

# **Laporan Event Organizer Program**



**Disusun oleh: Kelompok 24**

<b>Regina Arum S.R.</b>	<b>I0322104</b>
<b>Santiago Aghna Junior</b>	<b>I0322116</b>
<b>Tsamara Najwa Dzakiyya</b>	<b>I0322125</b>
<b>Vincent Arif Kurniawan</b>	<b>I0322127</b>



## BAB I

## DESKRIPSI MASALAH

**1.1 Latar Belakang**

Event Organizer (EO) adalah penyedia jasa profesional dalam menyelenggarakan acara. Misalnya acara pernikahan, ulang tahun, lamaran, ulang tahun pernikahan ataupun gathering. Karena kompetensi ini tidak dimiliki oleh orang lain maka penyelenggara suatu event dapat mempercayakan kepada organisasi yang berkompeten pada bidang tersebut. Semakin berkembangnya acara yang diselenggarakan, baik dari perusahaan ataupun pribadi, diperlukan panitia untuk menyusun dan mengelola acara. Kesulitan dalam menentukan konsep acara serta keterbatasan kemampuan dan SDM menjadi suatu kesulitan *customer* tersendiri.

Raven Media Event Organizer adalah salah satu Event Organizer (EO) yang ada di kota Solo. Sebagai salah satu EO yang berkembang di kota besar perlu adanya inovasi yang memiliki segmentasi yang berbeda dengan EO yang lainnya. Maka sebagai penyedia jasa, Raven Media Event Organizer membuat program EO yang mempermudah pilihan acara dan fasilitas yang dibutuhkan oleh *customer*. Program yang dibuat akan memudahkan *customer* dalam menentukan pilihan tanggal acara, tema acara, pilihan tempat dengan berbagai kapasitas, pilihan undangan hingga kemudahan dalam sistematika pembayaran. Selain itu Raven team juga menambahkan kode verifikasi pembayaran. Oleh karena itu Raven Media Event Organizer menjadi jawaban *customer* ketika akan menyelenggarakan suatu acara.

**1.2 Rumusan Masalah**

Rumusan masalah yang terbentuk adalah sebagai berikut.

1. Apakah pemilihan fasilitas acara dalam pengadaan event dapat dipermudah?
2. Apakah proses pencarian, pilihan layanan, pemesanan, dan pembayaran dapat disederhanakan?
3. Apakah pengadaan event dapat dilakukan di manapun dan kapanpun?
4. Bagaimana penyusunan konsep acara dengan rapi?
5. Bagaimana membantu *customer* mengendalikan biaya dalam pengadaan event?

**1.3 Tujuan**

Tujuan yang akan dicapai adalah sebagai berikut.

1. Memudahkan pemilihan fasilitas acara dalam pengadaan event.
2. Menyederhanakan proses pencarian, pilihan layanan, pemesanan, dan pembayaran.
3. Memudahkan pengadaan event kapanpun dan di manapun untuk efisiensi waktu.
4. Membuat konsep acara yang tersusun dengan rapi dan jelas.
5. Membantu *customer* mengendalikan biaya dalam pengadaan event.

## BAB II

### FLOWCHART

Bab ini menjelaskan mengenai Flowchart Raven Media Event Organizer.

#### 2.1 *Booking data*

Subbab ini menjelaskan mengenai alur flowchart yang paling awal dari program Event Organizer yaitu proses *booking data*. Alur pertama dari program yang dibuat yaitu klien baru diminta untuk mengisi biodata terlebih dahulu. Biodata tersebut terdiri dari pengisian nama dari klien baru yang akan menyelenggarakan acara lalu nomor yang dapat dihubungi oleh pihak EO. Setelah klien baru mengisi biodata, klien baru akan memilih tanggal dan tempat terlaksananya acara yang direncanakan. Program akan meminta klien baru untuk memasukkan tanggal, bulan, tahun serta tempat dari acara yang akan diadakan. Setelah semua data terisi, program akan mengolah data tersebut dan memberikan konfirmasi apakah di tanggal dan tempat tersebut sudah di *booking* atau belum, jika sudah ada klien yang sebelumnya mem-*booking* di tanggal dan tempat tersebut maka klien baru diminta untuk memasukkan ulang data tanggal dan tempat yang berbeda, namun jika di tanggal dan tempat yang dimasukkan klien baru tidak ada yang mem-*booking* maka akan lanjut ke program selanjutnya.

#### 2.2 *Opsi*

Subbab ini menjelaskan mengenai beberapa opsi yang ditawarkan Raven Media Event Organizer mulai dari opsi jenis acara, opsi jenis undangan, opsi *live* musik, opsi jenis hidangan tamu, opsi pembawa acara, opsi dokumentasi, hingga opsi percetakan.

##### 2.2.1 Opsi Jenis Acara

Pada opsi jenis acara program akan menampilkan pilihan acara yang akan diadakan. Pada program kami akan menampilkan 5 jenis acara diantaranya Lamaran, Resepsi Pernikahan, Ulang Tahun, Ulang Tahun Pernikahan, serta *Gathering*. Klien akan diminta untuk memilih satu dari lima pilihan jenis acara beserta judul dari acara tersebut, jika klien memilih lebih dari satu maka program akan tidak berjalan. Setelah memasukkan pilihan acara klien akan memberikan judul dari acara yang dipilih beserta jumlah tamu undangan yang akan datang sesuai dengan kapasitas gedung yang telah dipilih sebelumnya.

##### 2.2.2 Opsi Jenis Undangan

Pada opsi jenis undangan program akan menampilkan pilihan pencetakan dari undangan acara yang disebar apakah dalam penyebaran undangan perlu dicetak atau hanya disebar



melalui media elektronik. Pilihan yang ditampilkan program diantaranya *hard cover*, *soft cover*, dan digital.

#### 2.2.3 Opsi *Live* Musik

Pada opsi *live* musik yang disediakan oleh Raven Media Event Organizer adalah *Simple*, *Simple Trio*, *Band*, *Exclusive Band*, *Light Orchestra*, *Big Band*, dan tanpa *live* musik. Klien bisa memilih pilihan *live* musik yang disediakan jika ingin menampilkan *live* musik pada acaranya atau klien juga dapat memilih tanpa *live* musik jika tidak ingin menampilkan *live* musik di acaranya.

#### 2.2.4 Opsi Jenis Hidangan tamu

Pada opsi program akan menampilkan pilihan hidangan yang akan diberikan pada tamu undangan yaitu prasmanan atau *catering*. Klien dapat memilih salah satu diantara keduanya.

