

BAB IV

HASIL DAN PEMBAHASAN

Pada bab ini akan dibahas proses pengukuran dan penentuan kualitas desain program dengan menggunakan *tool* pengukuran dan penentuan peringkat yang telah dibahas pada bab sebelumnya. Terdapat lima buah sampel program yang digunakan dalam pembahasan bab ini. Seluruh aplikasi yang menjadi sampel untuk pengukuran kualitas ini merupakan aplikasi *open source* dan dapat diunduh melalui *website* dari masing-masing program.

Pembahasan pada bab ini terdiri dari tiga bagian. Pada bagian pertama akan dibahas terlebih dahulu mengenai masing-masing aplikasi yang dijadikan sebagai sampel program untuk percobaan pengukuran. Pada bagian berikutnya dibahas mengenai konfigurasi komponen penilaian dalam proses pengukuran kualitas desain. Bagian terakhir dari pembahasan di bab ini menampilkan hasil pengukuran kualitas desain dari sampel program beserta dengan evaluasi dari hasil pengukuran yang didapatkan.

Kelima sampel program yang digunakan sebagai dalam pengukuran kualitas desain adalah ADempiere ERP Business Suite, Freedom ERP, jAllInOne, Plazma dan Openbravo ERP. Pada sub bab berikut ini, kita akan membahas mengenai masing-masing aplikasi yang dijadikan sampel program pada proses pengukuran nanti.

4.1 SAMPEL PROGRAM

ADempiere ERP Business Suite

ADempiere adalah sebuah aplikasi bisnis berbasis *open source* yang cukup populer. Proyek ADempiere dimulai dari September 2006, merupakan pecahan dari aplikasi bisnis Compiere. Karena merupakan *fork* dari Compiere, ADempiere memiliki fungsionalitas dan tampilan program yang mirip dengan Compiere. Aplikasi yang tercakup pada ADempiere ini antara lain adalah Enterprise Resource Planning (ERP), Supply Chain Management (SCM), Customer Relationship Management (CRM), Financial Performance Analysis, Point Of Sales dan Web Store.

Aplikasi ini dikembangkan oleh komunitas dengan menggunakan teknologi *JBoss Application Server*. Proyek pengembangan Adempiere dijalankan dengan mengikuti pola pengembangan yang terdistribusi. Para pengembangnya terdiri dari individu-

individu yang saling berkomunikasi melalui Internet. Aplikasi ini sendiri di-*host* pada SourceForge (www.sourceforge.net) dan dapat diunduh dari <http://sourceforge.net/projects/adempiere/>.

Freedom ERP

Freedom merupakan salah satu aplikasi *free software* untuk manajemen bisnis. Aplikasi ini terdiri beberapa modul yaitu “Freedom Standard”, “Financial Freedom”, “Freedom Production”, “Freedom Telemarketing”, “Freedom POS” dan “Freedom Service”. Freedom Standard mencakup *feature* Financial Freedom dan Freedom Production. Aplikasi ini dapat diunduh dari alamat *web* <http://sourceforge.net/projects/freedom-erp/>.

jAllinOne

jAllinOne adalah aplikasi ERP/CRM berbasis Java, dikembangkan sebagai *Rich Internet Application* (RIA) dengan memanfaatkan protokol HTTP sebagai perantara antara aplikasi *client* dan *server*. Beberapa *feature* yang terdapat pada aplikasi ini adalah modul penjualan, pembelian, gudang, produksi, akuntansi, CRM dan beberapa *feature* lainnya. Proyek pengembangan aplikasi ini di-*host* pada <http://sourceforge.net/projects/jallinone/>.

Plazma

Plazma adalah aplikasi bisnis *open source*, modul-modul yang tersedia pada aplikasi ini adalah antara lain *Accounts Management*, *Contacts Management*, *Products Management*, *Sales Management*, *Tasks Management*, *Campaigns Management* dan *Analytical Reports*. Aplikasi ini dapat diunduh dari <http://sourceforge.net/projects/plazma/>.

