Keunggulan *display* elektronik ada pada kemampuan tampilan dinamis.

Informasi yang ditampilkan pada *unit display* cukup sederhana mengingat keterbatasan *display* yang diimplementasikan. Informasi berupa waktu dan tanggal ditampilkan dalam format dd mm yy (untuk tanggal) dan hh:mm:ss (untuk waktu). Informasi jadwal ditampilkan dalam format tabular dengan susunan seperti R-MK-W (R = Ruangan, MK = Mata Kuliah, W = Waktu). Untuk mata kuliah hanya ditampilkan dalam bentuk singkatan. Jadwal yang ditampilkan adalah jadwal harian, bukan jadwal keseluruhan semester.

Modul *display unit* ini terbagi atas beberapa submodul berdasarkan fungsinya masing-masing. *Unit interface* yang digunakan untuk modul ini disebut *Hardware Interface*. Fungsinya adalah sebagai jembatan komunikasi antara komputer PC dengan modul *display unit*. Untuk mengatur tampilan dari modul *display unit*, digunakan rangkaian-rangkaian mikrokontroler yang dikelompokkan sebagai *Display Driver Circuit*. Kemudian untuk menampilkan data digunakan penampil standar *LED-Matrix*, yang berfungsi menampilkan jadwal, dan penampil standar lainnya berupa penampil *Alphanumeric*, yang digunakan untuk menampilkan tanggal dan waktu.

Tiga kemampuan yang harus dimiliki submodul *Hardware Interface* adalah kemampuan menerima data, kemampuan mengolah data, dan kemampuan menyimpan data sementara. Ketiga kemampuan ini diimplementasikan pada komponen yang berbeda-beda namun merupakan satu kesatuan fungsional yaitu sebagai *interface*.

Kemampuan menerima data adalah kemampuan untuk menerima data yang dikirim oleh komputer PC. Hal ini menunjukkan bahwa komunikasi yang terjadi hanya bersifat satu arah saja. Fungsi dari modul *display unit* hanyalah untuk menampilkan *data* yang dikirim dari PC, jadi tidak diperlukan adanya *feedback* atas data yang telah diterima dari PC.

Menyebarkan data yang dikirim dari PC ke semua mikrokontroler pada *display driver circuit* dilakukan oleh sebuah mikrokontroler master (dalam modul ini). Pengiriman data dilakukan secara parallel dan dikirimkan per-huruf. Data dari PC disimpan sementara di memori mikrokontroler master.

Modul *Interface* RS-232

Komunikasi antara *hardware interface* dengan PC menggunakan komunikasi serial dengan implementasi standar RS-232. Standar RS-232 dikembangkan oleh *Electronic Industry Association and the Telecommunications Industry Association* (EIA/TIA) dengan ketentuan level tegangan antara lain :

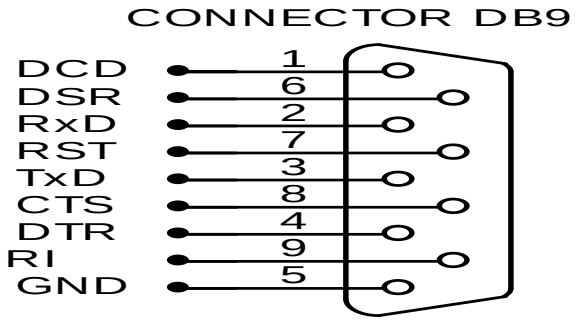
1. Untuk *logic* ‘1’ berada pada tegangan antara –3 sampai dengan –25 volt.
2. Daerah tegangan antara –3 Volt hingga +3 Volt adalah *invalid level*, yaitu daerah tegangan yang harus dihindari karena tidak memiliki level logika yang pasti. Sama halnya dengan daerah level tegangan yang lebih negatif dari –25 Volt atau lebih positif dari +25 Volt juga harus dihindari karena tegangan tersebut dapat merusak *line driver* pada saluran RS-232. Untuk *logic* ‘0’ berada pada tegangan antara +3 sampai +25 volt.
3. Arus pada kondisi *short circuit* tidak boleh lebih dari 500 mA.
4. Kemampuan untuk transmisi adalah sepanjang 15 meter (50 *feet*).

Komunikasi serial RS-232 bersifat asinkron, yang artinya sinyal *clock* tidak dikirimkan bersamaan dengan data. Masing-masing *word* disinkronkan dengan menggunakan *bit start*-nya dan *clock internal* pada masing-masing komputer. Transmisi data serial RS-232 disebut juga *single-ended* atau *unbalanced data transmission*. Dalam *unbalanced data transmission* ini, satu sinyal data dikirimkan melalui satu jalur kabel, kemudian informasi logika ditafsirkan dari beda tegangan kabel tersebut terhadap *ground*. Untuk pengiriman banyak sinyal, cukup dipakai kabel sebanyak sinyal yang dikirim dan kabel *ground* yang dipakai tetap sama. Transmisi data serial RS-232 biasanya digunakan untuk menghubungkan komputer dengan printer atau modem.

Kekurangan transmisi data serial RS-232 ini adalah sangat sensitif dan rentan terhadap gangguan. Hal ini disebabkan karena saluran *ground* merupakan bagian dari sistem transmisi, sehingga pergeseran beda tegangan terhadap *ground* sangat berpengaruh pada kualitas sinyal yang diterima. Transmisi data serial RS-232 ini hanya memiliki kemampuan komunikasi asinkron dua arah antara dua

peralatan (tidak *multipoint*). Keterbatasan jarak transmisi hanya dapat mencapai 50 *feet* atau kurang lebih 15 meter dengan kecepatan transfer 20 Kbps.

Port serial yang biasa digunakan untuk komunikasi serial memiliki 9 pin konektor yang berada di bagian belakang komputer, di mana masing-masing pin mempunyai fungsi yang berbeda-beda. Konektor ini adalah konektor yang paling umum dan disediakan pada setiap unit komputer PC sekarang ini. Konektor *port* serial DB-9 ditunjukkan pada Gambar 3.



Gambar 3. Konektor *port* Serial DB-9.

Display driver circuit disusun menggunakan mikrokontroler dimana setiap tiga huruf diatur oleh satu mikrokontroler. Setiap mikrokontroler akan menerima data yang berbeda berdasarkan urutan huruf yang diatur. Informasi *font* huruf disimpan dalam tiap-tiap mikrokontroler. Implementasi seperti ini lebih memudahkan proses *scanning* yang dilakukan per-3 huruf. Frekuensi *scanning* yang lebih tinggi juga dapat dicapai sehingga *display* bisa menampilkan tanpa terlihat *blinking* (kedap-kedip). Selain itu, ekspansi jumlah huruf dapat dilakukan dengan mudah tanpa harus mengubah keseluruhan sistem. Apabila hanya mengandalkan arus langsung dari keluaran mikrokontroler, nyala LED yang dihasilkan kurang terang. Akibatnya huruf yang ditampilkan *display* menjadi sulit terlihat. Transistor digunakan untuk mengatasi masalah ini.

Mikrokontroler merupakan komponen elektronika yang didalamnya terdapat rangkaian mikroprosesor, memori RAM/ ROM (*Random Access Memory*/*Read Only Memory*) dan I/O (*Input*/*Output*). Semua rangkaian tersebut terdapat dalam satu *chip* atau biasa disebut *single chip microcomputer*. Hal ini membedakan mikrokontroler dengan komponen elektronika yang lainnya. Keberadaan mikroprosesor memberikan kemampuan mengolah data pada komponen ini. Data yang dikirim dari PC melalui RS-232 akan disenarkan ke semua mikrokontroler pada *display driver circuit*. Program yang dimasukkan kedalam mikrokontroler akan mengambil 3 huruf dari data yang diterima sesuai dengan nomor identifikasi yang dikirim oleh PC.