#### 2.2.5 Opsi Pembawa Acara

Pada opsi ini klien akan diberikan pilihan pembawa acara yaitu satu orang atau dua orang yang akan membawakan pada acara dari klien nanti. Klien dapat memilih salah satu diantara keduanya.

#### 2.2.6 Opsi Dokumentasi

Pada opsi ini klien akan diberikan pilihan dokumentasi acara yaitu fotografi atau fotografi dan videografi. Klien dapat memilih salah satu diantara keduanya.

#### 2.2.7 Opsi Percetakan

Pada opsi percetakan ini program akan menampilkan pilihan percetakan foto sebagai tindak lanjut dari opsi dokumentasi sebelumnya. Pilihan yang ditampilkan diantaranya Paket Kenangan, Paket Kenangan Plus, Paket Pajangan, Paket *Big Screen*, Paket Komplit, dan Tanpa Percetakan. Klien dapat memilih pilihan paket percetakan yang disediakan namun klien juga dapat memilih pilihan tanpa percetakan jika tindak menginginkan percetakan.

### 2.3 *Preview* Biaya

Subbab ini menjelaskan mengenai bagian *preview* biaya. Bagian ini merupakan *output* dari program yang akan menampilkan seluruh pilihan yang telah dipilih dari seluruh opsi sebelumnya beserta harga yang akan ditampilkan sesuai input pilihan klien.

### 2.4 Pembayaran

Subbab ini menjelaskan mengenai skema pembayaran, yang memberikan pilihan kepada klien untuk mentransfer biaya event dengan opsi 3 bank, yaitu bank Mandiri, BCA, dan BNI. Setelah klien memilih bank tujuan untuk mentransfer maka akan muncul kode verifikasi pembayaran, apabila kode pembayaran valid maka pesanan klien pun berhasil. Jika kode



# Event Organizer Program

verifikasi yang dimasukkan klien salah maka klien memiliki kesempatan untuk memasukkan kode sebanyak 3 kali. Namun jika kode yang dimasukkan gagal lebih dari 3 kali maka otomatis pesanan akan hangus.

## BAB III

### KODE PROGRAM

#### 3.1 *From Date*

```
program_utama.py > ...
1  #program utama untuk di-run di terminal
2  from datetime import date
3  from dateutil.relativedelta import relativedelta
4
5  TanggalBase = date(1800,1,1)
6  hari = ["Rabu", "Kamis", "Jumat", "Sabtu", "Minggu", "Senin", "Selasa"]
7  data_klien = []
8  booking_data = []
9
```

Program mengimpor dua modul dari pustaka datetime: "date" dan "relativedelta". Modul "date" digunakan untuk memanipulasi dan melakukan operasi terkait tanggal, sedangkan modul "relativedelta" digunakan untuk menambahkan batas tanggal pesanan klien. Program memberikan batasan bagi klien untuk mengadakan acara secepat-cepatnya satu bulan dari tanggal pemesanan. Pendefinisian validitas tanggal dilakukan dengan mengecek apakah tanggal pemesanan lebih jauh dari tanggal batasan. Program mendefinisikan variabel "TanggalBase" yang berisi objek date yang mewakili tanggal 1 Januari 1800. Variabel ini digunakan sebagai titik referensi untuk melakukan perhitungan tanggal.

Program mendefinisikan sebuah list bernama "hari" yang berisi nama-nama hari dalam bahasa Indonesia, mulai dari hari Rabu hingga Selasa. Indeks 0 pada list ini merepresentasikan hari Rabu, karena tanggal basis yang digunakan (1 Januari 1800) adalah hari Rabu. Program mendefinisikan dua variabel, yaitu "data\_klien" dan "booking\_data" sebagai list kosong. Variabel ini nantinya akan digunakan untuk menyimpan data klien dan data pemesanan.

### 3.2 Def nama\_input()

```

9
10 #biodata
11 print('*~*50)
12 print('EVENT ORGANIZER PROGRAM \nJust For You!')
13 print('Kelola event yang ingin Anda adakan dengan \ncepat dan mudah, tanpa ribet!')
14 print('Courtesy of Kelompok 24 Kelas D IMPERIALE')
15 print('*~*50)
16 data_klien = []
17 def nama_input():
18     try:
19         nama = str(input('Masukkan nama Anda = '))
20         kondisi1 = all(x.isspace() for x in nama)
21         kondisi2 = all(x.isalpha() for x in nama)
22         kondisi3 = all(x.isalpha() or x.isspace() for x in nama)
23         if kondisi1 == True:
24             print('Maaf, nama tidak dapat berupa karakter kosong.')
25             nama_input()
26         elif kondisi2 == True:
27             data_klien.append(nama)
28         elif kondisi3 == True:
29             data_klien.append(nama)
30         else:
31             print('Maaf, nama harus berupa karakter huruf. Coba lagi')
32             nama_input()
33     except KeyboardInterrupt:
34         print('\nMohon tidak menghentikan program secara paksa')
35         nama_input()
36 nama_input()
37 def no_telp_input():
38     try:
39         no_telp = input('Masukkan nomor telepon Anda = +62')
40         cek_no_telp = no_telp.isnumeric()
41         if cek_no_telp == False:
42             print('Maaf, nomor telepon harus berupa karakter angka. Coba lagi')
43             no_telp_input()
44         else: data_klien.append(f'+62{no_telp}')
45     except KeyboardInterrupt:
46         print('\nMohon tidak menghentikan program secara paksa')
47         no_telp_input()
48 no_telp_input()
49 print('*~*50)
50

```

Program tersebut adalah sebuah program sederhana yang mengumpulkan biodata dari klien untuk keperluan event organizer. Berikut adalah penjelasan mengenai program tersebut:

Program mendefinisikan variabel "data\_klien" sebagai list kosong. Variabel ini akan digunakan untuk menyimpan data klien. Program mendefinisikan sebuah fungsi bernama "nama\_input()". Fungsi ini bertujuan untuk meminta input nama dari pengguna dan melakukan validasi terhadap input yang diberikan.

Di dalam fungsi "nama\_input()", terdapat beberapa kondisi validasi yang dilakukan terhadap input nama pengguna:

Kondisi 1: Jika nama hanya terdiri dari spasi, maka program akan mencetak pesan kesalahan dan memanggil fungsi "nama\_input()" lagi.