Openbravo ERP

Openbravo merupakan sebuah aplikasi ERP yang berbasis *open source* yang didesain khusus untuk perusahaan kecil dan menengah. Aplikasi ini berjalan pada lingkungan *web*. Fungsi yang terdapat pada aplikasi ini antara lain *procurement*, manajemen gudang, manajemen proyek dan layanan, manajemen produksi dan manajemen finansi-

Tabel 4.1: Urutan Ranking Sampel Program Pada www.sourceforge.net (sumber www.sourceforge.net, pertanggal 19 Juni 2008)

Nama	Aktivitas	Ranking	Terdaftar	Update Terakhir	Download
Openbravo ERP	100.00%	3	9 Maret 2006	18 Juni 2008	657.400
ADempiere ERP Business Suite	100.00%	8	9 September 2006	16 Juni 2008	245.313
Plazma	99.82%	397	1 Juli 2005	5 Juni 2008	18.795
jAllinOne	99.43%	1.267	14 Agustus 2006	13 Mei 2008	3.926
Freedom ERP	96.41%	7.934	1 Februari 2005	15 Mei 2007	9.539

Tabel 4.2: Versi Program Sampel

Nama Program	Versi
ADempiere ERP Business Suite	3.15
Freedom ERP	1.1.3.5
jAllinOne	0.9.15
Plazma	0.1.7
Openbravo ERP	2.35

al. Aplikasi ini didistribusikan dibawah Openbravo Public License. Aplikasi ini dapat diunduh secara bebas melalui <http://sourceforge.net/projects/openbravo/>.

Untuk memposisikan sampel program yang telah dipilih ke dalam perbandingan kualitas desain, berikut ini akan ditampilkan perbandingan dari beberapa sampel yang digunakan. Tabel 4.1 menampilkan urutan ranking sampel yang didapatkan dari www.sourceforge.net.

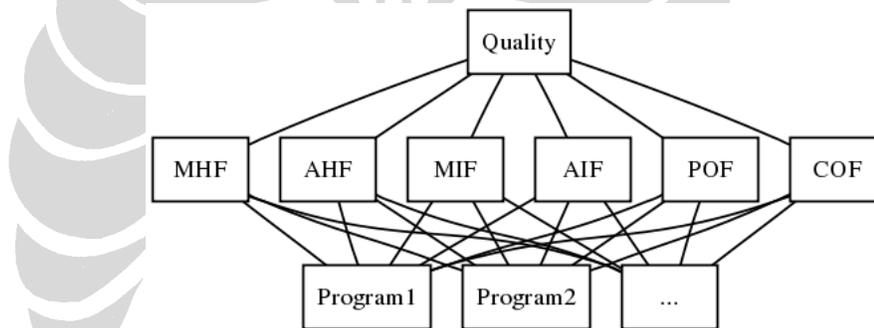
4.2 PENGUKURAN KUALITAS DESAIN

Pada subbab ini kita akan mengukur kualitas desain dari kelima sampel program yang telah dibahas pada subbab sebelumnya. Tabel 4.2 menampilkan versi program yang diukur pada proses pengukuran ini. Tahapan yang dilalui pada proses pengukuran ini adalah, menginput sampel program yang diukur, menentukan komponen penilaian dan bobot penilaian dari masing-masing komponen tersebut dan kemudian menentukan rentang nilai GOOD MOOD dari seluruh *metric* yang diikuti sertakan sebagai komponen penilaian.

Proses pengukuran kualitas desain dimulai dengan menentukan *file JAR* dari masing-masing program sampel yang berisikan *class* utama dari program sampel yang diukur. Beberapa sampel program terdiri dari lebih dari satu komponen yang membentuk program (terdiri lebih dari satu *file JAR*). Untuk sampel program ini, seluruh *file JAR*

Tabel 4.3: *File JAR* dari masing-masing sampel program

Program	<i>File JAR</i>
ADempiere ERP Business Suite	Adempiere.jar
Freedom ERP	freedomfnc.jar freedomgms.jar freedompcp.jar freedompdv.jar freedomstd.jar freedomtmk.jar
Plazma	plazma-bsolution-client-gui-0.1.6.jar plazma-bsolution-client-rich-0.1.6.jar plazma-bsolution-server-0.1.6.jar
jAllinOne	clientjaio.jar serverjaio.jar
Openbravo ERP	openbravo-core.jar



Gambar 4.1: Hirarki Komponen Penilaian

utama akan diikuti sertakan dalam pengukuran dan diperlakukan sebagai unit program yang terpisah dalam pengukuran. Tabel 4.3 menampilkan nama *file JAR* yang diikuti sertakan dalam pengukuran kualitas desain sesuai dengan paket programnya.