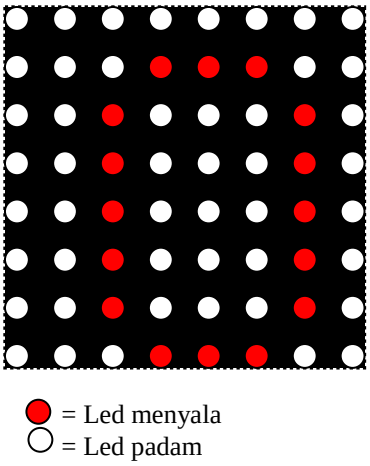
Salah satu produsen mikrokontroler terbesar di dunia adalah Intel Corp. Produk mikrokontroler Intel yang banyak digunakan secara umum adalah mikrokontroler dari keluarga MCS-52. Hal ini karena mikrokontroler tersebut telah dikenalkan sejak lama, selain itu juga harga jualnya yang murah dan telah tersedianya beberapa perangkat lunak pendukung mikrokontroler ini (*compiler, linker, simulator*).

Selain MCS-52, Intel juga memproduksi keluarga mikrokontroler MCS-51 yang relatif terkenal. Contohnya adalah mikrokontroler seri 8031. Hal ini dikarenakan mikrokontroler ini paling murah harganya sehingga relatif mudah mendapatkannya. Kelemahan dari mikrokontroler ini adalah tidak memiliki memori di dalamnya. Memori untuk program harus ditambahkan sendiri dalam bentuk *chip* EPROM dari luar. Akibat penggunaan memori eksternal, pada mikrokontroler 8031 ada 2 port yang tidak bisa digunakan untuk fungsi lain karena alokasi untuk akses memori eksternal.

Mikrokontroler seri 8751 dari keluarga MCS-51 memiliki memori program internal dalam bentuk EPROM. Program dapat disimpan dalam EPROM internal, dengan demikian ke empat port pada mikrokontroler tersebut dapat digunakan semua. Perubahan program dapat dilakukan dengan menghapus isi EPROM dengan sebuah EPROM *Eraser*.

Mikrokontroler seri 8051 juga memiliki memori internal yang bersifat *read only*. Walaupun menggunakan memori read only, tapi tetap dapat diisi program dengan daya tahan 1000 kali penulisan kembali.

LED-Matrix Display Unit, *Display* standar yang digunakan dalam modul ini adalah *display LED-Matrix*. Ukuran huruf menggunakan 5 × 7 led untuk satu karakter ASCII. Sehingga semakin banyak data yang harus ditampilkan, semakin banyak led yang dibutuhkan dan semakin besar ukuran fisik modul *display* ini. *LED-Matrix* hanya digunakan untuk menampilkan jadwal saja. Pada Gambar 4 merupakan gambaran unit *display* yang digunakan.



Gambar 4. *Unit display* untuk menampilkan satu karakter ASCII

Gambar 5. Unit *display alphanumeric*

Alphanumeric Display Unit, Untuk menampilkan elemen tanggal dan waktu, digunakan *display* standar *alphanumeric*. Display ini memiliki konsep kerja yang hampir sama dengan konsep kerja LED-Matrix. Perbedaannya adalah *alphanumeric display* tidak memerlukan *scanning* untuk menampilkan satu huruf. Pada Gambar 5 diperlihatkan *alphanumeric display* yang digunakan.

Display Simulator

Sebuah *software* simulasi digunakan untuk menunjukkan tampilan *display* secara lengkap (7 baris *LED-Matrix*). *Software* ini dibuat hanya untuk tujuan simulasi saja. Implementasi *software* ini terpisah dari *Software Interface* yang berfungsi untuk membuat jadwal.

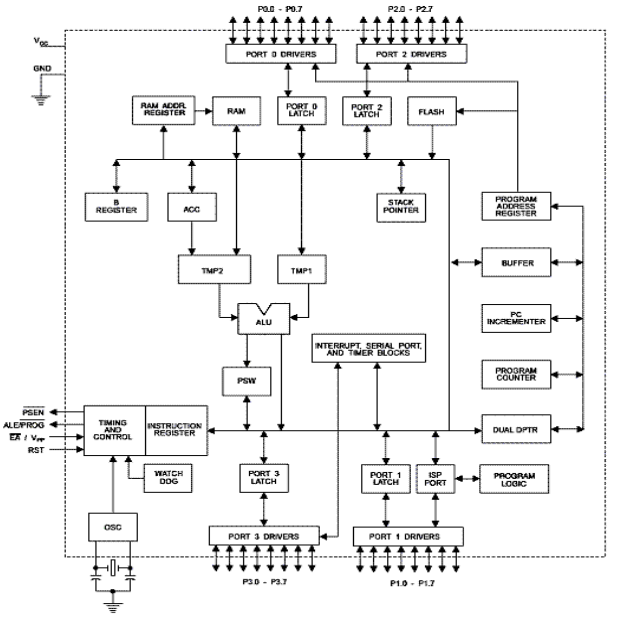
Tujuan dari *simulator* ini adalah untuk menunjukkan kemampuan *update* jadwal dinamis yang bisa ditampilkan oleh *display* yang dirancang. Modul *display* jika diimplementasikan secara lengkap akan tampak seperti yang ditunjukkan oleh *simulator*. Kemampuan simulator dibatasi hanya simulasi tampilan saja, tidak mensimulasikan *blinking* dan *scanning* yang terjadi pada modul *display* yang sebenarnya.

METODA DAN REALISASI RANCANGAN

Pemilihan Tipe Komponen

Perancangan sistem ini memerlukan komponen elektronika yang memiliki kemampuan proses data. Komponen elektronika dengan kemampuan seperti ini digolongkan sebagai mikrokontroler. Mikrokontroler AT89C51 merupakan salah satu jenis mikrokontroler CMOS 8 bit yang memiliki performa yang tinggi dengan disipasi daya yang rendah, cocok dengan produk MCS-51. Kemudian memiliki sistem pemograman kembali ISP Flash Memori 4 Kbyte dengan daya tahan 1000 kali *write/erase*. *Chip* ini diproduksi menggunakan *nonvolatile memory* yang sangat rapat dan setara dengan standar industri pin dan instruksi 80C51. *Flash* PEROM dari

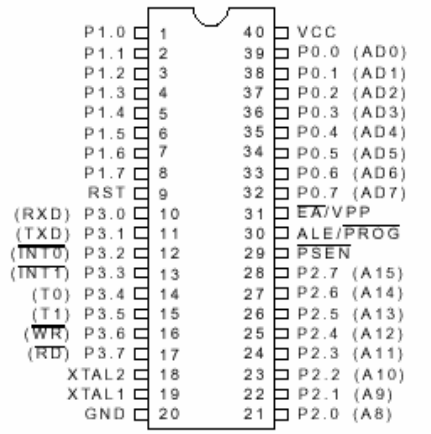
mikrokontroler ini dapat menyimpan data dan dapat diprogram ulang secara *in-sistem* melalui *serial peripheral interface* (SPI) atau bisa juga dengan menggunakan *programmer* pada umumnya. Gambar 6 menunjukkan blok diagram dari AT89S51.



Gambar 6. Blok diagram AT89S51

Di samping itu terdapat RAM Internal dengan kapasitas 128 x 8 bit. Dan frekuensi pengoperasian hingga 24 MHz. Frekuensi ini tergantung *clock external* yang biasanya dibangkitkan menggunakan *crystal oscillator*. Mikrokontroler ini juga memiliki 32 port I/O yang terbagi menjadi 4 buah port dengan 8 jalur I/O, kemudian terdapat pula Sebuah port serial dengan kontrol serial *full duplex*, dua *timer/counter* 16 bit dan sebuah osilator internal dan rangkaian pewaktu. Konfigurasi pin pada mikrokontroler dapat dilihat pada Gambar 7. Pin lain selain 32 I/O *port* memiliki fungsi sesuai dengan yang dipaparkan berikut ini :

A. Pin 29, *Program Store Enable* (PSEN) merupakan sinyal pengontrol pengakses program memori eksternal masuk ke dalam bus selama proses pemberian/ pengambilan instruksi (*fetching*). Digunakan jika program yang dibuat tidak muat dalam memori mikrokontroler ini sehingga sebagian program disimpan dalam memori eksternal.