Kondisi 2: Jika semua karakter pada nama adalah alfabet, maka nama akan ditambahkan ke dalam variabel "data\_klien".

Kondisi 3: Jika semua karakter pada nama adalah alfabet atau spasi, maka nama akan ditambahkan ke dalam variabel "data\_klien".

Di dalam fungsi "nama\_input()", terdapat juga penanganan exception menggunakan blok "try-except" untuk menangkap exception KeyboardInterrupt. Jika pengguna mencoba menghentikan program dengan menekan Ctrl+C, program akan mencetak pesan dan



memanggil fungsi "nama\_input()" lagi. Setelah itu, program memanggil fungsi "nama\_input()" untuk meminta input nama dari pengguna.

Di dalam fungsi "no\_telp\_input()", terdapat satu kondisi validasi:

Jika nomor telepon tidak berupa karakter angka, maka program akan mencetak pesan kesalahan dan memanggil fungsi "no\_telp\_input()" lagi. Di dalam fungsi "no\_telp\_input()", juga terdapat penanganan exception menggunakan blok "try-except" untuk menangkap exception KeyboardInterrupt. Setelah itu, program memanggil fungsi "no\_telp\_input()" untuk meminta input nomor telepon dari pengguna.

### 3.3 Def tanggal()

```

50 |
51 | #tanggal acara
52 | def tanggal():
53 |     try:
54 |         while True:
55 |             print('Notes: Anda hanya dapat mengadakan acara secepat-cepatnya satu bulan dari sekarang.')
56 |             print('Kapan Anda akan mengadakan acara? (DD/MM/YYYY)')
57 |             dd = int(input('Tanggal - Masukkan angka tanggal (DD) = '))
58 |             mm = int(input('Bulan - Masukkan angka bulan (MM) = '))
59 |             yy = int(input('Tahun - Masukkan angka tahun (YYYY) = '))
60 |             tanggal_acara = date(int(yy), int(mm), int(dd))
61 |             tanggal_limit = date.today() + relativedelta(months=+1)
62 |             if tanggal_acara < tanggal_limit:
63 |                 print('Coba lagi.')
64 |             else:
65 |                 print('Tanggal valid.')
66 |                 break
67 |             selisih = tanggal_acara - TanggalBase
68 |             index_hari = selisih.days % 7
69 |             hari_acara = hari[index_hari]
70 |             booking_data.append(hari_acara)
71 |             booking_data.append(str(tanggal_acara))
72 |         except KeyboardInterrupt:
73 |             print('\nMohon tidak menghentikan program secara paksa')
74 |             tanggal()
75 |         except ValueError:
76 |             print('\nMohon memberikan input dengan benar')
77 |             tanggal()
78 |     tanggal()
79 |

```

Program tersebut adalah sebuah fungsi untuk memvalidasi dan mengumpulkan tanggal acara dari pengguna. Fungsi tersebut menggunakan loop while untuk terus meminta input tanggal hingga tanggal yang valid diberikan. Setelah mendapatkan tanggal yang valid, fungsi menghitung hari acara berdasarkan tanggal tersebut dan menyimpannya dalam list "booking\_data". Fungsi juga menangani pengecualian jika pengguna menghentikan program secara paksa atau memberikan input yang tidak valid.

### 3.4 Def pilih\_tempat()

```

79
80 #tempat
81 import lokasi
82 print('A. The Royal Surakarta Heritage')
83 print('B. Pose In Hotel')
84 print('C. Graha Saba Buana')
85 print('D. Ramada Suites (outdoor)')
86 print('E. Adhiwangsa Hotel & Convention (outdoor)')
87 def pilih_tempat():
88     lokasi.pilihan()
89     lokasi.verif()
90     booking_data.append(lokasi.tempat_dipilih[-1])
91 pilih_tempat()
92 print('='*50)
93

```

Program tersebut berfungsi untuk memilih dan memverifikasi tempat acara dari opsi yang disediakan. Setelah pengguna memilih tempat acara, program memverifikasinya menggunakan modul "lokasi" dan menyimpan pilihan tempat dalam list "booking\_data".

### 3.5 Def cek(x)

```

#cek tanggal dan tempat
def cek(x):
    baca_datatempat = open('booking.txt','r')
    data_ada = baca_datatempat.read()
    if str(x) in data_ada:
        print('Mohon maaf, tempat dan tanggal tersebut sudah dipesan oleh klien lain. \nMohon coba mengganti tanggal atau tempat')
        ubah = input('A. Tanggal\nB. Tempat\nApa yang ingin Anda ubah? ')
        if ubah.upper() == 'A':
            booking_data.clear()
            tanggal()
            booking_data.append(lokasi.tempat_dipilih[-1])
            cek(booking_data)
        if ubah.upper() != 'A':
            if ubah.upper() == 'B':
                print('A. The Royal Surakarta Heritage')
                print('B. Pose In Hotel')
                print('C. Graha Saba Buana')
                print('D. Ramada Suites (outdoor)')
                print('E. Adhiwangsa Hotel & Convention (outdoor)')
                booking_data.pop()
                pilih_tempat()
                cek(booking_data)
            else: 'Input tidak valid'
    else:
        datatempat = open('booking.txt', 'a')
        datatempat.write(f'\n{str(x)}')
        print('Selamat, tempat yang Anda pesan tersedia! Silakan melanjutkan pesanan Anda.')
        cek(booking_data)

```

Program diatas berfungsi untuk memeriksa ketersediaan tanggal dan tempat acara dalam file "booking.txt". Jika tempat dan tanggal sudah dipesan oleh klien lain, program memberikan opsi kepada pengguna untuk mengubah tanggal atau tempat acara. Jika tempat dan tanggal tersedia, program menyimpan informasi pemesanan ke dalam file "booking.txt" dan memungkinkan pengguna melanjutkan pesanan.

### 3.6 Def nama()

```

def namapria():
    nama_pria = input('Nama Pria: ')
    kondisi1 = all(x.isspace() for x in nama_pria)
    kondisi2 = all(x.isalpha() for x in nama_pria)
    kondisi3 = all(x.isalpha() or x.isspace() for x in nama_pria)
    if kondisi1 == True:
        print('Maaf, nama tidak dapat berupa karakter kosong.')
        namapria()
    elif kondisi2 == True:
        namafixpria.append(nama_pria)
    elif kondisi3 == True:
        namafixpria.append(nama_pria)
    else:
        print('Maaf, nama harus berupa karakter huruf. Coba lagi')
        namapria()
    return namafixpria[0]

```

Program di atas meminta dan memvalidasi nama pria dan nama wanita yang terlibat dalam acara. Nama-nama yang telah divalidasi disimpan dalam list terpisah.