Untuk memenuhi persyaratan dependensi terhadap *class* yang di-*import* oleh *class* utama dari masing-masing sampel program, maka *file JAR* yang mengandung *class-class* yang dibutuhkan tersebut juga harus diikuti sertakan dalam pengukuran. Dalam *tool* pengukuran dan penentuan peringkat kualitas desain ini, *file JAR* dependensi ini disebut dengan *file ClassPath*. *File ClassPath* yang diikuti sertakan pada proses pengukuran ini dapat dilihat pada Tabel 4.4.

Setelah menginput seluruh *file JAR* dan *file ClassPath* yang akan diikuti sertakan dalam pengukuran kualitas desain, tahapan berikutnya adalah menentukan komponen penilaian ke dalam hirarki penilaian AHP. Gambar 4.1 menampilkan hirarki komponen penilaian yang digunakan dalam proses pengukuran ini.

Tabel 4.4: *File ClassPath* dari masing-masing sampel program

Program	<i>File ClassPath</i>
ADempiere ERP Business Suite	/Adempiere/lib/adempiereApps.jar /Adempiere/lib/adempiereRoot.jar /Adempiere/lib/CCTools.jar /Adempiere/lib/CInstall.jar /Adempiere/lib/CompiereJasperReqs.jar /Adempiere/lib/CSTools.jar /Adempiere/lib/customization.jar /Adempiere/lib/derby.jar /Adempiere/lib/fyracle.jar /Adempiere/lib/j2ee.jar /Adempiere/lib/jboss.jar /Adempiere/lib/oracle.jar /Adempiere/lib/patches.jar /Adempiere/lib/postgresql.jar /Adempiere/lib/sqlj.jar
Freedom ERP	/freedom/opt/freedom/lib/comm.jar
Plazma	/plazma/dist/launcher.jar /plazma/dist/plazma-bsolution-common-0.1.6.jar /plazma/dist/plazma-framework-0.1.6.jar /plazma/dist/plazma-framework-ejb-0.1.6.jar /plazma/lib/spring-fix.jar /plazma/lib/eclipse/org.eclipse.core.commands_3.2.0.jar /plazma/lib/eclipse/org.eclipse.core.runtime_3.2.0.jar /plazma/lib/eclipse/org.eclipse.equinox.common_3.2.0.jar /plazma/lib/eclipse/org.eclipse.jface_3.2.0.jar /erp/plazma/lib/eclipse/org.eclipse.osgi_3.2.0.jar /plazma/dist/plazma-rcp-0.1.6.jar /plazma/lib/swt/linux-gtk/swt.jar
jAllinOne	/jallione/lib/clientos.jar /jallione/lib/serveros.jar /jallione/lib/itext-1.3.1.jar /jallione/lib/jasperreports-1.2.7.jar /jallione/lib/jcalendar.jar /jallione/lib/jnlp.jar /jallione/lib/looks-2.0.1.jar /jallione/lib/netscape.jar /jallione/lib/OfficeLnFs_2.7.jar
Openbravo ERP	/OpenBravoERP/AppsOpenbravo/lib/servlet-api.jar /OpenBravoERP/AppsOpenbravo/lib/commons-dbc.jar