Gambar 7. Konfigurasi *microcontroller* 89C51

B. Pin 30, *Address Latch Enable* (ALE)/ PROG merupakan penahan alamat memori eksternal (pada port 1) selama mengakses ke memori eksternal. Pin ini juga berfungsi sebagai pulsa/sinyal *input* pemograman (PROG) selama proses pemograman.
C. Pin 31, *External Access Enable* (EA) merupakan sinyal kontrol untuk pembacaan memori program. Apabila diset rendah (L) maka mikrokontroler akan melaksanakan seluruh instruksi dari memori program eksternal, sedangkan apabila diset tinggi (H) maka mikrokontroler akan melaksanakan instruksi dari memori program internal ketika isi program *counter* kurang dari 4096. ini juga berfungsi sebagai tegangan pemograman (V_{PP} = +12V) selama proses pemograman.
D. Pin 40, Merupakan positif sumber tegangan yang diberi simbol V_{CC}.

Adapun beberapa fasilitas yang terdapat pada mikrokontroler AT89S51 yang mendukung perancangan dan implementasi sistem yang dibuat adalah sebagai berikut:

- *In-Sistem Programmable*(ISP) *Flash Memory* sebesar 4 K*Byte*s yang dapat ditulis dan dihapus sebanyak 1000 kali

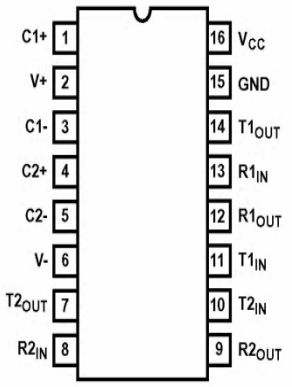
- Daerah operasi antara 4.0 volt sampai 5.0 volt
- Bekerja pada daerah frekuensi 0 Hz sampai 24 MHz
- Tiga tingkatan *program memory lock*
- 128 x 8-bit *internal* RAM
- 32 jalur *programmable* I/O
- Dua buah *timer/counter* 16 bit
- *LOW-power Idle and Power-down Modes*
- *Interrupt Recovery from Power-down Mode*
- *Dual Data Pointer*
- Waktu proses program yang cepat

Antarmuka serial RS-232 memiliki kelebihan dan kekurangan dibanding antarmuka paralel. Pada rancangan ini digunakan antarmuka serial karena beberapa kelebihannya, antara lain:

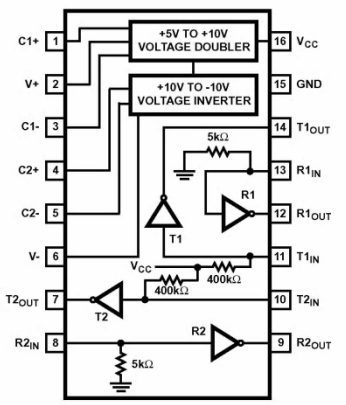
- Kabel untuk komunikasi serial bisa lebih panjang dibandingkan dengan paralel.
- Jumlah kabel serial lebih sedikit dibandingkan dengan kabel komunikasi paralel.
- Untuk teknologi *embedded sistem*, banyak mikrokontroler yang dilengkapi dengan komunikasi serial atau *Serial Communication Interface* (SCI).

Perancangan sistem menggunakan level sinyal/logika TTL. Sinyal port serial dari komputer menggunakan level sinyal/logika RS-232. Supaya sistem bisa berkomunikasi dengan komputer PC, maka sinyal port serial komputer harus dikonversi dulu ke pulsa TTL, dan sebaliknya sinyal dari peralatan harus dikonversi ke logika RS-232 sebelum dikirimkan ke serial port.

Konverter kedua sinyal tersebut yang paling mudah dan umum digunakan adalah IC (*Integrated Circuit*) MAX-232 buatan MAXIM Corp. Di dalam IC ini terdapat *Charge Pump* yang akan membangkitkan +10 Volt dan –10 Volt dari sumber +5 Volt tunggal. Dalam IC DIP (*Dual In-line Package*) 16 pin ini terdapat 2 buah *transmitter* dan 2 *receiver*, seperti terlihat pada Gambar 8 dan Gambar 9 di bawah.



Gambar 8. Konfigurasi Pin MAX-232

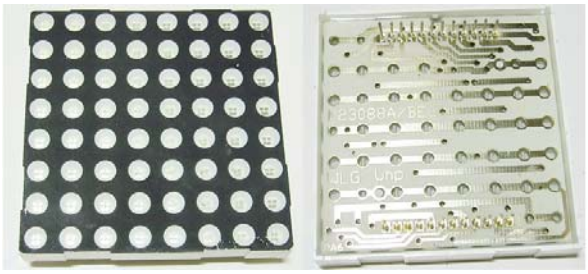


Gambar 9. Diagram Internal MAX-232

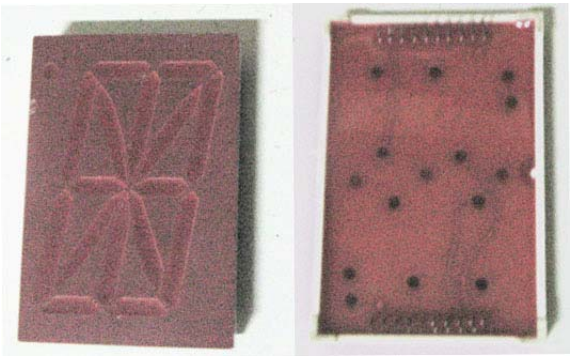
Paket LED-Matrix dipilih sebagai alat penampil utama dengan ukuran 8x8 LED. Keunggulan dari LED-Matrix adalah kemampuan untuk menampilkan lebih banyak variasi karakter. *Liquid Crystal Display* (LCD) juga menggunakan konsep tampilan LED-Matrix. Komponen LED-Matrix memiliki satu kelemahan yang dapat berakibat fatal jika tidak diperhatikan. Kelemahan terdapat pada tampilan karakter yang tidak ditampilkan secara keseluruhan tetapi ditampilkan perbaris.

Proses penampilan seperti ini disebut sebagai metoda *scanning*. Jika dilakukan pada frekuensi tinggi dapat terlihat seperti tampilan statis. Namun pada frekuensi rendah akan terlihat kedap-kedip LED (*blinking*). Karena itu penggunaan LED-Matrix harus memperhitungkan kecepatan *scanning* (frekuensi) yang mampu didukung oleh unit pengontrol atau disebut *driver*. Komponen LED-Matrix yang dipilih untuk display unit

dalam sistem ini adalah LED-Matrix Pack M23088A/BEG yang memiliki 12 x 2 pin. Gambar 10 memperlihatkan tampak depan dan tampak belakang LED-Matrix Pack M23088A/BEG. Konfigurasi pin LED-Matrix Pack M23088A dapat dilihat pada Gambar 11.