### 3.7 Def namaultah() dan Def umurultah()

```

162
163 def namaultah():
164     nama_ultah = input('Nama orang yang berulang tahun: ')
165     kondisi1 = all(x.isspace() for x in nama_ultah)
166     kondisi2 = all(x.isalpha() for x in nama_ultah)
167     kondisi3 = all(x.isalpha() or x.isspace() for x in nama_ultah)
168     if kondisi1 == True:
169         print('Maaf, nama tidak dapat berupa karakter kosong.')
170         namaultah()
171     elif kondisi2 == True:
172         namafixultah.append(nama_ultah)
173     elif kondisi3 == True:
174         namafixultah.append(nama_ultah)
175     else:
176         print('Maaf, nama harus berupa karakter huruf. Coba lagi')
177         namaultah()
178     return namafixultah[0]
179
180 def umurultah():
181     usia = input('Masukkan usia ulang tahun (angka): ')
182     if usia.isnumeric() == False:
183         print('Maaf, umur harus berupa karakter angka. Coba lagi')
184         umurultah()
185     else:
186         umurfixultah.append(usia)
187     return umurfixultah[0]
188

```

Program di atas meminta dan memvalidasi nama dan usia orang yang berulang tahun dalam acara. Terdapat fungsi "namauliah()" untuk memvalidasi nama orang yang berulang tahun dan fungsi "umurultah()" untuk memvalidasi usia. Nama dan usia yang divalidasi disimpan dalam list yang sesuai.

### 3.8 Def judulgathering()

```
188
189 def judulgathering():
190     judul = input('Masukkan judul acara Anda: ')
191     kondisi1 = all(x.isspace() for x in judul)
192     kondisi2 = all(x.isalpha() for x in judul)
193     kondisi3 = all(x.isalnum() or x.isspace() for x in judul)
194     if kondisi1 == True:
195         print('Maaf, judul tidak dapat berupa karakter kosong.')
196         judulgathering()
197     elif kondisi2 == True:
198         judulfix.append(judul)
199     elif kondisi3 == True:
200         judulfix.append(judul)
201     else:
202         print('Maaf, nama harus berupa karakter huruf. Coba lagi')
203         judulgathering()
204     return judulfix[0]
205
```

Program di atas meminta dan memvalidasi judul acara gathering yang akan diadakan. Pengguna diminta untuk memasukkan judul acara, kemudian program memeriksa apakah judul tersebut valid berdasarkan kriteria huruf dan karakter yang diterima. Jika valid, judul acara disimpan dalam list "judulfix". Jika tidak valid, program mencetak pesan kesalahan dan meminta pengguna untuk memasukkan judul yang valid. Fungsi "judulgathering()" mengembalikan judul acara yang telah divalidasi dan disimpan dalam list "judulfix".



3.9 Def `jenis_acara()`

```

205
206 def jenis_acara():
207     print('='*50)
208     print('A. Lamaran')
209     print('B. Resepsi Pernikahan')
210     print('C. Pesta Ulang Tahun')
211     print('D. Ulang Tahun Pernikahan')
212     print('E. Gathering')
213     try:
214         x = input('Acara apa yang akan Anda adakan? (A/B/C/D/E) ')
215         if x.upper() == 'A':
216             print('Anda memilih acara lamaran')
217             data_klien.append(f'Lamaran {str(namapria())} dan {str(namawanita())}')
218         if x.upper() != 'A':
219             if x.upper() == 'B':
220                 print('Anda memilih acara resepsi pernikahan')
221                 data_klien.append(f'Resepsi pernikahan {str(namapria())} dan {str(namawanita())}')
222             if x.upper() != 'B':
223                 if x.upper() == 'C':
224                     print('Anda memilih acara pesta ulang tahun')
225                     data_klien.append(f'Pesta ulang tahun {str(namaultah())} ke-{str(umurultah())}')
226                 if x.upper() != 'C':
227                     if x.upper() == 'D':
228                         print('Anda memilih acara ulang tahun pernikahan')
229                         data_klien.append(f'Pesta ulang tahun pernikahan {str(namapria())} dan {str(namawanita())} yang ke-{str(umurultah())}')
230                     if x.upper() != 'D':
231                         if x.upper() == 'E':
232                             print('Anda memilih acara gathering')
233                             data_klien.append(f'Gathering {str(judulgathering())}')
234                         else:
235                             print('Input tidak valid')
236                             jenis_acara()
237     except KeyboardInterrupt:
238         print('\nMohon tidak menghentikan program secara paksa')
239         jenis_acara()
240 jenis_acara()
241

```

Program tersebut adalah sebuah menu interaktif yang memungkinkan pengguna memilih jenis acara yang akan diadakan. Setelah memilih, program akan menyimpan data acara beserta nama-nama yang telah divalidasi dalam list "data\_klien". Jika input pengguna tidak valid, program akan memberikan pesan kesalahan dan meminta input kembali. Program juga menangani pengecualian jika pengguna menghentikan program secara paksa dengan menekan kombinasi tombol keyboard.

## 3.10 Def undangan()

```

242 #jumlah undangan
243 tempatpilihan = booking_data[2]
244 maksimal = lokasi.lokasi[tempatpilihan]['kapasitas']
245 print('='*50)
246 print(f'Kapasitas maksimal gedung yang anda pilih adalah {maksimal}')
247 diundang = 0
248 while True:
249     undang = int(input('Berapa orang yang Anda undang? '))
250     if undang > maksimal:
251         print('Mohon maaf, kuota gedung tidak mencukupi.')
252     else:
253         print(f'Anda mengundang {undang} orang.')
254         diundang += undang
255         data_klien.append(diundang)
256         break
257
258 #jenis undangan
259 import undangan_terbaru as undt
260 print('='*50)
261 print('A. Hard Cover')
262 print('B. Soft Cover')
263 print('C. Digital')
264 def pilih_undangan():
265     undt.cover()
266     undt.verif()
267     data_klien.append(undt.cover_dipilih[-1])
268     pilih_undangan()
269

```

Program tersebut mengatur jumlah undangan dan memilih jenis undangan untuk acara yang akan diadakan. Pertama, program mendapatkan lokasi yang dipilih dari data pemesanan dan menampilkan kapasitas maksimal gedung. Pengguna diminta untuk memasukkan jumlah undangan, dan program memeriksa apakah jumlah tersebut melebihi kapasitas maksimal gedung. Selanjutnya, program meminta pengguna untuk memilih jenis undangan dari opsi yang tersedia. Informasi terkait jumlah undangan dan jenis undangan yang dipilih disimpan dalam list "data\_klien".