Tabel 4.6: Criteria Comparison Matrix

	MHF	AHF	MIF	AIF	POF	COF
MHF	1.000	0.125	1.000	4.000	2.000	4.000
AHF	8.000	1.000	4.000	6.000	9.000	9.000
MIF	1.000	0.250	1.000	5.000	6.000	7.000
AIF	0.250	0.167	0.200	1.000	0.333	1.000
POF	0.500	0.111	0.167	3.000	1.000	2.000
COF	0.250	0.111	0.143	1.000	0.500	1.000

cat: MHF — Method Hiding Factor, AHF — Attribute Hiding Factor, MIF — Method Inheritance Factor, AIF — Attribute Inheritance Factor, POF — Polymorphism Factor, COF — Coupling Factor

Tabel 4.7: Criteria Priority Vector

Komponen	Bobot
Attribute Hiding Factor	0.537
Method Inheritance Factor	0.199
Method Hiding Factor	0.127
Polymorphism Factor	0.063
Attribute Inheritance Factor	0.040
Coupling Factor	0.035

Komponen penilaian pada proses pengukuran ini terdiri dari seluruh *metric* dari MOOD yang diletakkan pada tingkat hirarki yang sama. Dengan menggunakan konfigurasi hirarki penilaian seperti ini, masing-masing komponen penilaian dapat ditentukan bobot penilaiannya dibandingkan dengan komponen penilaian lainnya.

Proses selanjutnya adalah menentukan bobot penilaian untuk masing-masing komponen penilaian. Perbandingan bobot penilaian untuk masing-masing komponen penilaian dapat dilihat pada Tabel 4.6 yang ditampilkan sebagai matriks perbandingan antar masing-masing komponen penilaian. Urutan bobot penilaian untuk masing-masing komponen selengkapnya dapat dilihat pada Tabel 4.7. Untuk rentang nilai GOOD MOOD dari masing-masing *metric* di-set dengan rentang nilai yang terdapat pada Tabel 3.1.

Pada subbab berikutnya, akan dibahas mengenai hasil pengukuran yang didapatkan dari proses pengukuran yang dikonfigurasi dengan komponen penilaian yang telah dibahas.

Tabel 4.8: Hasil MOOD Scores

Program	MHF	AHF	MIF	AIF	POF	COF
/freedom/opt/freedom/lib/freedomstd.jar	0.2466	0.9166	0.9795	0.8731	0.0086	0.0061
/freedom/opt/freedom/lib/freedomfnc.jar	0.2174	0.8778	0.9755	0.8642	0.0097	0.0065
/freedom/opt/freedom/lib/freedomgms.jar	0.2605	0.9215	0.9787	0.8662	0.0089	0.0062
/freedom/opt/freedom/lib/freedompcp.jar	0.1773	0.8410	0.9675	0.8636	0.0139	0.0087
/freedom/opt/freedom/lib/freedompdv.jar	0.2304	0.9045	0.9770	0.8680	0.0090	0.0074
/freedom/opt/freedom/lib/freedomtmk.jar	0.2548	0.9271	0.9804	0.8722	0.0084	0.0072
/plazma/dist/plazma-bsolution-client-gui-0.1.6.jar	0.6146	0.9950	0.9854	0.9654	0.0078	0.0005
/plazma/dist/plazma-bsolution-client-rich-0.1.6.jar	0.5630	0.9843	0.9865	0.9502	0.0097	0.0007
/plazma/dist/plazma-bsolution-server-0.1.6.jar	0.4829	0.7774	0.9626	0.7800	0.0253	0.0008
/jallione/lib/clientjaio.jar	0.1123	0.9697	0.9357	0.7425	0.0185	0.0012
/jallione/lib/serverjaio.jar	0.0347	0.9046	0.6775	0.0050	0.0616	0.0006
/OpenBravoERP/AppsOpenbravo/lib/openbravo-core.jar	0.1479	0.8440	0.6717	0.4606	0.1770	0.0212
/Adempiere/lib/Adempiere.jar	0.1463	0.3708	0.8841	0.7208	0.0174	0.0010

4.3 HASIL PENGUKURAN KUALITAS DESAIN

Hasil pengukuran kualitas desain terhadap sampel program dapat dilihat pada Tabel 4.8 dan Tabel 4.9. Tabel 4.8 dan Tabel 4.9 menampilkan nilai MOOD dan nilai GOOD MOOD dari masing-masing sampel program yang diukur. Nilai akhir untuk kualitas desain menurut komponen penilaian AHP ditampilkan pada Tabel 4.10.