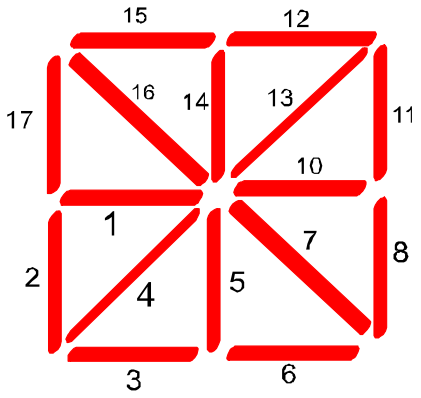
Gambar 10. Gambaran Fisik LED-Matrix Pack M23088A



Gambar 11 Gambaran Fisik *Alphanumeric Display*

Alpha numeric display merupakan alat penampil basis LED yang disusun sedemikian rupa sehingga dapat menampilkan huruf dan angka. Komponen ini merupakan pengembangan dari alat penampil 7 *segment*. Alat penampil 7 *segment* sangat terbatas dan hanya mampu menampilkan angka saja. *Alphanumeric display* hanya menambahkan 9 *segment* pada *display* 7 *segment* menjadi total 16 *segment*. Sistem memerlukan komponen ini sebagai penampil tanggal dan jam. Komponen ini lebih mudah digunakan dibandingkan dengan LED-Matrix. Kelemahan komponen ini adalah pada keterbatasan karakter yang bisa ditampilkan yaitu hanya bisa menampilkan huruf (terbatas hanya huruf kapital) dan angka. Adapun satu-satunya tanda baca yang bisa ditampilkan hanyalah berupa tanda titik (.). Kondisi seperti ini membuat komponen ini cocok digunakan untuk menampilkan tanggal dan jam.

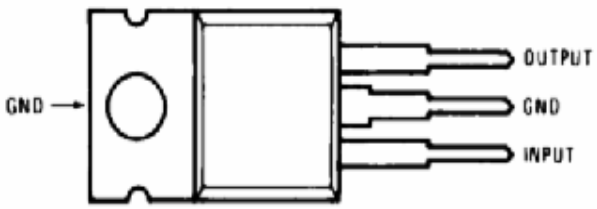
Gambar 11 memperlihatkan bentuk fisik komponen *Alphanumeric Display*. Konfigurasi pin komponen dapat dilihat pada Gambar 12.



Gambar 12. Konfigurasi pin *alphanumeric display*

Pada rancangan ini, membutuhkan catu daya dengan tegangan DC sebesar +5Volt yang stabil. Untuk mendapatkan tegangan yang stabil maka diperlukan sebuah IC yang mampu meregulasi tegangan DC +5V. IC regulator dapat mengurangi *ripple* pada tegangan keluaran dari rangkaian penyearah tegangan.

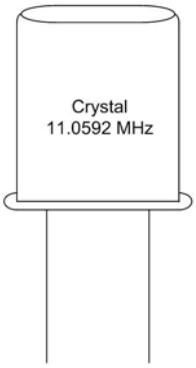
IC regulator yang digunakan agar diperoleh tegangan DC +5V yakni IC regulator 7805. Konfigurasi pin serta jumlah pin dari regulator tersebut berjumlah 3 buah *pin*. IC regulator ini mampu mengeluarkan arus hingga 1 Ampere, disertai pula dengan pembatas arus guna membatasi arus keluaran puncak dari IC ini agar aman. Gambar 13 berikut ini merupakan gambar dari IC regulator 7805.



Gambar 13. Konfigurasi pin IC regulator 7805

Komponen *Crystal Oscillator* digunakan untuk menghasilkan osilasi atau sebagai osilator. Kristal yang digunakan memiliki nilai frekuensi osilasi 11.0592 MHz. Frekuensi osilasi ini menentukan kecepatan siklus mesin dalam mikrokontroller. Pemilihan komponen ini sangat berpengaruh pada kecepatan *scanning* LED-Matrix. Semua mikrokontroler yang digunakan dalam sistem ini menggunakan kristal dengan frekuensi yang seragam. Bisa dikatakan bahwa dalam sistem ini semua mikrokontroler bekerja pada frekuensi yang sama.

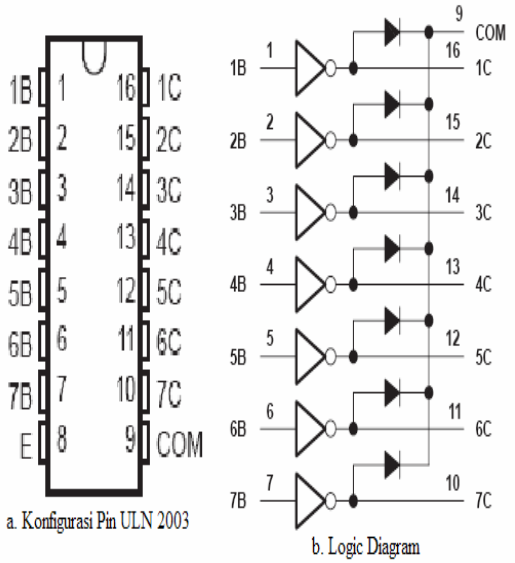
Baud-rate komunikasi serial sangat ditentukan oleh nilai kristal ini karena berpengaruh pada kemampuan mikrokontroler membaca perubahan bit pada jalur komunikasi serial. Secara tidak langsung hal ini mempengaruhi kecepatan transmisi data. Adapun gambar dari kristal adalah sebagai berikut :



Gambar 14. *Crystal Oscillator* 11,059MHz

Port I/O mikrokontroler digunakan untuk menyalakan LED yang ada pada LED-Matrix. *Port I/O* mikrokontroler hanya mampu mengeluarkan arus sebesar 15,0 mA. Untuk menyalakan satu LED dalam paket LED-Matrix dibutuhkan arus minimal 10,0 mA (kondisi LED menyala redup). Implementasi mikrokontroler dalam sistem ini mengharuskan satu *port I/O* mampu untuk menyalakan maksimal 5 LED secara bersamaan.

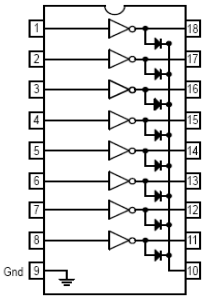
Supaya arus keluaran mikrokontroler mampu menyalakan 5 LED bersamaan perlu menggunakan penguat arus. Penguat arus yang digunakan dalam sistem ini adalah berupa IC ULN seri 2003 untuk driver LED-Matrix. IC ini dipilih karena bisa digunakan secara langsung dengan mikrokontroler. IC ini menggunakan rangkaian darlington sebagai penguat arus yang penguatannya mampu mencapai 500 mA. Dalam satu IC terdapat 7 pasangan darlington yang digunakan untuk mengendalikan 7 baris dalam LED-Matrix. Gambaran konfogurasi pin dan diagram logika ULN 2003 seperti pada Gambar 15.



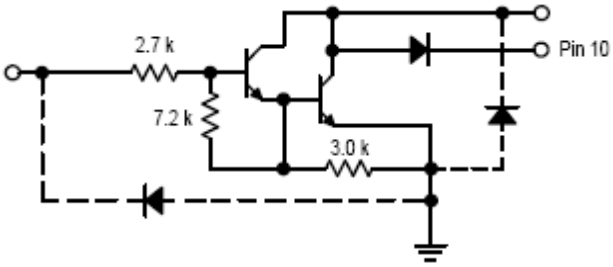
Gambar 15. Konfigurasi Pin dan *Logic Diagram* ULN 2003

IC ULN seri 2803 digunakan sebagai driver *alphanumeric display*. Perbedaan kedua seri ini hanya pada jumlah pasangan darlington dimana pada seri 2803 ini memiliki 8 pasang rangkaian darlington. Penggunaan pasangan darlington disesuaikan dengan jumlah *alphanumeric display* yang digunakan. Gambaran konfigurasi pin pada ULN 2803 yang digunakan dapat dilihat pada Gambar 16 dan gambar rangkaian darlington yang digunakan dapat dilihat pada Gambar 17.

Komputer merupakan unit yang paling vital dalam sistem ini. Dalam komputer terdapat *software* yang melakukan penjadwalan otomatis dan *database* yang menyimpan data yang digunakan oleh sistem. Jika komputer tidak mampu menjalankan software pada sistem ini maka akan terjadi kegagalan sistem secara total.