### 3.11 *Def musik()*

```

269 |
270 | #opsi live musik
271 | import band
272 | print('='*50)
273 | print('A. Simple')
274 | print('B. Simple Trio')
275 | print('C. Band')
276 | print('D. Exclusive Band')
277 | print('E. Light Orchestra')
278 | print('F. Big Band')
279 | print('G. Tanpa Live Musik')
280 | def musik():
281 |     band.pilihan()
282 |     band.verif()
283 |     data_klien.append(band.formasi_dipilih[-1])
284 | musik()
285 |

```

Program di atas merupakan menu interaktif untuk memilih opsi live musik acara. Pengguna diminta memilih salah satu opsi yang tersedia, kemudian program menggunakan modul "band" untuk memilih dan melakukan verifikasi terhadap opsi yang dipilih. Informasi terkait formasi live musik yang dipilih dan hasil verifikasinya disimpan dalam list "data\_klien".

### 3.12 *Def pesan\_makan() dan Def pilih\_mc()*

```

285 |
286 | #hidangan
287 | import hidangan
288 | print('='*50)
289 | print('A. Prasmanan')
290 | print('B. Catering')
291 | def pesan_makan():
292 |     hidangan.makanan()
293 |     hidangan.verif()
294 |     data_klien.append(hidangan.hidangan_dipilih[-1])
295 | pesan_makan()
296 |
297 | #MC
298 | import Modul_MC_Ago as mc
299 | print('='*50)
300 | def pilih_mc():
301 |     mc.main()
302 |     mc.verif()
303 |     data_klien.append(mc.mc_dipilih[-1])
304 | pilih_mc()
305 |

```

Program tersebut berfungsi untuk memilih opsi hidangan makanan (prasmanan atau catering) dan opsi MC (Master of Ceremony) untuk acara yang akan diadakan. Pengguna diberikan pilihan untuk memilih jenis hidangan dan MC melalui menu interaktif. Setelah

pemilihan selesai, informasi terkait pilihan hidangan dan MC yang telah divalidasi akan disimpan dalam list "data\_klien".

### 3.13 Def pilih\_dokum() dan Def pilih\_cetak()

```
306 #dokumentasi
307 import Modul_Dok_Wawa as md
308 print('='*50)
309 print('A. Fotografi')
310 print('B. Fotografi dan videografi')
311 def pilih_dokum():
312     md.pilihanD()
313     md.verif()
314     data_klien.append(md.dokum_dipilih[-1])
315 pilih_dokum()
316
317 #paket cetak foto
318 import cetak
319 print('='*50)
320 print('A. Paket Kenangan')
321 print('B. Paket Kenangan Plus')
322 print('C. Paket Pajangan')
323 print('D. Paket BigScreen')
324 print('E. Paket Komplit')
325 print('F. Tanpa Percetakan')
326 def pilih_cetak():
327     cetak.pilihancetak()
328     cetak.verif()
329     data_klien.append(cetak.cetak_dipilih[-1])
330 pilih_cetak()
331
```

Program di atas berfungsi untuk memilih opsi dokumentasi (fotografi atau fotografi dan videografi) serta paket cetak foto untuk acara yang akan diadakan. Modul "Modul\_Dok\_Wawa" digunakan untuk pemilihan opsi dokumentasi, sedangkan modul "cetak" digunakan untuk pemilihan opsi paket cetak foto. Pengguna diminta untuk memilih opsi yang diinginkan menggunakan input huruf yang sesuai. Setelah pemilihan selesai, informasi terkait opsi yang dipilih dan hasil verifikasi disimpan dalam list "data\_klien". Implementasi terkait pemilihan dan verifikasi opsi dokumentasi dan paket cetak foto terdapat di dalam modul yang diimpor sebelumnya, namun implementasi rinci tidak terlihat dalam kode yang diberikan.



3.14 *Def preview()*

```

331 |
332 | #preview output
333 | biaya_total = 0
334 | print('='*50)
335 | print('='*50)
336 | print('PREVIEW BIAYA PENGADAAN EVENT')
337 | def preview():
338 |     global biaya_total
339 |     print(f'Nama Klien = {data_klien[0]}')
340 |     print(f'Nomor telepon klien = {data_klien[1]}')
341 |     print(f'Acara yang diadakan = {data_klien[2]}')
342 |     print(f'Waktu pelaksanaan acara = {booking_data[0]}, {booking_data[1]}')
343 |     print(f'Lokasi pelaksanaan acara = {booking_data[2]}, biaya = {"Rp{:,.2f}".format(lokasi.lokasi[booking_data[2]]["harga"])}')
344 |     biaya_total += lokasi.lokasi[booking_data[2]]["harga"]
345 |     print(f'Jumlah undangan = {data_klien[3]} orang')
346 |     if data_klien[4] == 'Digital':
347 |         print(f'Jenis undangan = {data_klien[4]}, biaya = {"Rp{:,.2f}".format(undt.undangan[data_klien[4]]["Dengan harga"])}')
348 |         biaya_total += undt.undangan[data_klien[4]]["Dengan harga"]
349 |     else:
350 |         print(f'Jenis undangan = {data_klien[4]}, biaya = {"Rp{:,.2f}".format((undt.undangan[data_klien[4]]["Dengan harga"])*data_klien[3])}')
351 |         biaya_total += (undt.undangan[data_klien[4]]["Dengan harga"])*data_klien[3]
352 |     print(f'Formasi musik = {data_klien[5]}, biaya = {"Rp{:,.2f}".format(band.formasi[data_klien[5]]["harga"])}')
353 |     biaya_total += band.formasi[data_klien[5]]["harga"]
354 |     print(f'Tipe hidangan = {data_klien[6]}, biaya = {"Rp{:,.2f}".format((hidangan.tipe_hidangan[data_klien[6]]["harga"])*data_klien[3])}')
355 |     biaya_total += (hidangan.tipe_hidangan[data_klien[6]]["harga"])*data_klien[3]
356 |     print(f'MC = {data_klien[7]}, biaya = {"Rp{:,.2f}".format(mc.opsi_mc[data_klien[7]]["harga"])}')
357 |     biaya_total += mc.opsi_mc[data_klien[7]]["harga"]
358 |     print(f'Tipe dokumentasi = {data_klien[8]}, biaya = {"Rp{:,.2f}".format(md.dokumentasi[data_klien[8]]["harga"])}')
359 |     biaya_total += md.dokumentasi[data_klien[8]]["harga"]
360 |     print(f'Percetakan = {data_klien[9]}, biaya = {"Rp{:,.2f}".format(cetak.paket_cetak[data_klien[9]]["harga"])}')
361 |     biaya_total += cetak.paket_cetak[data_klien[9]]["harga"]
362 |     print('Estimasi Biaya Total (belum termasuk sewa kostum jika menghendaki:)' )
363 |     print("Rp{:,.2f}".format(biaya_total))
364 |     preview()
365 |

