
USULAN TUGAS AKHIR

1. IDENTITAS PENGUSUL

Nama : **Himmatul Azizah**
NRP : **5107 100 048**
Dosen Wali : **Dr. Ir. Joko Lianto, M.Sc**

2. JUDUL TUGAS AKHIR

Ekstraksi Metadata dari Dokumen Software Requirements Specification (SRS)

3. LATAR BELAKANG

Dokumen *Software Requirements Specification* (SRS) merupakan sebuah deskripsi lengkap dari *behavior* sebuah sistem yang akan dikembangkan. Dalam dokumen ini berisi rincian kebutuhan fungsionalitas dan non-fungsionalitas. Dokumen SRS ini berfungsi untuk mencatat semua kebutuhan pengguna perangkat lunak, sebagai kontrol saat proses pengembangan perangkat lunak dilakukan, sehingga setiap tahapan pengerjaan pengembangan sesuai dengan yang diharapkan, sebagai acuan pada saat pengujian dilakukan sehingga hasil akhir sesuai dengan yang dibutuhkan, sebagai pedoman jika terdapat perbedaan antara pengguna dengan pengembang sistem terhadap hasil pengembangan perangkat lunak, dan sebagai bukti bahwa pengembang dalam hal ini sistem analis telah melakukan tahap *software requirement analysis*.

Saat sebuah perangkat lunak mulai dikembangkan, pengembang perangkat lunak wajib membuat dokumen SRS. Dokumen ini harus ditulis dengan baik dan dimengerti secara menyeluruh oleh pengguna dan pengembang perangkat lunak. Pada kenyataannya, pengembang perangkat lunak terkadang sulit untuk menganalisa sebuah dokumen SRS yang digunakan sebagai pedoman saat pembuatan perangkat lunak. Sering kali para pengembang perangkat lunak kurang hati-hati dan tidak teliti, sehingga mengakibatkan terjadinya kesalahan analisa yang sungguh menimbulkan banyak kerugian. Kesalahan analisa dokumen SRS yang diketahui ketika sudah memasuki penulisan kode, atau pengujian, bahkan hampir pada tahap penyelesaian, adalah malapetaka besar bagi sebuah kelompok pembuat perangkat lunak. Biaya dan waktu yang diperlukan menjadi banyak yang tersia-sia. Biaya yang diperlukan untuk memperbaiki sebuah kesalahan karena analisa kebutuhan yang tidak benar, bisa menjadi dua puluh lima kali lipat, jika kesalahan tersebut ditemukan pada tahap pengujian fungsi perangkat lunak.

Seringkali para pengembang perangkat lunak harus berulang kali menemui pengguna perangkat lunak yang sedang dikembangkan karena kesalahan analisa. Terkadang pula pengguna memberi waktu pengerjaan yang sangat singkat sehingga perangkat lunak yang dihasilkan menjadi tidak sempurna.

Untuk meminimalisir masalah-masalah tersebut, *Departement of Computer Science and Software Engineering, Concordia University, Montreal, Canada* melakukan penelitian mengenai perangkat lunak yang dapat membantu pengembang perangkat lunak dalam menyelesaikan masalah tersebut berupa sebuah perangkat lunak yang dapat menghasilkan *high level contextual view* pada aktor dan layanan dari sebuah sistem

perangkat lunak secara otomatis. Perangkat lunak ini disebut *Requirement Engineering Assistance Diagnostic* (READ) proyek. Pada perangkat lunak ini, pengembang perangkat lunak dapat terbantu pada langkah awal pemodelan use case. Hasil evaluasi dari READ proyek yang dilakukan oleh Shadi Moradi Seresht dan Olga Ormandjieva yang merupakan bagian dari *Departement of Computer Science and Software Engineering, Concordia University, Montreal, Canada* itu sendiri adalah perangkat lunak ini dapat mendeteksi ambiguitas pada level pemahaman awal. Sekarang ini masih diteliti visualisasi dari kebutuhan non-fungsional (NFR) yang secara otomatis diekstrak dan diklasifikasi dari teks input. Kedepannya NFR akan disoroti untuk meningkatkan visibilitasnya, dan secara tidak langsung terhubung dengan kebutuhan fungsional, dan keduanya akan ditingkatkan melalui domain model dan CUCM (*Convert Use-Case Model*). Kendala dari perangkat lunak ini adalah ketidaktetapan pada dokumen teks yang besar sehingga menyebabkan dokumen tersebut sulit untuk diidentifikasi. [2]