Gambar 16. Konfigurasi Pin dan Koneksi Pin Dalam ULN2803



Gambar 17. Rangkaian Darlington yang Digunakan ULN 2003/2803

Penentuan sistem minimum untuk komputer yang digunakan agar bisa menjalankan sistem dengan baik adalah ;

- Menggunakan sistem operasi Microsoft Windows XP Service Pack 2
- Memori RAM minimum 256MB, dianjurkan menggunakan 512MB RAM
- Prosesor minimum menggunakan Intel Pentium 4 2.8GHz dengan kemampuan *hardware Hyper Threading* atau prosesor yang setara.
- Menggunakan *Java Runtime Environment* 1.6
- Menggunakan sistem database MySQL Server 5.0

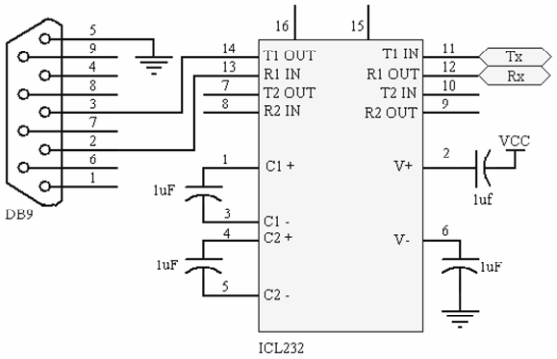
Realisasi Rancangan

Setelah mengumpulkan teori-teori pendukung sistem, realisasi rancangan dapat dilakukan sepenuhnya. Pembahasan realisasi rancangan sistem ini dimulai dari pembahasan realisasi modul-modul perangkat keras kemudian dilanjutkan dengan pembahasan modul perangkat lunak.

Perangkat keras terbagi atas beberapa modul antara lain modul *Hardware Interface* dan modul *Display Driver Circuit*. Perangkat lunak terdiri dari dua bagian yaitu bagian *Software Interface* dan bagian *Database*.

Realisasi komunikasi RS-232 pada modul *hardware interface*

Hubungan *hardware* dengan PC adalah melalui serial port yang mengadopsi standar RS-232 yang umumnya tersedia pada setiap PC. Namun standar RS-232 mempunyai *level* tegangan yang berbeda dengan format data digital TTL. Oleh sebab itu untuk menghubungkan alat dengan PC dibutuhkan rangkaian konverter yang bisa menyesuaikan data TTL ke RS 232 dan sebaliknya. Rangkaian ini digunakan untuk menyelaraskan beda *level* tegangan RS 232 dengan *level* tegangan TTL. Berikut adalah gambar rangkaian *interface* RS 232.



Gambar 18. Rangkaian interface RS-232

Penggunaan dan fungsi pin-pinnya adalah sebagai berikut :

- C1+ dan C1- dihubungkan ke elco 1 µF untuk menaikkan tegangan internal.
- C2+ dan C2- juga dihubungkan ke elco 1 µF sebagai pembalik tegangan. $T1_{in}$ berfungsi sebagai *transmitter* 1 TTL *input* dengan tambahan resistor *pull-up* sebesar 400 K pada V_{cc}.
- $R1_{out}$ berfungsi sebagai *receiver* 1 TTL *output*.

- V_{s+} dihubungkan ke elco 1µF lalu ke V_{cc} agar dapat menghasilkan tegangan *supply* sebesar +10 Volt.
- V_{s-} dihubungkan ke elco 1µF agar dapat menghasilkan tegangan *supply* sebesar –10 Volt selanjutnya dihubungkan ke *ground*.
- $T1_{out}$ dihubungkan ke pin 2 pada konektor com serial yang berfungsi sebagai *transmitter output* RS232 dengan tegangan ±10Volt.
- $R1_{in}$ dihubungkan pada pin konektor com serial yang akan berfungsi sebagai *receiver* data 1 *input* dengan tambahan 3 resistor *pull-down* 5 K ke *ground*.

IC *converter* yang digunakan pada rangkaian ini adalah IC MAX 232 yang umum digunakan.

Realisasi mikrokontroler pada modul *display driver circuit*

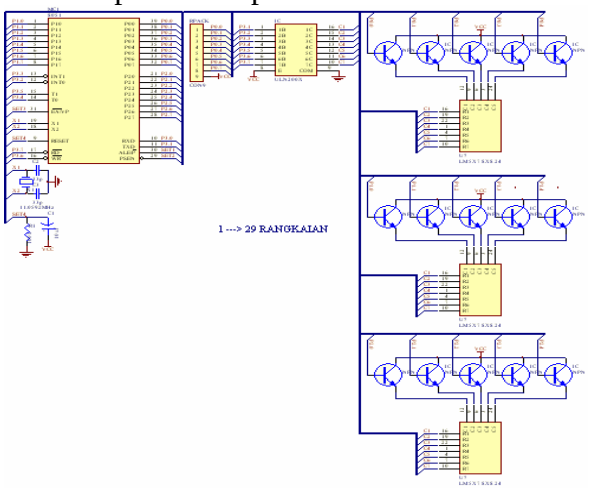
Pada modul ini, peranan mikrokontroler adalah sebagai driver untuk mengatur dua jenis display yaitu LED-Matrix dan *Alphanumeric Display*. Sebagai driver miktokontroler ini melakukan tugas sebagai *buffer* data yang dikirim dari PC. Setelah data kiriman PC di-*buffer*, mikrokontroler ini akan melakukan *scanning* data yang diterjemahkan dalam bentuk font.

Rangkaian realisasi mikrokontroler sebagai *driver LED-Matrix display* merupakan rangkaian pengontrol utama tiga paket LED-Matrix. Fungsinya mengolah data yang dikirimkan oleh komputer kemudian data tersebut dikirim ke rangkaian driver LED-Matrix dan mengatur aktivitas *scanning*. Mikrokontroler ini membaca data ASCII dan menerjemahkannya dalam bentuk font yang sudah ditentukan sesuai huruf yang akan ditampilkan.

Satu mikrokontroler dalam rangkaian ini mampu mengendalikan 3 LED-Matrix. Dengan penguatan arus menggunakan IC ULN 2003 pada bus *cathode* LED-Matrix memungkinkan satu pin pada satu port (menggunakan pin P3.1 sampai P3.7) mampu menyalakan ± 15 LED secara bersamaan dalam keadaan nyala terang. Pin-pin yang digunakan yaitu:

1. P0 merupakan *output* ke anoda led matrik kedua.
2. P1 merupakan *output* ke anoda led matrik ketiga.
3. P2 merupakan *output* ke anoda led matrik pertama.
4. P3.0 digunakan untuk komunikasi serial (RX) sedangkan P3.1 sampai P3.7 tersambung dengan IC ULN 2003 yang akan digunakan sebagai bus katoda pada tiap led matrik.
5. Kaki pin 9 (RST) dihubungkan ke rangkaian Reset untuk mereset *hardware*. Baik secara *software* ataupun manual. Maka pin RST ini harus berlogik 1 (aktif *high*).
6. Kaki pin 18 (XTAL 2) dan kaki pin (XTAL 1) dihubungkan ke rangkaian Osilator Kristal untuk memberikan *clock input*-an yang dibutuhkan oleh mikrokontroler. Dalam hal ini kristal yang digunakan adalah 11.0592 MHz agar mikrokontroler bekerja dengan kecepatan maksimum.
7. Kaki pin 20 (GND) dihubungkan ke *ground*.
8. Kaki pin 40 (V_{cc}) dihubungkan ke sumber tegangan 5 Volt DC.