```

Program tersebut berfungsi untuk menampilkan preview atau pratinjau dari biaya pengadaan event berdasarkan pilihan-pilihan yang telah dibuat sebelumnya. Program mencetak informasi terkait klien, acara, waktu dan lokasi pelaksanaan acara, jumlah undangan, jenis undangan, formasi musik, tipe hidangan, MC, tipe dokumentasi, dan percetakan yang dipilih. Untuk setiap pilihan, program juga mencetak biaya yang terkait dengan pilihan tersebut. Selain itu, program menghitung estimasi biaya total berdasarkan biaya dari setiap pilihan yang telah dipilih sebelumnya. Estimasi biaya total belum termasuk biaya sewa kostum, jika ada.

3.15 *Def nota() dan Contact Person*

```

365 |
366 | #contact person
367 | import random
368 | cp = open('contact_person.txt').read().splitlines()
369 | pilih_cp = random.choice(cp)
370 | print(f'Contact Person: {pilih_cp}')
371 |
372 | #Nota
373 | def nota():
374 |     global biaya_total
375 |     biaya_total = 0
376 |     print('='*50)
377 |     print('='*50)
378 |     print('NOTA PENGADAAN EVENT')
379 |     preview()
380 |     print('LUNAS')
381 |     print('='*50)
382 |     print('='*50)
383 |

```

Program tersebut memiliki dua fungsi: "contact person" untuk memilih secara acak contact person dari daftar yang ada, dan "nota" untuk mencetak informasi rinci tentang biaya dan pilihan yang telah dibuat, serta menandai bahwa pembayaran telah dilakukan.

### 3.16 Def *catat\_output()*

```
#catat output valid
def catatan_output():
    with open('output_valid_klien.txt','a') as f:
        print('*50', file=f)
        print(f'Nama Klien = {data_klien[0]}',file=f)
        print(f'Nomor telepon klien = {data_klien[1]}',file=f)
        print(f'Acara yang diadakan = {data_klien[2]}',file=f)
        print(f'Waktu pelaksanaan acara = {booking_data[0]}, {booking_data[1]}',file=f)
        print(f'Lokasi pelaksanaan acara = {booking_data[2]}',file=f)
        print(f'Jumlah undangan = {data_klien[3]} orang',file=f)
        print(f'Jenis undangan = {data_klien[4]}',file=f)
        print(f'Formasi musik = {data_klien[5]}',file=f)
        print(f'Tipe hidangan = {data_klien[6]}',file=f)
        print(f'MC = {data_klien[7]}',file=f)
        print(f'Tipe dokumentasi = {data_klien[8]}',file=f)
        print(f'Pilihan percetakan = {data_klien[9]}', file=f)
        print('Estimasi Biaya Total:',file=f)
        print("Rp{:, .2f}".format(biaya_total),file=f)
        print('*50', file=f)
        print('\n',file=f)
```

Program tersebut berfungsi untuk mencatat output yang valid ke dalam file teks "output\_valid\_klien.txt". Fungsi "catat\_output()" digunakan untuk mencetak informasi yang relevan, seperti nama klien, nomor telepon klien, acara yang diadakan, waktu pelaksanaan acara, lokasi pelaksanaan acara, jumlah undangan, jenis undangan, formasi musik, tipe hidangan, MC, tipe dokumentasi, pilihan percetakan, dan estimasi biaya total. Setiap informasi dicetak dalam format yang sesuai dan ditulis ke dalam file "output\_valid\_klien.txt" dengan menggunakan perintah "print()" dengan argumen "file=f".

## 3.17 Def carabayar()

```
404 |
405 | #pembayaran
406 | print('Silakan melanjutkan ke pembayaran.')
407 | print('A. BCA')
408 | print('B. Mandiri')
409 | print('C. BNI')
410 | def carabayar():
411 |     try:
412 |         cara_bayar = input('Anda akan membayar melalui bank apa? ')
413 |         if cara_bayar.upper() == 'A':
414 |             print('Nomor rekening bank BCA = 1234567890 a.n. abcdefg')
415 |         if cara_bayar.upper() != 'A':
416 |             if cara_bayar.upper() == 'B':
417 |                 print('Nomor rekening bank Mandiri = 1234567890 a.n. abcdefg')
418 |             if cara_bayar.upper() != 'B':
419 |                 if cara_bayar.upper() == 'C':
420 |                     print('Nomor rekening bank BNI = 1234567890 a.n. abcdefg')
421 |                 else:
422 |                     print('Input tidak valid')
423 |                     carabayar()
424 |     except KeyboardInterrupt:
425 |         print('\nMohon tidak menghentikan program secara paksa')
426 |         carabayar()
427 | carabayar()
428 |
```

Program mencetak opsi metode pembayaran yang tersedia menggunakan fungsi "print()". Pengguna diminta untuk memilih salah satu opsi metode pembayaran dengan memasukkan huruf yang sesuai (A, B, atau C) menggunakan fungsi "input()". Program menggunakan pernyataan percabangan "if" untuk mengecek pilihan pengguna dan mencetak nomor rekening yang sesuai terkait dengan metode pembayaran yang dipilih.

Jika pengguna memilih opsi selain A, B, atau C, program akan mencetak pesan "Input tidak valid" dan memanggil kembali fungsi "carabayar()" untuk meminta pengguna memilih kembali.

Jika pengguna secara paksa menghentikan program dengan menekan tombol Ctrl+C (KeyboardInterrupt), program akan mencetak pesan "Mohon tidak menghentikan program secara paksa" dan memanggil kembali fungsi "carabayar()" untuk meminta pengguna memilih kembali. Program berfungsi untuk membantu pengguna memilih metode pembayaran yang sesuai untuk melanjutkan ke tahap pembayaran acara.