Penelitian lain yang dilakukan oleh Giuseppe Lami dari Software Engineering Institute, Carnegie Mellon University pada tahun 2005 mengenai QuARS (*Quality Analyzer for Requirements Specifications*) yang merupakan perangkat lunak yang digunakan untuk menganalisa dokumen kebutuhan yang diambil dari proyek industri yang nyata. Pada perangkat lunak ini, dokumen kebutuhan dalam file format apapun dapat diproses. Kesempurnaan analisa tergantung dari kelengkapan dan keakurasian kamus yang terdapat pada perangkat lunak tersebut. Sekarang ini QuARS hanya dapat memproses teks dokumen dan kebutuhan yang tidak sempurna. [5]

Berangkat dari kekurangan aplikasi yang sudah ada di atas, pada tugas akhir ini penulis mengusulkan untuk membangun sebuah perangkat lunak yang dapat membantu pengembang perangkat lunak dalam mengekstraksi metadata dari sebuah dokumen SRS. Berbeda dengan penelitian sebelumnya yang inputnya berupa daftar kebutuhan, pada perangkat lunak yang akan dibangun kali ini inputnya berupa dokumen SRS. Dokumen SRS merupakan dokumen yang memiliki detail penjelasan keseluruhan aspek perangkat lunak. Didalamnya terdapat informasi perangkat lunak yang berupa teks dan gambar. Metadata dari dokumen SRS dapat membantu pengembang perangkat lunak dalam mempercepat pengerjaan perangkat lunak dan mengurangi biaya yang diperlukan selama proses pengembangan perangkat lunak. Dalam mengekstraksi metadata dari dokumen SRS, dibutuhkan algoritma *part-of-speech* untuk mengekstraksi metadata dokumen yang berupa teks dan algoritma *optical character recognition* untuk mengekstraksi metadata dokumen yang berupa gambar.

4. RUMUSAN MASALAH

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana mengubah dokumen SRS yang berbentuk word .docx menjadi file xml
2. Bagaimana mengaplikasikan algoritma *part-of-speech* dalam mengekstraksi metadata dari dokumen SRS yang berupa teks.
3. Bagaimana mengaplikasikan algoritma *optical character recognition* dalam mengekstraksi metadata dari dokumen SRS yang berupa gambar.

5. BATASAN MASALAH

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, diantaranya sebagai berikut:

1. Dokumen SRS yang akan dianalisa adalah dokumen SRS yang berbahasa Inggris.
2. Dokumen SRS ini merupakan file yang menggunakan Microsoft Word versi 2007 ke atas.
3. Hasilnya berupa metadata yang berbentuk file XML.

6. TUJUAN TUGAS AKHIR

Tujuan dari Tugas Akhir ini adalah membuat sebuah kakas bantu untuk mengekstraksi sebuah dokumen SRS.

7. MANFAAT TUGAS AKHIR

Manfaat yang diharapkan dari Tugas Akhir ini dari sisi pengembang perangkat lunak yaitu dapat membantu menghemat waktu pembuatan perangkat lunak, menganalisa kebutuhan yang diperlukan pada perangkat lunak, dan meminimalisir biaya yang dibutuhkan dalam membuat perangkat lunak. Sedangkan dari sisi pengguna yaitu pengguna tidak perlu lagi terlalu sering bertemu dengan pengembang perangkat lunak.

8. DASAR TEORI

Dasar teori yang mendukung penyelesaian Tugas Akhir ini diantaranya sebagai berikut :

a) Microsoft Word 2007

Microsoft Word (MS Word) merupakan program untuk mengolah kata. Program ini bisa digunakan untuk menulis dokumen misalnya karya tulis, skripsi, novel, dan sebagainya. Selain menulis dokumen, MS Word juga dapat digunakan untuk bekerja dengan tabel, menulis teks dengan kreasi, menyisipkan gambar, maupun yang lainnya.