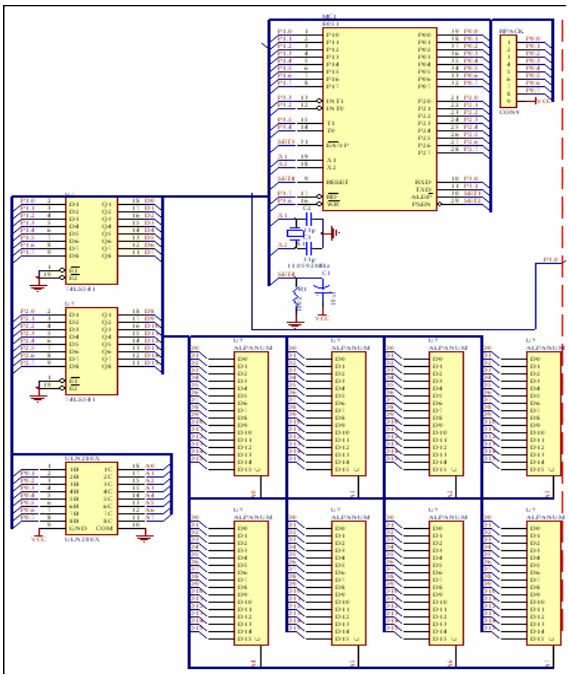
Gambar rangkaian mikrokontroler *driver* dapat dilihat pada Gambar 19.



Gambar 19. Rangkaian mikrokontroler *driver* LED-Matrix

Rangkaian Realisasi Mikrokontroler Sebagai *Driver Alphanumeric Display* merupakan rangkaian pengontrol utama duabelas *alphanumeric display*. Fungsinya mengolah data yang dikirimkan oleh komputer kemudian data tersebut dikirim ke rangkaian driver *alphanumeric*. Rangkaian *Alphanumeric Display driver* dapat dilihat pada Gambar 20. Pin-pin yang digunakan yaitu:

1. P0 merupakan *output* ke katoda masing-masing alphanumeric melalui IC UNL280X yang berfungsi sebagai penguat arus.
2. P1 dan P2 merupakan *output* ke anoda alphanumerik melalui IC 74LS541 sebagai line drivernya.
3. P2 merupakan *output* ke anoda led matrik pertama.
4. P3 digunakan untuk komunikasi serial (RX) dengan PC.



Gambar 20. Rangkaian mikrokontroler driver *alphanumeric display*

5. Kaki pin 9 (RST) dihubungkan ke rangkaian Reset untuk mereset *hardware*. Baik secara *software* ataupun manual. Maka pin RST ini harus berlogik 1 (aktif *high*).
6. Kaki pin 18 (XTAL 2) dan kaki pin (XTAL 1) dihubungkan ke rangkaian Osilator Kristal untuk memberikan *clock input*-an yang dibutuhkan oleh mikrokontroler. Dalam hal ini kristal yang digunakan adalah 11.0592 MHz agar mikrokontroler bekerja dengan kecepatan maksimum.
7. Kaki pin 20 (GND) dihubungkan ke *ground*.
8. Kaki pin 40 (V_{cc}) dihubungkan ke sumber tegangan 5 Volt DC.

Realisasi perangkat lunak

Perangkat lunak dalam sistem ini terbagi dua yaitu bagian *user interface* yang disebut sebagai *Software Interface*. Bagian *Software Interface* dirancang sebagai antar-muka bagi pengguna sistem. Perangkat lunak yang satu lagi antara lain adalah sistem *database*. Fungsi dari sistem *database* adalah sebagai penampung informasi yang diperlukan.

Rancangan sistem *database* disusun sedemikian rupa sehingga pengembangan *scalability* sistem dapat dilakukan di *database*. Pengembangan seperti penambahan dosen dan matakuliah atau penambahan ruangan dan peralatan.

Software Interface dikembangkan menggunakan bahasa pemrograman Java. Versi Java yang digunakan adalah versi *Java Developement Kit* (JDK) 1.6 yang mendukung konsep *Object Oriented Programming* (OOP). *Programming Environment* (*Integrated Developement Environment* (IDE)) yang digunakan adalah NetBeans IDE versi 5.5.

Software interface dalam sistem ini memiliki tiga fungsi utama. Yang pertama adalah sebagai *user interface* bagi operator dalam menggunakan sistem ini. Fungsi kedua adalah sebagai pembuat jadwal otomatis dengan menggunakan sistem penjadwalan otomatis. Karena sistem ini menggunakan *database* sebagai penyimpanan *data*, maka diperlukan fungsi ke tiga yaitu sebagai *database maintenance system*. Sebagai *user*

interface, *software* ini dikembangkan menggunakan *library user interface* yang disediakan oleh Java yaitu javax.swing *library*. *User interface* ini menyediakan berbagai fungsi *software* dalam bentuk *button* dan menampilkan data dalam bentuk tampilan yang dapat dimengerti oleh *user*. Adapun tampilan *user interface* dapat dilihat pada Gambar 21.

Gambar 21. Tampilan *user interface*

Sistem penjadwalan otomatis yang terdapat dalam sistem ini diimplementasikan dalam bentuk *framework* untuk memudahkan pengembangan dan penggunaannya. *Class hierarchy* yang tersusun didalam *framework* JSched terbagi dalam beberapa paket *class*.

Paket *Graphical User Interface* (GUI) (jsched.gui) berisi berbagai *class* yang diperlukan untuk membangun *user interface*. Paket ini banyak menggunakan *library* javax.swing untuk membangun *Graphical User Interface* (GUI). Semua *class* yang ada dalam paket ini merupakan turunan (*inherit*) dari *class* yang ada dalam *library javax.swing*.

Penggunaan *database* pada sistem ini mengharuskan pembuatan *class* untuk merepresentasikan *entity* yang terdapat dalam *table database*. Tujuan pengembangan *entity class* adalah agar data dari *database* dapat digunakan dalam *program*. Paket yang berisi berbagai *class* untuk manipulasi data dan semua *entity class* yang diperlukan ini diberi nama paket jsched.data. Paket ini tidak memiliki *hierarchy*, hanya gambaran *relation* antar *class* yang ada.

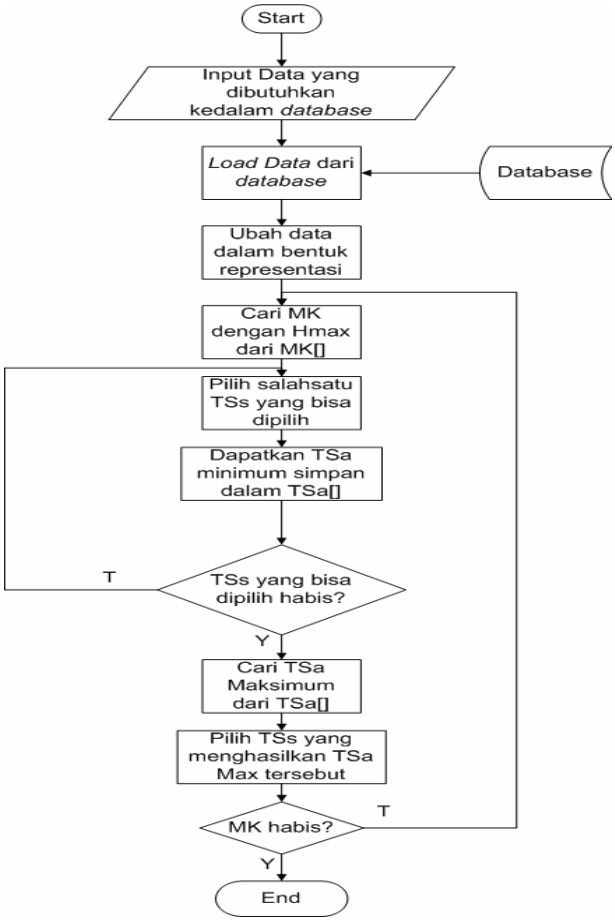
Untuk memproses data menggunakan *sistem searching* pada AI, perlu merepresentasikan data dalam bentuk yang disesuaikan dengan algoritma *searching* yang digunakan. Representasi data akan menambahkan fungsi-fungsi yang diperlukan oleh algoritma *searching*. Paket yang digunakan untuk representasi data berisi berbagai *class* yang merupakan turunan dari *entity class* yang ada pada paket data.