## 3.18 Def generate()

```
428
429 #otp
430 import random
431 kode_otp = random.randrange(100000,999999)
432 def generate():
433     kesempatan = 3
434     x = open('otp.txt', 'w')
435     tulis = str(kode_otp)
436     x.write(tulis)
437     x.close()
438     print('Mohon perhatikan bahwa kesempatan Anda mengisi kode OTP hanya 3 kali.')
439     while kesempatan != 0:
440         otp_input = int(input('Masukkan kode OTP = '))
441         if otp_input != kode_otp:
442             print(f'Coba lagi. Kesempatan Anda {kesempatan - 1} kali.')
443             kesempatan -= 1
444         else:
445             print('Terima kasih! Pembayaran telah terverifikasi.')
446             nota()
447             catat_output()
448             break
449     if kesempatan == 0:
450         print('Anda gagal verifikasi pembayaran sebanyak 3 kali.\nMohon maaf, pesanan Anda kami anggap tidak valid.')
451         with open('booking.txt','r') as baca_datatempat:
452             data_ada = baca_datatempat.readlines()
453             datahapus = str(booking_data)
454             with open('booking.txt','w') as baca_datatempat:
455                 for baris in data_ada:
456                     if baris.strip('\n') != datahapus:
457                         baca_datatempat.write(baris)
458             booking_data.clear()
459             data_klien.clear()
460     generate()
```

Program tersebut berfungsi sebagai mekanisme untuk menghasilkan dan memverifikasi kode OTP dalam proses pembayaran acara. Pengguna diberikan 3 kesempatan untuk memasukkan kode OTP yang dihasilkan secara acak. Jika kode yang dimasukkan sesuai, program mencetak pesan pembayaran terverifikasi dan melanjutkan dengan mencetak nota serta mencatat output valid. Jika pengguna gagal memasukkan kode yang benar sebanyak 3 kali, program mencetak pesan bahwa verifikasi pembayaran tidak berhasil dan menghapus data pemesanan. Data pemesanan dihapus dan program mengosongkan data yang telah ada.



## BAB IV

## HASIL RUNNING PROGRAM

Bab ini menjelaskan tentang *output* yang ditampilkan program setelah menjalankan keseluruhan program.

## 4.1 Biodata, Waktu dan Tempat Pelaksanaan Acara

```
=====
Masukkan nama Anda = wawa
Masukkan nomor telepon Anda = +622345678
=====
```

Gambar 4.1.1 Biodata Klien

Dari **Gambar 4.1.1** menunjukkan hasil *running* nama beserta nomor telepon yang diinput oleh klien.

```
=====
Notes: Anda hanya dapat mengadakan acara secepat-cepatnya satu bulan dari sekarang.
Kapan Anda akan mengadakan acara? (DD/MM/YYYY)
Tanggal - Masukkan angka tanggal (DD) = 21
Bulan - Masukkan angka bulan (MM) = 09
Tahun - Masukkan angka tahun (YYYY) = 2023
Tanggal valid.
A. The Royal Surakarta Heritage
B. Pose In Hotel
C. Graha Saba Buana
D. Ramada Suites (outdoor)
E. Adhiwangsa Hotel & Convention (outdoor)
Pilih di sini (A/B/C/D/E) = d
{'jenis ruangan': 'outdoor', 'kapasitas': 200, 'alamat': 'Jl. Adi Sucipto No.56, Gatak, Gajahan, Kec.Colomadu, Kab. Karanganyar', 'harga': 5000000}
Apakah Anda yakin? (Y/N) y
Tempat yang Anda pilih adalah Ramada Suites
=====
Selamat, tempat yang Anda pesan tersedia! Silakan melanjutkan pesanan Anda.
=====
```

Gambar 4.1.2 Waktu dan Tempat Pelaksanaan

Dari **Gambar 4.1.2** menunjukkan hari, tanggal, bulan, tahun, dan tempat diikuti dengan kalimat ‘Tanggal Valid’ yang menunjukkan belum ada klien sebelumnya yang memilih pada waktu dan tempat yang dipilih klien baru.

## 4.2 Opsi

## 4.2.1 Acara

Pada hasil *running* acara menunjukkan hasil pemilihan jenis acara yang menampilkan keterangan lanjutan sesuai acara yang dipilih.

```
=====
A. Lamaran
B. Resepsi Pernikahan
C. Pesta Ulang Tahun
D. Ulang Tahun Pernikahan
E. Gathering
Acara apa yang akan Anda adakan? (A/B/C/D/E) a
Anda memilih acara lamaran
Nama Pria: ahmad
Nama Wanita: tsamara
=====
Kapasitas maksimal gedung yang anda pilih adalah 200
Berapa orang yang Anda undang? 199
Anda mengundang 199 orang.
=====
```

Gambar 4.2.1 Acara

Dari **Gambar 4.2.1** menunjukkan klien memilih acara lamaran yang menampilkan keterangan lanjutan berupa nama pria dan nama wanita beserta jumlah tamu undangan sesuai kapasitas gedung yang dipilih.

## 4.2.2 Jenis Undangan

```
=====
A. Hard Cover
B. Soft Cover
C. Digital
Pilih jenis cetak isi undangan! (A/B/C) a
Undangan berjenis Hard Cover adalah undangan fisik yang
dapat Anda berikan kepada setiap tamu yang diundang
dengan bahan hard cover yang awet.
Biaya percetakan akan dikalikan jumlah undangan.
{'Undangan anda akan dicetak': 'Hard Cover', 'Dengan harga': 8500}
Apakah Anda yakin? (Y/N) y
Jenis undangan yang Anda pilih adalah Hard Cover
=====
```

**Gambar 4.2.2** Jenis Undangan

Dari **Gambar 4.2.2** menunjukkan pilihan klien terkait jenis cetak isi undangan yang diikuti deskripsi singkat beserta harganya.

## 4.2.3 Live Musik

```
=====
A. Simple
B. Simple Trio
C. Band
D. Exclusive Band
E. Light Orchestra
F. Big Band
G. Tanpa Live Musik
Formasi apa yang Anda pilih? (A/B/C/D/E/F/G) f
{'alat musik': 'Drum, Bass, Keyboard, Gitar, Saxophone, Cello, Viola, Violin 1, Violin 2, Terompet, Trombone, Alto, Saxophone', 'penyanyi': 4, 'harga': 10000000}
Apakah Anda yakin? (Y/N) y
Formasi yang Anda pilih adalah big band
=====
```

**Gambar 4.2.3** Live Musik

Dari **Gambar 4.2.3** menunjukkan pilihan klien terkait pilihan *live* musik yang diikuti keterangan lanjutan beserta harganya.