Sampel program ditampilkan pada Tabel 4.10 diurutkan dari program yang memiliki nilai tertinggi sampai yang terendah. Semakin tinggi nilai AHP dari sampel program, semakin baik pula kualitas desain dari program tersebut yang didasarkan pada masing-masing bobot penilaian pada AHP.

Dari hasil pengukuran yang didapatkan, terlihat bahwa Adempiere menduduki peringkat yang paling akhir berdasarkan kualitas desain. Openbravo ERP juga menduduki posisi bawah. Hal ini cukup mengejutkan, kedua aplikasi tersebut merupakan aplikasi ERP *open source* yang cukup populer dan banyak di-*download* (Tabel 4.1).

Bagaimana dengan hasil pengukuran apabila sampel program di atas diurutkan dengan mengikuti metode yang digunakan Nurmaya? Tabel 4.11 menampilkan rata-rata, variance dan standar deviasi dari hasil pengukuran sampel program. Tabel 4.12 menampilkan rentang *Confidence Interval Good OOD* yang digunakan dalam proses penentuan urutan program berdasarkan kualitas program.

Tabel 4.13 menampilkan status penilaian program berdasarkan CIGOOD. Program yang nilai *metric*-nya tidak berada pada rentang CIGOOD (Tabel 4.12) diberikan nilai

Tabel 4.9: Hasil GOOD MOOD Scores

Program	MHF	AHF	MIF	AIF	POF	COF
/freedom/opt/freedom/lib/freedomstd.jar	0.623	0.958	0.990	0.937	0.504	0.503
/freedom/opt/freedom/lib/freedomfnc.jar	0.609	0.939	0.988	0.932	0.505	0.503
/freedom/opt/freedom/lib/freedomgms.jar	0.630	0.961	0.989	0.933	0.504	0.503
/freedom/opt/freedom/lib/freedompcp.jar	0.589	0.921	0.984	0.932	0.507	0.504
/freedom/opt/freedom/lib/freedompdv.jar	0.615	0.952	0.989	0.934	0.504	0.504
/freedom/opt/freedom/lib/freedomtmk.jar	0.627	0.964	0.990	0.936	0.504	0.504
/plazma/dist/plazma-bsolution-client-gui-0.1.6.jar	0.807	0.997	0.993	0.983	0.504	0.500
/plazma/dist/plazma-bsolution-client-rich-0.1.6.jar	0.782	0.992	0.993	0.975	0.505	0.500
/plazma/dist/plazma-bsolution-server-0.1.6.jar	0.741	0.889	0.981	0.890	0.513	0.500
/jallione/lib/clientjaio.jar	0.556	0.985	0.968	0.871	0.509	0.501
/jallione/lib/serverjaio.jar	0.517	0.952	0.839	0.503	0.531	0.500
/OpenBravoERP/AppsOpenbravo/lib/openbravo-core.jar	0.574	0.922	0.836	0.730	0.589	0.511
/Adempiere/lib/Adempiere.jar	0.573	0.685	0.942	0.860	0.509	0.500

Tabel 4.10: AHP Weighted Scores

Program	Score
/plazma/dist/plazma-bsolution-client-gui-0.1.6.jar	0.083
/plazma-bsolution-client-rich-0.1.6.jar	0.083
/freedom/opt/freedom/lib/freedomtmk.jar	0.079
/freedom/opt/freedom/lib/freedomgms.jar	0.079
/freedom/opt/freedom/lib/freedomstd.jar	0.078
/jallione/lib/clientjaio.jar	0.078
/freedom/opt/freedom/lib/freedompdv.jar	0.078
/freedom/opt/freedom/lib/freedomfnc.jar	0.077
/plazma/dist/plazma-bsolution-server-0.1.6.jar	0.077
/freedom/opt/freedom/lib/freedompcp.jar	0.076
/OpenBravoERP/AppsOpenbravo/lib/openbravo-core.jar	0.074
/jallione/lib/serverjaio.jar	0.073
/Adempiere/lib/Adempiere.jar	0.065