b) *Software Requirements Specification (SRS)*

Software Requirements Specification (SRS), sebuah spesifikasi *requirement* atau kebutuhan untuk sebuah sistem perangkat lunak, adalah dokumen yang dibuat ketika sebuah perangkat lunak akan dikembangkan. Didalamnya terdapat detail penjelasan dari keseluruhan aspek dari sebuah perangkat lunak. Ketika sebuah perangkat lunak akan dikembangkan dan memiliki spesifikasi yang sedikit atau ketika sebuah sistem terlalu kompleks, dokumen SRS sangatlah dibutuhkan. [1]

c) Use Case Diagram

Menurut Ivar Jacobson, use case diagram secara sederhana merupakan bantuan untuk mendefinisikan apa yang ada diluar sistem (aktor) dan apa yang harus dilakukan oleh sistem (use case).[1] Didalam sebuah use case diagram terdapat dua komponen utama, yaitu aktor dan use case. Aktor adalah orang – orang yang menggunakan sebuah sistem yang sama, tapi memiliki kepentingan berbeda terhadap sistem tersebut. Aktor merepresentasikan orang yang menjalankan atau mengoperasikan sebuah sistem. Use case merupakan kebutuhan – kebutuhan yang dijalankan oleh sebuah sistem.

d) Metadata

Ketika sebuah dokumen dibuat menggunakan Microsoft Word, program menciptakan *binary file* dengan ekstensi “doc”. Pembuat program ini, memilih memasukkan kode biner ke dokumen untuk menyatakan *bold text*, kode biner yang

menyatakan *page break*, dan kode biner lainnya untuk menyatakan semua informasi yang disediakan oleh “doc” file. Kode yang dimasukkan ke dokumen adalah metadata atau data mengenai data.[4] Misalnya pada Microsoft Word, data aslinya berupa “a” dan metadatanya berupa “a should be in bold”.

Definisi sederhana dari metadata adalah data mengenai data. Metadata ini mengandung informasi mengenai isi dari suatu data yang dipakai untuk keperluan manajemen file / data itu nantinya dalam suatu basis data. Jika data tersebut dalam bentuk teks, metadatanya biasanya berupa keterangan mengenai nama ruas (field), panjang field, dan tipe fieldnya: integer, character, date, dll. Untuk jenis data gambar (image), metadata mengandung informasi mengenai siapa pemotretnya, kapan pemotretannya, dan setting kamera pada saat dilakukan pemotretan. Satu lagi untuk jenis data berupa kumpulan file, metadatanya adalah nama-nama file, tipe file, dan nama pengelola (administrator) dari file-file tersebut.

e) XML

XML (*Extensible Markup Language*) adalah bahasa yang sangat umum yang dapat ditemukan dimanapun pada Internet, tetapi juga merupakan bahasa yang sedang berkembang dewasa ini dengan aplikasi nyata yang berguna, khususnya untuk pengelolaan, tampilan, dan organisasi data. XML menyangkut tentang deskripsi dan struktur sebuah data, jadi sebelum mempelajari konsep dibalik XML, harus mengerti dahulu tentang bagaimana cara komputer menyimpan dan mengakses data. [4]

XML didesain untuk mampu menyimpan data secara ringkas dan mudah diatur. Kata kunci utama XML adalah data (jamak dari datum) yang jika diolah bisa memberikan informasi.

XML menyediakan suatu cara terstandarisasi namun bisa dimodifikasi untuk menggambarkan isi dari dokumen. Dengan sendirinya, XML dapat digunakan untuk menggambarkan sembarang view database, tetapi dengan suatu cara yang standar.

f) Algoritma *Part-of-speech tagging*

Part-of-speech tagging (POS *tagging* atau POST) , juga disebut *grammatical tagging* atau *word-category disambiguation*, merupakan proses menandai kata pada teks sebagai hubungan dari bagian tertentu pada dokumen, didasarkan pada kedua definisi, serta konteksnya yakni hubungan dengan kata-kata yang berdekatan dan terkait dalam kalimat, frase, atau paragraf. Bentuk sederhana dari pemakaian POS adalah pada anak kecil yang baru belajar mengidentifikasi kata benda, kata kerja, kata keterangan, kata sifat, dll. [8]