## 4.2.4 Jenis Hidangan

```
=====
A. Prasmanan
B. Catering
Anda memilih tipe hidangan apa? (A/B) b
Catering adalah tipe hidangan yang disajikan melalui jasa waiter/waitress.
Makanan akan disajikan kepada para tamu dengan diantarkan oleh waiter/waitress
sesuai dengan porsi yang ditentukan.
{'harga': 60000}
Apakah Anda yakin? (Y/N) y
Tipe hidangan yang Anda pilih adalah catering
=====
```

**Gambar 4.2.4** Jenis Hidangan

Dari **Gambar 4.2.4** menunjukkan pilihan klien terkait pilihan hidangan tamu yang diikuti deskripsi singkat beserta harganya.

## 4.2.5 Pembawa Acara

```
=====
Silakan pilih jumlah MC yang Anda inginkan:
1. Satu orang MC
2. Dua orang MC
Masukkan pilihan Anda (1 atau 2): 2
Anda memilih dua orang MC.
{'harga': 2000000}
Apakah Anda yakin? (Y/N) y
Jumlah MC yang Anda pilih adalah 2 orang
=====
```

**Gambar 4.2.5** Pembawa Acara

Dari **Gambar 4.2.5** menunjukkan pilihan klien terkait pilihan pembawa acara yang diikuti keterangan lanjutan beserta harganya.

## 4.2.6 Dokumentasi

```
=====
A. Fotografi
B. Fotografi dan videografi
Pilihan dokumentasi apa yang Anda pilih? (A/B) b
Dengan memilih paket fotografi dan videografi, tentunya Anda
tidak hanya puas dengan hasil foto dari tim dokumentasi kami,
tapi tim dokumentasi kami juga akan memfasilitasi Anda
dengan pembuatan video berkualitas tinggi.
{'harga': 2000000}
Apakah Anda yakin? (Y/N) y
Dokumentasi yang Anda pilih adalah fotografi dan videografi
=====
```

**Gambar 4.2.6** Dokumentasi

Dari **Gambar 4.2.6** menunjukkan pilihan klien terkait pilihan dokumentasi yang diikuti deskripsi singkat beserta harganya.

## 4.2.7 Paket Percetakan

```
=====
A. Paket Kenangan
B. Paket Kenangan Plus
C. Paket Pajangan
D. Paket BigScreen
E. Paket Komplit
F. Tanpa Percetakan
Paket percetakan apa yang Anda pilih? (A/B/C/D/E/F) e
{'isi': '10 lembar foto 8R + 5 lembar foto 12R + 3 foto dalam pigura kayu 6R + 3 foto dalam pigura kayu 12R + 1 foto dalam pigura kayu 20R', 'harga': 595000}
Apakah Anda yakin? (Y/N) y
Paket percetakan yang Anda pilih adalah Paket Komplit
=====
```

**Gambar 4.2.7** Paket Percetakan

Dari **Gambar 4.2.7** menunjukkan pilihan klien terkait pilihan paket percetakan yang diikuti keterangan lanjutan beserta harganya.



## 4.3 Preview Biaya Pengadaan Event

```
=====
PREVIEW BIAYA PENGADAAN EVENT
Nama Klien = wawa
Nomor telepon klien = +622345678
Acara yang diadakan = Lamaran ahmad dan tsamara
Waktu pelaksanaan acara = Kamis, 2023-09-21
Lokasi pelaksanaan acara = Ramada Suites, biaya = Rp50,000,000.00
Jumlah undangan = 199 orang
Jenis undangan = Hard Cover, biaya = Rp1,691,500.00
Formasi musik = big band, biaya = Rp10,000,000.00
Tipe hidangan = catering, biaya = Rp11,940,000.00
MC = 2 orang, biaya = Rp2,000,000.00
Tipe dokumentasi = fotografi dan videografi, biaya = Rp2,000,000.00
Percetakan = Paket Komplit, biaya = Rp595,000.00
Estimasi Biaya Total (belum termasuk sewa kostum jika menghendaki:)
Rp78,226,500.00
Contact Person: Najwa: +62 812-2951-7332
Silakan melanjutkan ke pembayaran.
A. BCA
B. Mandiri
C. BNI
Anda akan membayar melalui bank apa? c
Nomor rekening bank BNI = 1234567890 a.n. abcdefg
Mohon perhatikan bahwa kesempatan Anda mengisi kode OTP hanya 3 kali.
Masukkan kode OTP = 484178
Terima kasih! Pembayaran telah terverifikasi.
=====
```

**Gambar 4.3** Preview Biaya

Dari **Gambar 4.3** menunjukkan hasil keseluruhan input pilihan dari klien beserta harganya, gambar tersebut juga menampilkan pemilihan pembayaran melalui 3 (tiga) pilihan bank diikuti nomor rekening bank sesuai yang dipilih dan kode OTP untuk verifikasi pembayaran.

## 4.4 Nota Pengadaan Event

```
=====
NOTA PENGADAAN EVENT
Nama Klien = wawa
Nomor telepon klien = +622345678
Acara yang diadakan = Lamaran ahmad dan tsamara
Waktu pelaksanaan acara = Kamis, 2023-09-21
Lokasi pelaksanaan acara = Ramada Suites, biaya = Rp50,000,000.00
Jumlah undangan = 199 orang
Tipe hidangan = catering, biaya = Rp11,940,000.00
MC = 2 orang, biaya = Rp2,000,000.00
Tipe dokumentasi = fotografi dan videografi, biaya = Rp2,000,000.00
Percetakan = Paket Komplit, biaya = Rp595,000.00
Estimasi Biaya Total (belum termasuk sewa kostum jika menghendaki:)
Rp78,226,500.00
LUNAS
=====
```

**Gambar 4.4** Nota Pengadaan Event

Dari **Gambar 4.4** menampilkan hasil akhir dari program yaitu Nota Pengadaan Event. Nota ini berisi nama klien, nomor telepon klien, acara yang diadakan, waktu pelaksanaan acara, lokasi pelaksanaan acara, jumlah undangan, tipe hidangan, mc, tipe dokumentasi, percetakan, dan yang terakhir estimasi biaya total dengan keterangan LUNAS.