Tabel 4.11: Rata-rata, Variance dan Standar Deviasi Sampel Program

	Rata-rata	Variance	Standar Deviasi
MHF	0.2684	0.0312	0.1766
AHF	0.8642	0.0257	0.1602
MIF	0.9202	0.0126	0.1125
AIF	0.7563	0.0675	0.2598
POF	0.0289	0.0022	0.0468
COF	0.0052	0.0000	0.0058

Tabel 4.12: Confidence Interval Good OOD (CIGOOD)

	Lower Limit	Upper Limit
MHF	0.2622	0.2745
AHF	0.8586	0.8698
MIF	0.9162	0.9241
AIF	0.7472	0.7654
POF	0.0273	0.0305
COF	0.0050	0.0055

Tabel 4.13: Status Program berdasarkan hasil kalkulasi CIGOOD

	MHF	AHF	MIF	AIF	POF	COF
/home/david/project/erp/freedom/opt/freedom/lib/freedomstd.jar	0	0	0	0	0	0
/home/david/project/erp/freedom/opt/freedom/lib/freedomfnc.jar	0	0	0	0	0	0
/home/david/project/erp/freedom/opt/freedom/lib/freedomgms.jar	0	0	0	0	0	0
/home/david/project/erp/freedom/opt/freedom/lib/freedompcp.jar	0	0	0	0	0	0
/home/david/project/erp/freedom/opt/freedom/lib/freedompdv.jar	0	0	0	0	0	0
/home/david/project/erp/freedom/opt/freedom/lib/freedomtmk.jar	0	0	0	0	0	0
/home/david/project/erp/plazma/dist/plazma-bsolution-client-gui-0.1.6.jar	0	0	0	0	0	0
/home/david/project/erp/plazma/dist/plazma-bsolution-client-rich-0.1.6.jar	0	0	0	0	0	0
/home/david/project/erp/plazma/dist/plazma-bsolution-server-0.1.6.jar	0	0	0	0	0	0
/home/david/project/erp/jallione/lib/clientjaio.jar	0	0	0	0	0	0
/home/david/project/erp/jallione/lib/serverjaio.jar	0	0	0	0	0	0
/media/sda9/david/OpenBravoERP/AppsOpenbravo/lib/openbravo-core.jar	0	0	0	0	0	0
/home/david/Adempiere/lib/Adempiere.jar	0	0	0	0	0	0

0, dan program yang nilai *metric*-nya berada pada rentang CIGOOD diberikan nilai 1. Dari hasil penilaian tersebut, tidak ada satupun sampel program memiliki nilai *metric* di dalam rentang CIGOOD, dengan kata lain seluruh sampel program tersebut dikelompokkan sebagai kelompok tidak normal.

Hal ini disebabkan oleh karena terlalu kecilnya batasan nilai dari rentang CIGOOD (Tabel 4.12) yang dihasilkan dengan menggunakan Confidence Interval dengan nilai Confidence Level sebesar 90%. Untuk memperbesar rentang nilai CIGOOD, dapat dilakukan dengan menurunkan Confidence Level dalam menentukan CIGOOD.

Program yang dikelompokkan pada kelompok tidak normal dihitung menggunakan derajat *error*. Tabel 4.14 menampilkan hasil perhitungan derajat *error* untuk sampel program yang diukur.

Peringkat sampel program berdasarkan kualitas desain dari kualitas desain yang paling tinggi ke yang paling rendah adalah sebagai berikut:

Tabel 4.14: Derajat Error dari masing-masing sampel program

	MHF	AHF	MIF	AIF	POF	COF	Total
/home/david/project/erp/freedom/opt/freedom/lib/freedomstd.jar	0.0156	0.0468	0.0555	0.1078	0.0187	0.0007	0.24
/home/david/project/erp/freedom/opt/freedom/lib/freedomfnc.jar	0.0448	0.0080	0.0514	0.0989	0.0176	0.0011	0.22
/home/david/project/erp/freedom/opt/freedom/lib/freedomgms.jar	0.0017	0.0517	0.0546	0.1009	0.0184	0.0008	0.23
/home/david/project/erp/freedom/opt/freedom/lib/freedompcp.jar	0.0849	0.0176	0.0434	0.0982	0.0134	0.0032	0.26
/home/david/project/erp/freedom/opt/freedom/lib/freedompdv.jar	0.0318	0.0347	0.0529	0.1026	0.0183	0.0020	0.24
/home/david/project/erp/freedom/opt/freedom/lib/freedomtmk.jar	0.0074	0.0574	0.0563	0.1069	0.0189	0.0018	0.25
/home/david/project/erp/plazma/dist/plazma-bsolution-client-gui-0.1.6.jar	0.3400	0.1252	0.0613	0.2001	0.0194	0.0046	0.75
/home/david/project/erp/plazma/dist/plazma-bsolution-client-rich-0.1.6.jar	0.2885	0.1145	0.0625	0.1848	0.0176	0.0043	0.67
/home/david/project/erp/plazma/dist/plazma-bsolution-server-0.1.6.jar	0.2084	0.0812	0.0385	0.0146	0.0019	0.0042	0.35
/home/david/project/erp/jallione/lib/clientjaio.jar	0.1499	0.1000	0.0116	0.0047	0.0088	0.0039	0.28
/home/david/project/erp/jallione/lib/serverjaio.jar	0.2275	0.0348	0.2387	0.7422	0.0311	0.0044	1.28
/media/sda9/david/OpenBravoERP/AppsOpenbravo/lib/openbravo-core.jar	0.1143	0.0146	0.2446	0.2867	0.1465	0.0158	0.82
/home/david/Adempiere/lib/Adempiere.jar	0.1159	0.4878	0.0321	0.0264	0.0099	0.0041	0.68

1. /freedom/opt/freedom/lib/freedomfnc.jar
2. /freedom/opt/freedom/lib/freedomgms.jar
3. /freedom/opt/freedom/lib/freedomstd.jar
4. /freedom/opt/freedom/lib/freedompdv.jar
5. /freedom/opt/freedom/lib/freedomtmk.jar
6. /freedom/opt/freedom/lib/freedompcp.jar
7. /jallione/lib/clientjaio.jar
8. /plazma/dist/plazma-bsolution-server-0.1.6.jar
9. /plazma/dist/plazma-bsolution-client-rich-0.1.6.jar
10. /Adempiere/lib/Adempiere.jar
11. /plazma/dist/plazma-bsolution-client-gui-0.1.6.jar
12. /OpenBravoERP/AppsOpenbravo/lib/openbravo-core.jar
13. /jallione/lib/serverjaio.jar

Dari kedua proses penentuan peringkat kualitas desain, hasil yang didapatkan tidak memiliki perubahan yang signifikan. Adempiere dan Openbravo ERP masih tetap berada posisi bawah, walaupun pada penentuan peringkat dengan Confidence Interval, Openbravo ERP menempati posisi yang lebih rendah dibanding Adempiere. Posisi teratas yang pada penentuan peringkat dengan AHP diduduki oleh Plazma digantikan oleh Freedom ERP.

Penggunaan AHP pada *tool* yang dikembangkan ini merupakan penyempurnaan dari metode yang digunakan oleh Abreu [ABR96] dan Nurmaya [NUR07] dalam menentukan peringkat program berdasarkan kualitas desainnya. AHP menghilangkan ketidak-konsistensi-an penilaian dengan menggunakan *Confidence Interval*. Perbedaan yang terlihat dari kedua teknik pemeringkatan ini adalah pada penentuan rentang nilai *metric* yang dianggap memiliki kualitas desain yang baik.