Terkadang ada beberapa kata yang merepresentasikan lebih dari satu bagian dalam dokumen di tempat yang berbeda. Contohnya adalah “*The sailor dogs the hatch*”. Kata “*dogs*” disini dapat merupakan kata benda jamak, dapat pula merupakan kata kerja. Penggunaan *grammatical tagging* dapat menunjukkan bahwa “*dogs*” adalah kata kerja, dan bukan kata benda jamak, karena salah satu dari kata-kata harus menjadi kata kerja utama, dan pembacaan kata benda kurang memungkinkan untuk berada setelah kata “*sailor*” (*sailor* !→ *dogs*). Analisis semantik kemudian dapat meramalkan bahwa “*sailor*” dan “*hatch*” mengimplikasikan “*dogs*” sebagai 1) dalam konteks bahari (*sailor*→<kata kerja>←*hatch*) dan 2) tindakan yang diterapkan pada objek “*hatch*” ([subjek] *dogs* → *hatch*). Dalam konteks ini, “*dogs*” adalah istilah bahari yang berarti “mengikatkan (penetasan kedap air) dengan aman; berlaku untuk mengikuti”.

g) Microsoft Office Document Imaging (MODI)

Document imaging adalah proses dari pemindaian sebuah dokumen, diubah menjadi sebuah gambar digital yang kemudian disimpan dalam CD, DVD, atau penyimpanan yang bersifat magnetis lainnya. Dengan Microsoft Office Document Imaging, sebuah dokumen dapat dipindai dan diubah bentuknya menjadi sebuah file berformat *Tagged Image File Format* (TIFF) atau *Microsoft Document Imaging Format* (MDI). MODI juga memberikan kemampuan untuk melakukan *Optical Character Recognition* (OCR). OCR dapat menerjemahkan gambar menjadi sebuah teks, seperti dokumen yang dipindai. Dengan menggunakan OCR, teks yang diakui pada sebuah gambar dapat dideteksi. [7]

Beberapa kegunaan MODI yang digunakan dalam tugas akhir ini adalah :

- Melakukan *Optical Character Recognition* (OCR) pada dokumen.
- Menyalin teks dan gambar dari sebuah dokumen dan melekatkan hasilnya pada sebuah program Office.

9. RINGKASAN TUGAS AKHIR

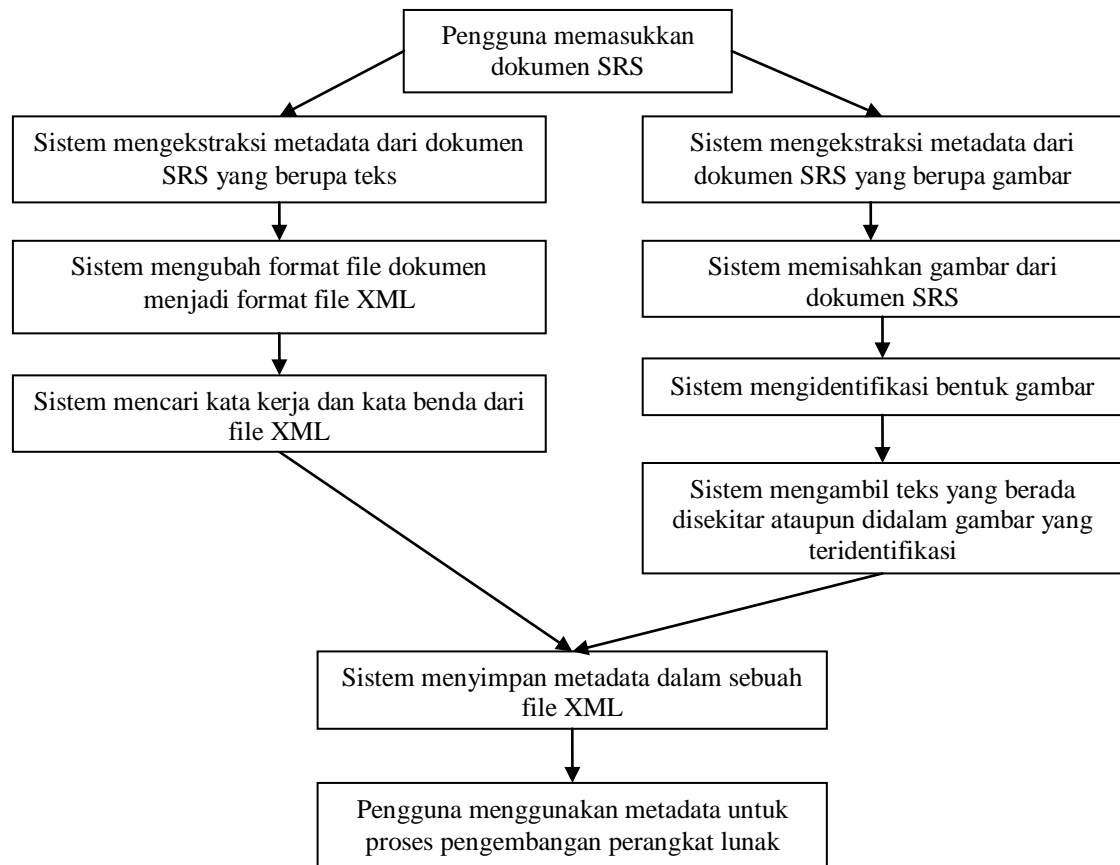
Pada tugas akhir ini, akan dibuat sebuah sistem yang dapat mengekstraksi metadata dari sebuah dokumen SRS. Dokumen SRS yang diproses harus berupa dokumen yang berbahasa Inggris dan memiliki file format Microsoft Word 2007 keatas. Hasil dari sistem ini adalah metadata dari dokumen SRS yang diharapkan dapat berguna dalam pengembangan perangkat lunak dan menghemat waktu pengerjaan sebuah perangkat lunak.