Paket lainnya berisi *class* tambahan yang diperlukan untuk memperluas kemampuan sistem dan menyederhanakan *problem definition* seperti implementasi *timeslot* dan yang lainnya dinamakan jsched.util. Dalam paket ini terdapat implementasi penjadwalan otomatis yang dijalankan dalam satu *thread* supaya tidak menggangu proses yang lainnya yang terdapat dalam sistem. Algoritma penjadwalan (algoritma *searching* dan perhitungan *heuristic*) diimplementasikan dalam satu *class* yang dinamakan *Scheduler*. Diagram alir algoritma penjadwalan dapat dilihat pada Gambar 22.

Paket-paket yang telah dibahas sudah cukup mewakili seluruh sistem bagian perangkat lunak. Namun sistem ini menggunakan perangkat keras berupa *display* untuk menampilkan hasil jadwal yang dibuat. Maka dari itu diperlukan fungsi yang mampu mengirim data jadwal yang sudah dibuat ke *hardware* berupa *display*.

Fungsi-fungsi ini dikelompokkan dalam satu paket jsched.io. Paket ini berisi inisialisasi *com port*.

Implementasi pengiriman data jadwal menggunakan bentuk *thread* supaya pengiriman data dapat dilakukan terus menerus dan *update* jadwal dapat dilakukan secara *real-time*. Selain itu juga diimplementasikan satu *class* untuk merepresentasikan jadwal kedalam format yang bisa dikirim ke *hardware*.



Gambar 22. Diagram alir algoritma penjadwalan otomatis

Dalam sistem penjadwalan otomatis, digunakan teknik *searching* AI yang menggabungkan dua teknik. Teknik pertama adalah penggunaan fungsi heuristik (heuristic function) *h(n)* untuk mengurutkan matakuliah mana yang sebaiknya dijadwalkan lebih dulu. Dalam istilah AI fungsi heuristik digunakan untuk mencari most constrained variable. Fungsi utama dari *h(n)* adalah untuk memandu searching supaya lebih terarah ke solusi yang dicari. Fungsi yang digunakan adalah jumlah *timeslot* yang kosong pada tabel jadwal dibagi dengan jumlah *timeslot* yang kosong pada data *availability* dosen. Dalam rumusan matematis dapat digambarkan sebagai berikut;

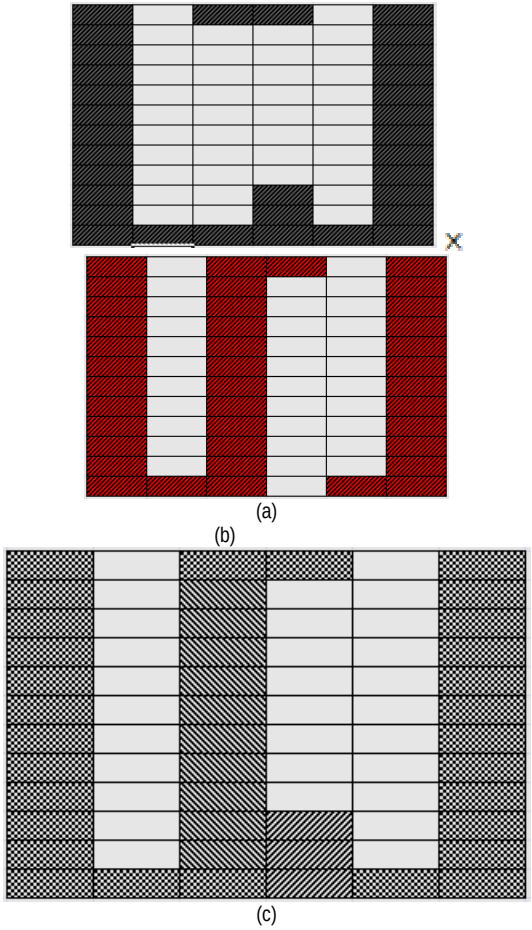
$$h(n) = \frac{\sum TS(s)}{\sum TS(a)}$$

Dimana $\sum TS(s)$ adalah jumlah *timeslot* yang kosong pada tabel jadwal, dan $\sum TS(a)$ merupakan jumlah *timeslot* yang kosong pada data *availability* dosen.

Teknik kedua yang digunakan dalam sistem ini adalah teknik *forward checking* yang dapat memilih *assignment* terbaik. Dalam sistem ini *assignment* terbaik yang bisa dipilih adalah *assignment* yang menimbulkan pengaruh minimal terhadap matakuliah yang lain. Tujuan *forward checking* adalah untuk meminimalkan kemungkinan *assignment* yang salah yang dapat menyebabkan *searching* menemukan jalan buntu.

Teknik *forward checking* diimplementasikan dengan cara melakukan *assignment* pada setiap *timeslot* yang memungkinkan dalam tabel jadwal. Setelah melakukan *assignment*, akan dilihat pengaruh dari setiap *assignment* terhadap perubahan data *availability* pada semua matakuliah yang ada. Hal ini dilakukan dengan melakukan *masking* tabel jadwal terhadap tabel *availability*. Jumlah timeslot yang bisa diisi setelah *masking* inilah yang dijadikan ukuran untuk mendapatkan *best choice*. Berikut ini adalah gambaran teknik *masking* yang digunakan.

Pada Gambar 23, gambar (a) menunjukkan *timeslot* yang *available* pada tabel jadwal. Arsiran hitam menunjukkan *timeslot* yang sudah terisi. Gambar (b) menunjukkan *timeslot* yang ada pada *availability*. Arsiran merah menunjukkan *timeslot* yang tidak termasuk dalam *availability*. Gambar (c) menunjukkan hasil *masking* dari gambar (a) dan gambar (b). Daerah arsiran menunjukkan *timeslot* yang tidak terjangkau. Dari gambar tersebut, jumlah *timeslot* yang tersedia setelah *assignment* adalah 27. Perhitungan seperti ini dilakukan untuk semua mata kuliah yang ada, kemudian ambil nilai terkecil untuk *assignment* ini. Lakukan hal ini pada semua *assignment* yang memungkinkan. Dari semua *assignment* tersebut dipilih nilai *assignment* terbesar.



Gambar 23. Teknik *masking timeslot*

Realisasi sistem *database*

Sistem *database* dibuat berdasarkan model relasi yang dibuat pada bab sebelumnya. Adapun tambahan yang ditambahkan hanyalah tabel-tabel pendukung saja. Seperti tabel untuk menyimpan data semester berjalan yang bisa digunakan untuk menentukan matakuliah yang dapat dimasukkan dalam penjadwalan.

Sistem *database* yang digunakan adalah MySQL yang menggunakan sintaks SQL (*Structured Query Language*). MySQL menggunakan bahasa *query* yang mudah dimengerti dalam sintaks SQL. Salah satu kelebihannya adalah kemampuan mendukung *relational database*. Kemampuan ini sangat dibutuhkan oleh sistem ini karena data-data yang ada pada sistem penjadwalan saling terkait satu dengan lainnya.

REALISASI RANCANGAN SISTEM

Realisasi Rancangan Perangkat *Display*

Perangkat *display* yang diimplementasikan menggabungkan rangkaian pada *hardware interface* dan *display driver circuit*. Penggabungan ini tidak sulit karena hanya menyambung bus dari *interface* RS-232. Keseluruhan paket *display* akan dibuat dalam bentuk kotak persegi panjang.