Subjek Pengukuran Yang Terdiri Dari Beberapa Modul

Pada uji coba pengukuran yang telah dibahas sebelumnya, beberapa sampel program yang digunakan terdiri dari beberapa modul program yang berbeda. Hal ini menyebabkan paket aplikasi yang diukur mendapatkan nilai yang berbeda-beda tergantung dari berapa banyak komponen yang diikuti sertakan dalam proses pengukuran. Untuk mendapatkan satu nilai untuk satu paket aplikasi yang diukur, dapat dilakukan dengan menginput *directory* yang mengandung seluruh komponen aplikasi tersebut sebagai satu objek aplikasi yang diukur.

Kenapa tidak memasukkan seluruh *file JAR* yang didistribusikan bersama dengan paket program yang menjadi sampel? Masalah yang timbul apabila seluruh *file JAR* dari program sampel dimasukkan sebagai input ke dalam *tool* pengukuran ini, seluruh *file JAR* yang diproses akan membutuhkan memori yang cukup besar. Hal ini terjadi karena pada beberapa *metric* seperti POF dan COF men-*scan* seluruh *class* terdapat pada sistem yang diukur. Dari percobaan pengukuran yang penulis lakukan dengan cara ini, selalu gagal dengan pesan “OutOfMemoryError”, yang menandakan memori yang tersedia sudah tidak mencukupi. Kegagalan ini terutama terjadi untuk sampel program yang memiliki kelas yang cukup banyak. Percobaan pengukuran dilakukan pada komputer dengan jumlah memori fisik sebesar 748Mb.

Untuk mengatasi masalah keterbatasan memori, penulis mencoba untuk mengubah strategi dalam memproses *class* dari paket aplikasi yang diukur. *Class* yang diinput, hanya didaftarkan nama *file*-nya, sedangkan proses *parsing* terhadap *class* tersebut (un-

Tabel 4.15: Rata-rata nilai *metric* jAllione

	/jallione/lib/clientjiao.jar	/jallione/lib/serverjiao.jar	Rata-rata jAllione
MHF	0.556	0.517	0.537
AHF	0.985	0.952	0.969
MIF	0.968	0.839	0.904
AIF	0.871	0.503	0.687
POF	0.509	0.531	0.520
COF	0.501	0.500	0.500

tuk membaca informasi mengenai *class* yang diukur, seperti jumlah method, hirarki *inheritance* dan sebagainya) dilakukan pada saat dibutuhkan saja. Untuk menghemat memori, objek *class* yang telah di-*parsing* (di dalam program disimpan sebagai objek `JavaClass`) akan dibuang setelah proses pengukuran selesai. Kelemahan terbesar dari strategi ini adalah, ketidak-efisien-an dalam proses pengukuran. Hal ini disebabkan karena proses *parsing* yang dilakukan berulang-ulang terhadap objek *class* yang sama. Oleh karena itu, waktu proses pengukuran akan menjadi lama terutama untuk paket aplikasi yang terdiri dari banyak *class*.

Jalan tengah untuk mendapatkan satu nilai untuk satu paket aplikasi yang diukur adalah dengan mendaftarkan *file* JAR yang merupakan komponen utama dari aplikasi yang diukur. Masing-masing JAR ini akan diukur secara mandiri, sehingga hasilnya adalah nilai *metric* yang hanya mengukur sebagian paket aplikasi, bukan keseluruhan. Dengan menghitung rata-rata dari nilai *metric* dari *file* JAR yang terpisah tadi, akan didapatkan nilai *metric* untuk paket aplikasi yang dibentuk dari keseluruhan *file* JAR.

Contoh berikut memperlihatkan proses penyatuan nilai *metric* dari paket aplikasi jAllione yang terdiri dari dua komponen. Kita akan mempergunakan hasil pengukuran yang telah didapatkan dari uji coba pengukuran yang telah dilakukan. Komponen jAllione yang diikuti sertakan dalam uji coba pengukuran adalah `clientjiao.jar` dan `serverjiao.jar`.

1. Hitung rata-rata *metric* dari seluruh komponen yang membentuk aplikasi yang diukur. Tabel 4.15 menampilkan rata-rata nilai *metric* jAllione dari hasil percobaan pengukuran yang dilakukan sebelumnya.
2. Masukkan nilai rata-rata *metric* ke dalam AHP dan kemudian hitung kembali nilai AHP