Pada sistem ini akan diberikan dua fitur, yaitu ekstraksi metadata yang berupa teks dan gambar. Pada fitur ekstraksi teks, sistem akan mengubah dokumen SRS tersebut menjadi sebuah XML file. File XML tersebut kemudian dipisah – pisah per bagian. Dari tiap bagian tersebut kemudian dicari tiap kata, mana yang merupakan kata benda sebagai kandidat aktor dan mana yang merupakan kata kerja sebagai kandidat fitur atau layanan pada perangkat lunak yang akan dikembangkan dengan menggunakan algoritma *part-of-speech*.

Pada fitur ekstraksi gambar, sistem akan mencari gambar yang terdapat pada dokumen SRS. Apabila sistem menemukan gambar di dalamnya maka sistem akan memisahkan gambar tersebut dari dokumen, kemudian gambar tersebut diidentifikasi bentuknya. Setelah menemukan bentuk yang diperkirakan sebagai aktor dan *use case*, sistem akan mengekstraksi teks dari gambar tersebut. Metode yang digunakan dalam mengekstraksi gambar dalam dokumen SRS adalah *optical character recognition*.

Keluaran dari tugas akhir ini adalah metadata dari dokumen SRS yang berupa kandidat aktor dan kandidat layanan atau fitur yang diberikan oleh perangkat lunak. Diharapkan nantinya dengan adanya sistem ini, maka para pengembang perangkat lunak dapat terbantu dalam menghemat waktu pengerjaan sebuah perangkat lunak.

Berikut ini merupakan alur dari aplikasi ini :



Gambar 1. Diagram Alur Sistem

10. METODOLOGI

Metodologi yang akan dilakukan dalam Tugas Akhir ini memiliki beberapa tahapan, di antaranya sebagai berikut:

1. Penyusunan Proposal Tugas Akhir

Tahap awal untuk memulai pengerjaan Tugas Akhir adalah penyusunan Proposal Tugas Akhir. Pada proposal ini, penulis mengajukan gagasan perancangan dan pengembangan sistem perangkat lunak yang dapat mengekstraksi dokumen SRS menjadi sebuah dokumen baru yang dapat membantu analis membuat sebuah use case diagram.

2. Studi Literatur

Tahap ini merupakan tahap pengumpulan informasi yang diperlukan untuk pengerjaan tugas akhir sekaligus mempelajarinya. Mulai dari pengumpulan literatur, diskusi, serta pemahaman topik tugas akhir di antaranya tentang :

- Metode untuk mengekstraksi sebuah SRS.
- Algoritma *optical character recognition* dan algoritma *optical character recognition* untuk membantu meringkas use case dari daftar kebutuhan.
- XML file format sebagai output dari sistem ini.