Rangkaian elektronika dibuat per 3 huruf LED-Matrix dalam satu *Printed Circuit Board* (PCB). Sedangkan untuk rangkaian *driver alphanumeric display* dibuat dalam satu board PCB. *Unit display* ini tidak menyediakan tombol *on/off* untuk mematikan atau menghidupkan perangkat ini. *Power supply* unit akan digabung langsung dengan unit ini. Selain itu tidak disediakan *memory* untuk menyimpan data jadwal yang dikirim dari komputer PC. Hal ini menyebabkan jika komputer PC gagal, maka sistem ini akan ikut gagal.

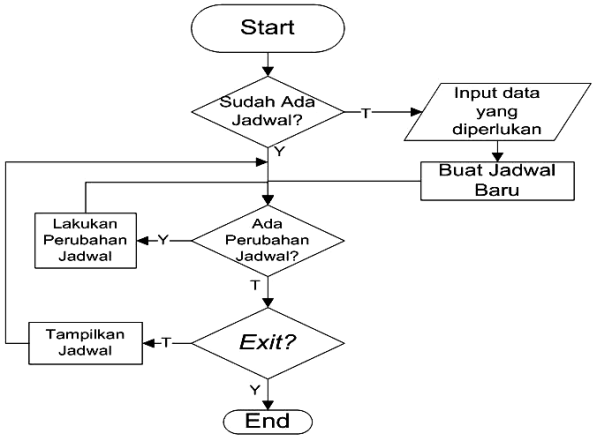
Realisasi Rancangan Keseluruhan Sistem

Rancangan ini terbagi atas dua bagian utama yaitu bagian perangkat keras yang berupa *display* dan perangkat lunak yang menjalankan sistem penjadwalan. Perangkat keras berfungsi menampilkan jadwal yang dibuat oleh sistem penjadwalan. Perangkat lunak berisi sistem penjadwalan otomatis dan *interface* bagi operator untuk bisa menggunakan sistem dengan baik.

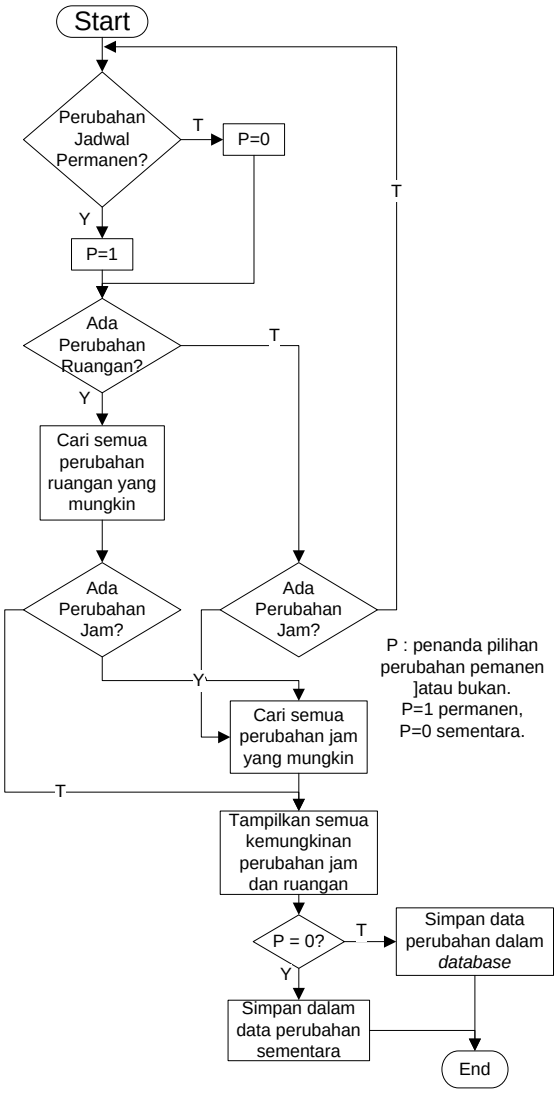
Proses kerja sistem dimulai dari pembuatan jadwal terlebih dahulu. Proses pembuatan jadwal itu sendiri diatur oleh sistem penjadwalan otomatis. Namun sebelum jadwal dapat dibuat, operator harus mengisi *database* yang dibutuhkan oleh. Setelah pembuatan jadwal selesai, sistem akan mengirimkan data jadwal untuk ditampilkan oleh unit *display*. Dalam kondisi ini, keadaan akan terus bertahan sampai adanya permintaan perubahan jadwal. Permintaan perubahan jadwal diajukan oleh *low level user*

seperti dosen. Jika terjadi perubahan jadwal, maka harus ditentukan dulu apakah perubahan jadwal ini permanen atau tidak. Jika permanen akan terjadi perubahan *database* dan jika tidak maka tidak perlu perubahan *database*.

Operator akan mendapat saran perubahan yang mungkin dari sistem ini. Saran perubahan ini tidak selalu harus diambil oleh operator. Dalam kondisi perubahan jadwal ini, yang aktif hanyalah perubahan jadwal manual. Sistem penjadwalan otomatis hanya menampilkan kemungkinan perubahan yang paling minimum. Jadi keputusan tetap berada ditangan operator. Diagram alir cara kerja sistem dapat dilihat pada Gambar 24. Diagram alir proses perubahan jadwal dapat dilihat pada Gambar 25.



Gambar 24. Diagram alir cara kerja keseluruhan sistem



Gambar 25. Diagram alir proses perubahan jadwal

KESIMPULAN DAN SARAN

Kesimpulan

Berdasarkan realisasi dan hasil pengujian yang dilakukan terhadap rancangan Tugas Akhir ini, maka dapat ditarik kesimpulan sebagai berikut:

- Sistem penjadwalan otomatis mampu melakukan penjadwalan dengan baik terbatas pada kemampuan sistem dan formulasi masalah penjadwalan ini.
- Sistem membutuhkan memori yang lebih besar jika ingin menjadwalkan matakuliah dalam jumlah yang lebih banyak.
- *Display* LED-Matriks kurang efisien jika diterapkan dalam rancangan ini karena keterbatasan tampilan dan memakan biaya yang cukup besar.

Saran

Bagi yang ingin mengembangkan rancangan alat ini lebih lanjut, disarankan agar ;

- Pengembangan sistem penjadwalan otomatis dengan pendekatan *Expert Sistem*. Meskipun perlu membuat sebuah *knowledge base* yang besar, tapi teknik

searching sudah disederhanakan sehingga kebutuhan memori bisa diminimalkan.

- Jika ingin mengembangkan *display unit*, gunakan PCB *double layer through hole* agar dimensi model peragaan dapat dikecilkan.
- Gunakan mikrokontroler yang memiliki ukuran memori internal lebih besar supaya data kiriman PC dapat di tampung sementara dalam mikrokontroler. Dengan demikian sistem tidak lagi bergantung secara total pada unit PC.

Daftar Pustaka

D. Thomas, A. James, Y. Aloimonos, *Artificial Intelligence Theory And Practice;* Redwood City-California: The Benjamin Cummings Publishing Company, 1995, ch:7 pp:298

D. V. Ajay, *Microcontrollers Theory And Applications*, New Delhi: McGraw-Hill Publishing Company Limited, 2005, ch:3 pp:26, ch:7 pp:85

R. F. Richard, *Java in 60 Minutes a Day*, Indianapolis: Wiley Publishing Inc., 2003, chap:15 pp:503, ch:18 pp:629

S. Robert, *Algorithms In Java*, Third edition Part 5 Graph Algorithms ; Addison-Wesley, 2003,ch:18 pp:87

S. J. Russell, P. Norvig, *Artificial Intelligence A Modern Approach*; New Jersey:Prentice Hall, 1992,ch:4 pp:101

http://www.ento.vt.edu/Publications/ NDS-Paper- fn.html**,** Diakses pada tanggal 24 April 2007

http://www.aaai.org/papers.asp, Diakses pada tanggal 25 April 2007