3. Analisis

Tahapan ini menganalisa algoritma yang akan digunakan untuk mengekstrak dokumen dan mengambil inti daftar kebutuhan perangkat lunak yang berguna dalam pembuatan use case, menganalisa data yang akan digunakan, menganalisa input yang akan diproses, dan menganalisa output.

4. Perancangan Sistem

Tahapan ini digunakan untuk merancang sistem, dimana di dalam sistem tersebut terdapat antarmuka pengguna, antarmuka pencarian kandidat aktor, dan antarmuka pencarian kandidat layanan.

5. Implementasi

Implementasi merupakan tahap untuk membangun perangkat lunak pengekraksi dokumen SRS menjadi sebuah dokumen baru yang dapat membantu analis membuat sebuah *use case diagram*. Sistem dibangun dengan berpedoman pada konsep-konsep yang sudah ditentukan pada tahap sebelumnya.

6. Pengujian dan Evaluasi

Pada tahap ini sistem telah selesai diimplemetasikan dan akan diuji. Pengujian dan evaluasi akan dilakukan dengan menguji apakah output dari perangkat lunak ini dapat membantu analis dalam membuat *use case diagram*.

7. Penyusunan Buku Tugas Akhir

Tahap terakhir merupakan penyusunan laporan yang memuat dokumentasi pelaksanaan tugas akhir yang mencakup seluruh konsep, teori, implementasi, serta hasil dari ujicoba dari aplikasi yang telah dibuat. Secara garis besar, Buku Tugas Akhir yang nantinya akan dibuat terdiri dari beberapa bagian antara lain :

1. Pendahuluan
 - 1.1 Latar Belakang
 - 1.2 Permasalahan
 - 1.3 Batasan Tugas Akhir
 - 1.4 Tujuan
 - 1.5 Metodologi
 - 1.6 Sistematika Penulisan
2. Tinjauan Pustaka
3. Desain dan Implementasi
4. Uji Coba dan Evaluasi
5. Kesimpulan dan Saran
6. Daftar Pustaka

11. JADWAL KEGIATAN TUGAS AKHIR

Tugas Akhir ini diharapkan bisa dikerjakan menurut jadwal sebagai berikut:

No .	Kegiatan	Bulan											
		Maret 2011				April 2011				Mei 2011			
1.	Penyusunan Proposal Tugas Akhir												
2.	Studi Literatur & Perancangan Sistem												
3.	Implementasi & Pembuatan Sistem												

[illegible]

12. DAFTAR PUSTAKA

- [2] Ahia, Airin dan Tabibi. 2008. *Extracting Summary – level Use – Cases Descriptions by Text-Partitioning*. 11th Workshop on Requirements Engineering.
- [3] Al-Hasbi, Kania. 2011. *RPL*. (<http://wmhasbi.wordpress.com/2010/12/24/rpl/>, diakses 10 Maret 2011)
- [4] Hunter, David, dkk. 2007. *Beginning XML 4th Edition*. Canada : Wiley Publishing, Inc.
- [5] Lami, Giuseppe. 2005. *QuARS: A Tool for Analyzing Requirements*. Pittsburgh, PA USA: Carnegie Mellon Software Engineering Institute.
- [7] Microsoft. *About Microsoft Office Document Imaging* (<http://office.microsoft.com/en-us/help/about-microsoft-office-document-imaging-HP001077103.aspx>, diakses 13 Maret 2011)
- [1] Pressman, Roger. 2010. *Software Engineering : A practitioner's Approach 7th Edition*. Newyork USA : Mc Graw Hill.
- [6] Saiint. 2010. *Analisis Kebutuhan Dalam Rekayasa Perangkat Lunak*. (<http://suryainformation.wordpress.com/2010/05/23/analisis-kebutuhan-dalam-rekayasa-perangkat-lunak/>, diakses 12 Maret 2011)
- [8] Wikipedia. 2011. *Part-of-speech tagging*. (http://en.wikipedia.org/wiki/Part-of-speech_tagging, diakses 10 Maret 2011)

LEMBAR PENGESAHAN

Surabaya, Maret 2011

Menyetujui,

Dosen Pembimbing I

Dosen Pembimbing II

Sarwosri, S. Kom, M. T.

NIP. 19760809 200112 2 001

**Daniel Oranova Siahaan, S.Kom, M.Sc.,
PD.Eng.**

NIP. 19741123 200604 1 001